

KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Electrical Engineering

Kernel-based data fusion for machine learning: methods and applications in bioinformatics and text mining

Shi Yu

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

Nov. 2009

Kernel-based data fusion for machine learning: methods and applications in bioinformatics and text mining

Shi YU

Jury:
Prof. dr. ir. H. Hens, president
Prof. dr. ir. B. De Moor, promoter
Prof. dr. ir. Y. Moreau, co-promoter
Prof. dr. ir. G. Bontempi (ULB)
Prof. dr. ir. T. De Bie (Univ. of Bristol)
Prof. dr. L. Dehaspe
Prof. dr. ir. P. Dupont (UCL)
Prof. dr. ir. J. Suykens

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

U.D.C. number

Nov 2009

© Katholieke Universiteit Leuven – Faculty of Engineering
Address, B-3001 Leuven (Belgium) Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever. All rights reserved.

No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher. Legal depot number
ISBN number

Preface

This thesis is the product of my years as a research assistant in the Bioinformatics group, SCD/SISTA research division of the Electrical Engineering Department of the Katholieke Universiteit Leuven. It has been a exciting journey of learning and growth, in a relaxing and quiet Gothic town, accompanied with many interesting colleagues and friends, and will be a memorable and treasurable experience in my life.

The thesis dedicate to my beloved family, my wife, my son and my parents, for their love, patience, support, understanding and sacrifice.

My deepest gratitude goes to my promoter Prof. Bart De Moor for giving me the opportunity to join the group and start my doctoral research. I would like appreciate for his continuous supports with respect to my funding and all the university paperwork required through these years. Through him, I also acknowledge all the help I received from the administrative and professional staff at SCD/SISTA. I would like to express my heartfelt gratitude to Professor Yves Moreau, my co-promoter, for his illuminating guidance and precise instructions during my study and work. I appreciate for his confidence on my work and his enthusiasms in cross-discipline topics which have a profound impact on my personal interest and independent research, and his active and detailed discussions and contributions to the manuscripts we prepared together. I would like to equally express my sincere appreciation to Prof. Johan Suykens, who is also the member in my doctoral committee and examination committee, for his introduction of kernel methods in the early days. During the preparation of the papers we collaborated, I was impressed by his concrete and rigorous attitude. The mathematical expressions and the structure of several Chapters in this thesis were significantly improved by his suggestions.

I would like to acknowledge Prof. Hugo Hens, Prof. Luc Dehaspe, Prof. Pierre Dupont, Prof. Gianluca Bontempi and Prof. Tijl De Bie for their acceptance as members of my examination committee who provided me valuable constructive comments and suggestions for improvement of this work. I am honored of having a jury of such excellent quality and renowned international reputation. In particular,

I was inspired by Tijl's interesting work on kernel fusion for gene prioritization problem in my first year of Ph.D program. Since then on I has been attracted to the topic of kernel fusion and Tijl had many insightful discussions with me on various topics, and the communication continues after he moved to Bristol.

Next, I would like to convey my gratitude and respect to several colleagues who led me to the bioinformatics and machine learning world. Dr. Steven Van Vooren, Dr. Bert Coessens and Dr. Frizo Janssens were the daily supervisors of my master thesis and they introduced me to the world of bioinformatics through biomedical text mining. Steven spent substantial amount of time explaining me the code, correcting my first paper and answering my questions in research. Frizo also provided me well-processed experimental data and contributed a lot to our collaborated papers. Dr. Carlos Alzate encouraged me to write my own code for machine learning problems, which has been proved very helpful and necessary for the algorithmic innovations in my later work. Dr. Kristiaan Pelckmans provided many professional explanations to my questions along these years. Carlos and Kristiaan have introduced me lots of knowledge about kernel methods, as well as successful experiments and failed attempts on many interesting problems.

This thesis would not have been possible without the collaboration with some smart and diligent colleagues in SCD/SISTA. Léon-Charles Tranchevent did excellent works retrieving and organizing a massive amount of biological data. He is one of the most important contributors of my papers. I am thankful for Xinhai Liu for his interests and insights of my work. Together with him, we applied and refined our methods in the scientometrics problems. I was delighted to collaborate with Dr. Sonia Leach, Tillmann Falck and Anneleen Daemen on many various topics and applications.

I would also like to take this opportunity to express my thanks to the kind supports outside our research group. Prof. Marc Van Hulle (Laboratorium voor Neurofysiologie, Campus Gasthuisberg) encouraged and provided me recommendation to pursue the high quality training and research. Prof. Wolfgang Glänzel (Steunpunt O&O Indicatoren) allows me to freely try my methods on the very expensive and high quality data. Since the year 2007, Dr. Jieping Ye (Arizona State University) has been discussing with me on various topics and his outstanding works deepen my understanding of the essence of kernel based data fusion.

This work is a product of four years of innovation, collaboration, experiments, freedom, happiness and frustrations. The obtained results are encouraging for further developments. It was a joy for me writing this thesis and I, sincerely, would like to share it with you.

Shi Yu
Leuven, November 2009

Abstract

The emerging problem of data fusion offers plenty of opportunities, also raises lots of interdisciplinary challenges in computational biology. Developments in high-throughput technologies generate Terabytes of genomic data at awesome rate, which raises big challenges to the conventional statistical modeling methods. How to combine and leverage the mass amount of data sources to obtain significant and complementary high-level knowledge is a state-of-art interest in machine learning and bioinformatics communities.

In this dissertation, we investigate the issue of data fusion in supervised learning and unsupervised learning, which are fundamental and active topics in machine learning research. The dissertation is organized as three parts. Firstly, we propose theoretical discussions about a set of unsupervised algorithms in statistics and machine learning. To equip these algorithms with the workhorse of kernel based data fusion, we investigate and extend the technique of multiple kernel learning proposed in the supervised learning context. Our discussion is centered around the feasibility and the efficiency of kernel fusion for unsupervised learning. Many aspects of the problem are addressed, including the sparseness of kernel coefficients, the efficiency of optimization and the choice of loss functions. Secondly, we focus on the innovation of kernel fusion algorithms to solve specific data fusion problems. Three new algorithms are proposed. Two of them are clustering algorithms that combine multiple data sources as kernels or Laplacians in clustering analysis. The third algorithm extends the canonical correlation analysis to a weighted kernelized version. Our discussion of canonical correlation focuses on the effect of dimensionality reduction in multi-source learning. Thirdly, we substantiate the values of the proposed algorithms in several real bioinformatics and journal scientometrics applications. These applications are algorithmically categorized as two approaches, ranking and clustering. In both categorizations, the machine learning performance is significantly improved by thoroughly combining information of various data sources and data representations. In particular, the kernel based novelty detection algorithm is successfully applied in gene prioritization problem and the database and algorithm are solidified in an open online software.

Acronyms

1-SVM	One class Support Vector Machine
AdacVote	Adaptive cumulative Voting
AL	Average Linkage Clustering
ARI	Adjusted Rand Index
BSSE	Between Clusters Sum of Squares Error
CCA	Canonical Correlation Analysis
CL	Complete Linkage
CSPA	Cluster based Similarity Partition Algorithm
CV	Controlled Vocabulary
CVs	Controlled Vocabularies
EAC	Evidence Accumulation Clustering
EACAL	Evidence Accumulation Clustering with Average Linkage
ESI	Essential Science Indicators
EVD	Eigenvalue Decomposition
FDA	Fisher Discriminant Analysis
GO	The Gene Ontology
HGPA	Hyper Graph Partitioning Algorithm
ICD	Incomplete Cholesky Decomposition
ICL	Inductive Constraint Logic
IDF	Inverse Document Frequency

ILP	Inductive Logic Programming
KCCA	Kernel Canonical Correlation Analysis
KEGG	Kyoto Encyclopedia of Genes and Genomes
KFDA	Kernel Fisher Discriminant Analysis
KL	Kernel Laplacian Clustering
KM	K means clustering
LDA	Linear Discriminant Analysis
LSI	Latent Semantic Indexing
LS-SVM	Least Squares Support Vector Machine
MCLA	Meta Clustering Algorithm
MEDLINE	Medical Literature Analysis and Retrieval System Online
MKCCA	Multiple Kernel Canonical Correlation Analysis
MKL	Multiple Kernel Learning
MSV	Mean Silhouette Value
NAML	Nonlinear Adaptive Metric Learning
NMI	Normalized Mutual Information
PCA	Principal Component Analysis
PPI	Protein Protein Interaction
PSD	Positive Semi-definite
QCLP	Quadratic Constrained Linear Programming
QCQP	Quadratic Constrained Quadratic Programming
OKKC	Optimized data fusion for Kernel K-means Clustering
OKLC	Optimized data fusion for Kernel Laplacian Clustering
QMI	Quadratic Mutual Information Clustering
QP	Quadratic Programming
RBF	Radial Basis Function
RI	Rand Index

SC	Spectral Clustering
SDP	Semi-definite Programming
SILP	Semi-infinite Linear Programming
SIP	Semi-infinite Programming
SL	Single Linkage Clustering
SMO	Sequential Minimization Optimization
SOCP	Second Order Cone Programming
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
TSSE	Total Sum of Squares Error
WL	Ward Linkage
WMKCCA	Weighted Multiple Kernel Canonical Correlation Analysis
WoS	Web of Science
WSSE	Within Cluster Sum of Squares Error

Notations

$\vec{w}$	non-zero column vector
$\vec{w}^T$	transpose of $\vec{w}$
Q, P	positive (semi-) definite matrix
A, B	real value matrix
W	orthogonal real value matrix
$\mathbb{C}^{n\times m}$	$n \times m$ matrix of complex values
$\mathbb{R}^{n\times m}$	$n \times m$ matrix of real values
$\mathbb{R}^m$	$m \times 1$ vector of real values
$\mathcal{F}$	Hilbert space
C_{xx}	sample covariance matrix of data X
C_{xy}	sample covariance matrix of data X and Y of equal sample size
K	kernel matrix (not centered or submatrix of a centered kernel matrix)
G	centered kernel matrix
L	Laplacian matrix
Ω	combined kernel matrix
θ	coefficient of the kernel matrix
$\vec{\alpha}, \vec{\beta}, \vec{\eta}$	column vector of dual variables
λ, κ, ν	regularization parameter
i	index parameter for the number of classes, $i = 1, ..., q$
j	index parameter for the number of kernels, $j = 1, ..., p$
k	index parameter for the number of data points, $k = 1, ..., N$
τ	index parameter for the loop of iterative algorithm
μ	the mean vector
$\mathcal{L}$	Lagrangian
$\mathcal{J}$	objective function
L_∞	infinity norm
L_1	1-norm
L_2	2-norm

Contents

Contents vii

1 Introduction 1

1.1 General Background 1

1.2 Historical background of multi-source learning and data fusion . . 4

1.2.1 Canonical correlation and its probabilistic interpretation . . 4

1.2.2 Inductive logic programming and the multi-source learning search space 5

1.2.3 Additive model and ensemble learning 6

1.2.4 Bayesian networks for data fusion 8

1.2.5 Kernel based data fusion 10

1.3 Objectives and Challenges 18

1.4 Chapter by Chapter Overview 19

1.5 Relevant research topics in ESAT-SCD, K.U.Leuven 21

1.6 Contributions of the thesis 23

2 Rayleigh quotient-type problems in unsupervised learning 27

2.1 Optimization of Rayleigh quotient 27

2.2 Rayleigh quotient-type problem in unsupervised learning 30

2.3 Summary 36

3 Kernel fusion for one class and multiclass problem 39

3.1 Introduction . . . 39

3.2 Problem Definition . . . 40

3.3 One class SVM kernel fusion for ranking . . . 44

3.4 Kernel fusion for classification . . . 46

3.4.1 Support Vector Machine . . . 46

3.4.2 Least Squares Support Vector Machine. . . . 48

3.5 Case studies of genomic data fusion . . . 50

3.5.1 Disease relevant gene prioritization by genomic data fusion 50

3.5.2 Clinical decision support by integrating microarray and proteomics data . . . 53

3.6 Summary . . . 56

4 Kernel fusion for large scale data 59

4.1 Introduction . . . 59

4.2 Low rank kernel fusion for One class Support Vector Machine (1-SVM) 60

4.2.1 Fixed-Size 1-SVM Multiple Kernel Learning (MKL): Second Order Cone Programming (SOCP) formulation . . . 60

4.2.2 Fixed-Size 1-SVM MKL: Separable Quadratic Programming (QP) formulation . . . 62

4.2.3 Numerical experiment . . . 62

4.3 Large scale multi-class MKL . . . 65

4.3.1 Low rank approximation for Support Vector Machine (SVM) MKL: conic formulation . . . 65

4.3.2 Low rank kernel fusion for SVM MKL: separable QP formulation . . . 66

4.3.3 SVM MKL: Semi-infinite Programming (SIP) formulation . 68

4.3.4 Least Squares Support Vector Machine (LS-SVM) MKL: SIP formulation . . . 72

4.3.5 Numerical Experiments . . . 75

4.4 Summary 79

5 Optimized data fusion for Kernel K-means Clustering 81

5.1 Introduction 81

5.2 Objective of K-means clustering 82

5.3 Optimizing multiple kernel matrices for K-means 83

5.4 Bi-level optimization of K-means on multiple kernel matrices . . . 86

5.4.1 The role of cluster assignment 86

5.4.2 Optimizing the kernel coefficients as KFDA 87

5.4.3 Regularization term 88

5.4.4 Soft clustering 89

5.5 Optimized data fusion for kernel K-means clustering 89

5.6 Experimental result 90

5.6.1 Synthetic data sets 90

5.6.2 UCI machine learning data sets 94

5.7 Summary 96

6 Optimized data fusion for K-means Laplacian Clustering 97

6.1 Introduction 97

6.2 Clustering by multiple kernels and Laplacians 98

6.2.1 Background of Kernel Laplacian Clustering (KL) clustering 98

6.2.2 Combining multiple kernels and multiple Laplacians 99

6.3 Experiment 102

7 Weighted Multiple Kernel Canonical Correlation 107

7.1 Introduction 107

7.2 Weighted Multiple Kernel Canonical Correlation 108

7.2.1 Linear Canonical Correlation Analysis (CCA) on Multiple Data Sets 108

7.2.2 Multiple Kernel CCA 109
7.2.3 Weighted Multiple Kernel Canonical Correlation Analysis (WMKCCA) 110
7.3 Computational Issue 111
7.3.1 Standard Eigenvalue Problem for WMKCCA 111
7.3.2 Incomplete Cholesky Decomposition 112
7.3.3 Incremental Eigenvalue Decomposition (EVD) solution for WMKCCA 114
7.4 Learning by WMKCCA 115
7.5 Experiment 116
7.5.1 Classification in canonical spaces 116
7.5.2 Efficiency of incremental EVD solution 119
7.6 Summary 120

8 Multi-view text mining for disease gene prioritization 121

8.1 Introduction 121
8.2 Background: Computational gene prioritization 122
8.3 Single view gene prioritization: a fragile model with respect to the uncertainty 123
8.4 Data fusion for gene prioritization: distribution free method 125
8.5 Multi-view text mining gene prioritization 128
8.5.1 Construction of controlled vocabularies from multiple bio-ontologies 128
8.5.2 Vocabularies selected from subsets of ontologies 130
8.5.3 Merging and mapping of controlled vocabularies 131
8.5.4 Text mining 133
8.5.5 Dimensionality reduction of gene-by-term data by Latent Semantic Indexing 134
8.5.6 Algorithms and evaluation of gene prioritization task 134
8.5.7 Benchmark data set of disease genes 136

8.6 Results . . . 136
8.6.1 Multi-view performs better than single view . . . 136
8.6.2 Effectiveness of multi-view demonstrated on various number of views . . . 139
8.6.3 Effectiveness of multi-view demonstrated on disease examples 139
8.7 Discussion and Summary . . . 142

9 Endeavour MerKator: an open software for cross-species gene prioritization 143

9.1 Introduction . . . 143
9.2 Conceptual overview of Endeavour MerKator . . . 144
9.3 Implementation of Endeavour MerKator . . . 145
9.3.1 Offline and online processes of the software . . . 145
9.3.2 Centering the kernel matrices . . . 148
9.3.3 Missing values . . . 149
9.3.4 Cross-species information integration . . . 151
9.4 Summary . . . 162

10 Clustering by heterogeneous data sources 165

10.1 Introduction . . . 165
10.2 Clustering algorithms for data fusion . . . 166
10.2.1 Ensemble clustering . . . 166
10.2.2 Kernel fusion . . . 167
10.2.3 Clustering by data fusion . . . 169
10.3 Experimental Results . . . 170
10.3.1 Digit Recognition Data . . . 170
10.3.2 Disease gene clustering by multi-view text mining . . . 173
10.4 Summary . . . 178

11 Combining lexical information and citation links in journal sets analysis 179

11.1 Introduction 179
11.2 Background: hybrid clustering for journal set analysis 180
11.3 Processing of journal data 181
11.3.1 Data sources and data processing 181
11.3.2 Kernels of lexical information 182
11.3.3 Laplacians of citations 182
11.3.4 Essential Science Indicators (ESI) labels 183
11.4 Experimental Setup 183
11.5 Results 184
11.6 Optimized data fusion for Kernel Laplacian Clustering (OKLC)-*light* 188
11.7 Summary 189

12 Conclusions and Future Research **190**

12.1 Conclusions 190
12.2 Future research 191
12.2.1 Kernel based sensor fusion 191
12.2.2 Integration of expression data and interaction networks for disease genes identification 192
12.2.3 A joint framework of data fusion with feature selection . . . 193
12.2.4 Other relevant topics 194
12.3 Closing remarks 194

A List of algorithms **197**

B List of applications and applied algorithms **199**

Curriculum vitae **215**

Chapter 1

Introduction

When I have presented one point of a subject and the student cannot from it, learn the other three, I do not repeat my lesson, until one is able to.

– "The Analects, VII.", Confucius (551 BC - 479 BC) –

1.1 General Background

The history of *learning* is accompanied by the pace of evolution and the progress of civilization. Surprisingly, several modern ideas of learning, such as pattern analysis and machine intelligence, can be traced back thousands of years in the analects of oriental philosophers [37] (the quoted text) and the intelligent artifacts appeared in Greek mythologies [128]. *Machine learning*, a contemporary topic rooted in computer science and engineering, has always being inspired and enriched by the unremitting efforts of biologists and psychologists during their investigation and understanding of the nature. The Baldwin effect, proposed by James Mark Baldwin 110 years ago, concerns the the costs and benefits of *learning* in the context of evolution and greatly influences the development of evolutionary computation. The introduction of *perceptron* and the *backpropagation* algorithm have united the joint-efforts of mathematicians, scientists and engineers to replicate the biological *intelligence* by artificial means. About 15 years ago, Vapnik [178] introduced the support vector method which makes use of kernel functions, which has offered plenty of opportunities to solve complicated problems but also brought lots of interdisciplinary challenges in statistics, optimization theory and the applications therein. At present, many powerful methods have been

invented for various learning problems. Nevertheless, if we compare these methods with the biochemical intelligence of a primitive organism, we should realize that our expedition to imitate the adaptability and the exquisiteness of *learning*, has just begun.

Learning from multiple sources

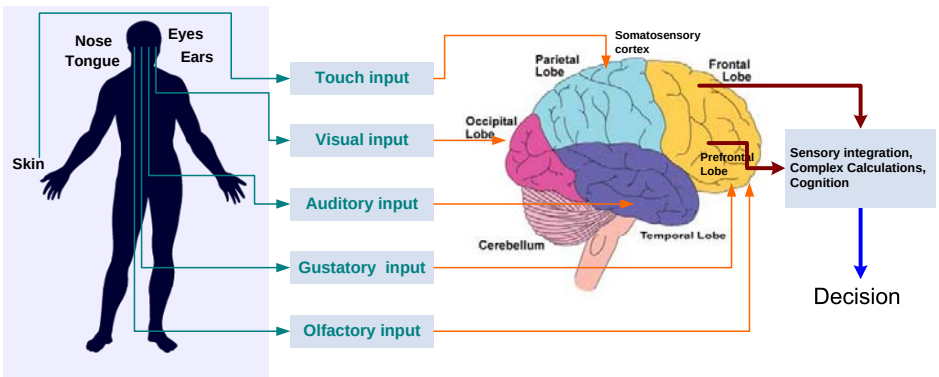


Figure 1.1: The decision of human beings relies on the integration of multiple senses. Information travels from the eyes is forwarded to the occipital lobes of the brain. Sound information is analyzed by the auditory cortex in the temporal lobes. Smell and taste are analyzed in the olfactory bulb contained in prefrontal lobes. Touch information passes to the somatosensory cortex laying out along the brain surface. Information comes from different senses is integrated and analyzed at the frontal and prefrontal lobes of the brain, where the most complex calculations and cognitions occur. The figure of human body is adapted from [35]. The figure of brain is adapted from [78].

As shown in Figure 1.1, our brains are amazingly adept at *learning from multiple sources*. Information travels from multiple senses is forwarded to the brain, and is integrated and prioritized by complex calculations using biochemical energy. This type of *integration* and *prioritization* are extraordinary adaptive to the environment and the stimulus. For example, when one is sitting in the auditorium and listening to a talk of a lecturer, the most important information comes from the visual and auditory sense; though at the very moment the brain is also receiving inputs from the other senses, i.e., the temperature, the smell, the taste, etc., it exquisitely suppresses the irrelevant senses and keeps the mind focusing on the most relevant information. This type of *prioritization* is also observed in the senses of the same category. For instance, some sensitive parts of the body, i.e., fingertips, toes, lips, etc., have much stronger representations than the less sensitive areas. For human beings, some abilities of multiple-source learning

are given by birth, some others are obtained by professional training. Figure 1.2 illustrates the mechanical drawing of a simple component in a telescope, which is composed of projections in several perspectives. To manufacture it, an experienced operator of the machine tool need to investigate this drawing, combine the multiple 2-D perspectives as a 3-D reconstruction of the component. This process also relies on the human brain's ability to integrate different input streams into a representation of its environment. However, in contrast to the body senses example illustrated before, the reconstruction of 3-D structure from multiple 2-D views requires professional training and practice. In the past two centuries, in mechanical industry, the communications between designers and manufactories, the productions from tiny components to giant megastrcutures have been all based on this type of multi-perspective representation and learning. Currently, specialized computer software (i.e., AutoCAD, TurboCAD, etc.) are able to automate the human-like reconstruction process using advanced images and graphics techniques, visualization methods, and geometry algorithms.

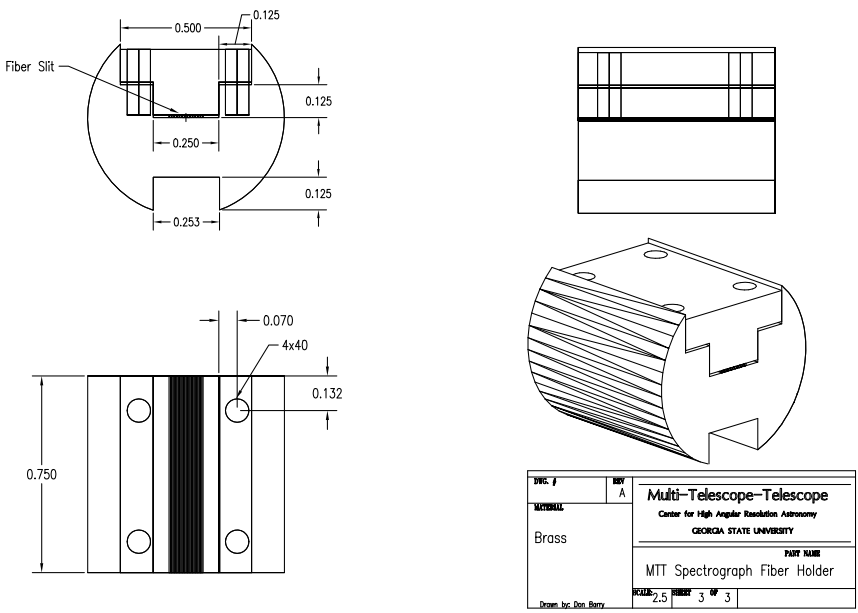


Figure 1.2: The method of the multiview orthographic projection applied in modern mechanical drawing origins from the applied geometry method developed by Gaspard Monge in 1780s [167]. The 3-D structure of a component is projected on three orthogonal planes and several views are obtained. These views are known as the right side view, the front view, and the top view in the inverse clockwise order. The drawing of the telescope component is adapted from [15].

In *machine learning*, we are motivated to imitate the functions of the brain to incorporate *multiple data sources*. Human brains are powerful in learning abstractive knowledge but computers are good at detecting statistical significance and numerical patterns. In the era of information overflow, *data mining* and *machine learning* are indispensable tools to extract information and knowledge from data. To achieve this, many efforts have been spent on the invention of sophisticated methods and the construction of huge scale database. Beside these efforts, an important strategy is to investigate the *dimension* of information and data, which may enable us to coordinate the data ocean into homogeneous threads and to gain complete insight of the knowledge. For example, in time-varied data the common dimension is *time*, such as the stock market data, the weather monitoring data, the medical records of a patient, and so on. In bioinformatics research, another common dimension of data is *facet*: the amount of data is ever increasing due to the recent advances in *high throughput* biotechnologies and they can be viewed as representations of genomic entities projected in various facets. Thus, the idea of incorporating more *facets* of genomic data in analysis may be beneficial, by reducing the noise, as well as improving statistical significance and leveraging the interactions and correlations between the data sets to obtain more refined and higher-level information [173]. This technique is also known as *data fusion* and considerable work has been devoted to the current issue of *genomic data fusion*.

1.2 Historical background of multi-source learning and data fusion

1.2.1 Canonical correlation and its probabilistic interpretation

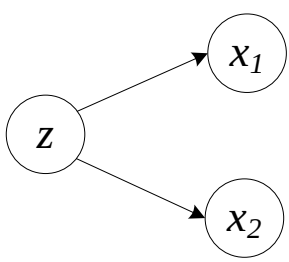


Figure 1.3: Graphical model for canonical correlation analysis.

The early approaches of multi-source learning can be dated back to the statistical methods extracting a set of features for each data source by optimizing a dependency criterion, such as CCA [77] and its kernel variants [71][103], and

methods that optimize mutual information between extracted features [16]. As to be shown in Chapter 2, CCA is solved analytically as a generalized eigenvalue problem. The probabilistic interpretation of CCA is provided by Bach & Jordan [12] and Klami & Kaski [96]. For example, as proposed in [12], the maximum likelihood estimates of the parameters W_1,W_2,Ψ_1,Ψ_2,μ_1,μ_2 for the model defined in Figure 1.3 and by

$$z \sim \mathcal{N}(0, I_d), \quad min\{m_1, m_2\} \geq d \geq 1$$

$$x_1|z \sim \mathcal{N}(W_1 z + \mu_1, \Psi_1), \quad W_1 \in \mathbb{R}^{m_1 \times d}, \ \Psi_1 \succeq 0$$

$$x_2|z \sim \mathcal{N}(W_2 z + \mu_2, \Psi_2), \quad W_2 \in \mathbb{R}^{m_2 \times d}, \ \Psi_2 \succeq 0$$

are

$$\widehat{W_1} = \tilde{\Sigma}_{11} U_{1d} M_1$$

$$\widehat{W_2} = \tilde{\Sigma}_{22} U_{2d} M_2$$

$$\widehat{\Psi}_1 = \tilde{\Sigma}_{11} - \widehat{W_1}\widehat{W_1}^T$$

$$\widehat{\Psi}_2 = \tilde{\Sigma}_{22} - \widehat{W_2}\widehat{W_2}^T$$

$$\hat{\mu}_1 = \tilde{\mu}_1$$

$$\hat{\mu}_2 = \tilde{\mu}_2$$

where M_1,$M_2 \in \mathbb{R}^{d \times d}$ are arbitrary matrices such that $M_1 M_2^T = P_d$ and the spectral norms of M_1 and M_2 are smaller than one, and the i-th columns of U_{1d} and U_{2d} are the first d canonical directions, and P_d is the diagonal matrix of the first d canonical correlations.

The analytical model and probabilistic interpretation of CCA enable the use of local CCA models to identify common underlying patterns or same distributions from data consist of independent pairs of related data points. CCA has been widely applied in multi-source information retrieval, signal processing, robotics, and many other fields.

1.2.2 Inductive logic programming and the multi-source learning search space

Inductive Logic Programming (ILP) [126] is a supervised machine learning method combines automatic learning and first order logic programming [115]. It requires three main sets of information, the automatic solving and deduction theory set aside [149]:

1. a set of known vocabulary, rules, axioms or predicates, describing the domain knowledge base $\mathcal{K}$;
2. a set of positive examples $\mathcal{E}^+$ that the system is supposed to describe or characterize with the set of predicates of $\mathcal{K}$;
3. a set of negative examples $\mathcal{E}^-$ that should be excluded from the deducted description or characterization.

Given these data, the ILP solver is then able to find the set of hypotheses $\mathcal{H}$, expressed with the predicates and terminal vocabulary of $\mathcal{K}$ such that the largest possible subset of $\mathcal{E}^+$ verifies $\mathcal{H}$, and such that the largest possible subset of $\mathcal{E}^-$ does not verify $\mathcal{H}$. The hypotheses in $\mathcal{H}$ are searched in a so-called *hypothesis* space. Different strategies can be used to explore the hypothesis search space (i.e., the Inductive Constraint Logic (ICL) proposed by De Raedt & Van Laer [50], etc.). The search stops when a clause that covers no negative example while covering some positive examples is reached. At each step, the best clause is refined by adding new literals to its body, applying variable substitutions, etc. The search space can be restricted by a so-called language bias (i.e., a declarative bias used by ICL [49]).

In a multi-source learning problem, data points indexed by the same identifier are represented in various data sources. Aggregation is a operation to merge the same data points of different sources. In ILP, the aggregation function can be simply the set union associated to inconsistency elimination. However, the aggregation may result in a huge search space, which in many situations the learning algorithm is not able to cope with it or takes too much computational time [62]. One possible solution is to specify an efficient language bias to reduce the learning search space as proposed by Fromont *et al.* [62] using a new learning process from the aggregated data.

1.2.3 Additive model and ensemble learning

The idea of using multiple classifiers has received increasing attention as it has been realized that such approaches can be more robust (i.e. less sensitive to the tuning of their internal parameters, to inaccuracies and other defects in the data) and more accurate than a single classifier alone. Since these approaches are characterized as to generate multiple learning models on a single data or multiple data sources and to ensemble these models as a unified "powerful" model, we denote them as *Ensemble Learning*. Bagging and boosting are the most powerful learning techniques introduced in the last twenty years.

Bootstrap aggregation, or bagging, is a technique proposed by Breiman [30] that can be used with many classification methods and regression methods to

reduce the variance associated with prediction, and thereby improve the prediction process. It is a relatively simple idea: many bootstrap samples are drawn from the available data, some prediction method is applied to each bootstrap sample, and then the results are combined, by averaging for regression and simple voting for classification, to obtain the overall prediction, with the variance being reduced due to the averaging [165].

Boosting, like bagging, is a committee-based approach that can be used to improve the accuracy of classification or regression methods. Unlike bagging, which uses a simple averaging of results to obtain an overall prediction, boosting uses a weighted average of results obtained from applying a prediction method to various samples [165]. The motivation for boosting is a procedure that combines the outputs of many "weak" classifiers to produce a powerful "committee". The most popular boosting framework is proposed by Freund and Schapire called "AdaBoost.M1" [60]. The "weak classifier" in boosting can be assigned as any classifier, i.e., when applying the classification tree as the "base learner" the improvements are often dramatic [29]. Though boosting is originally applied to combine "weak classifiers", some approaches also involve "strong classifiers" in the boosting framework, i.e., the ensemble of feed-forward neural networks [54][101].

In boosting, the elementary objective function is extended from a single source to multiple sources as an additive expansion. More generally, basis function expansions take the form

$$f(\vec{x}) = \sum_{j=1}^{p} \theta_j b(\vec{x}; \vec{\gamma}_j), \tag{1.1}$$

where $\theta_j, j = 1, ..., p$ are the expansion coefficients, and $b(\vec{x}; \vec{\gamma}) \in \mathbb{R}$ are usually simple functions of the multivariate input $\vec{x}$, characterized by a set of parameters $\vec{\gamma}$ [72]. The notion of additive expansions in mono-source can be straightforwardly extended to multi-source learning as

$$f(\vec{x}_j) = \sum_{j=1}^{p} \theta_j b(\vec{x}_j; \vec{\gamma}_j), \tag{1.2}$$

where the function considers $\vec{x}_j,\ j = 1, ..., p$ as multiple representations of a data point. The prediction function is therefore given by

$$P(x) = \text{sign}\left(\sum_{j=1}^{p} \theta_j P_j(\vec{x}_j)\right), \tag{1.3}$$

where $P_j(\vec{x}_j)$ is the prediction function of each single data source. Additive expansions like this are the essence of many machine learning techniques proposed for enhanced mono-source learning or multi-source learning.

1.2.4 Bayesian networks for data fusion

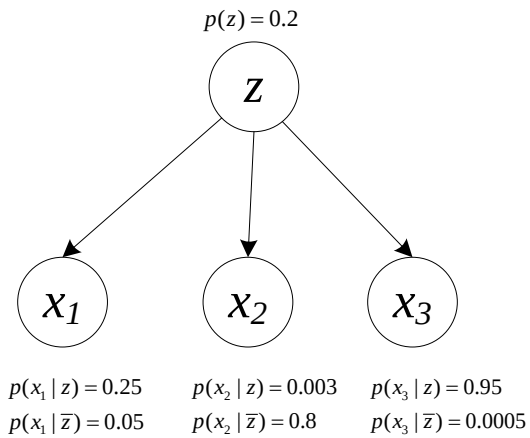


Figure 1.4: A simple Bayesian network

Bayesian networks [139] are probabilistic models that graphically encode probabilistic dependencies between random variables [140]. The graphical structure of the model imposes qualitative dependence constraints. A simple example of Bayesian network is shown in Figure 1.4. A directed arc between variables z and x_1 denotes conditional dependency of x_1 on z, as determined by the direction of the arc. The dependencies in Bayesian networks are measured quantitatively. For each variable and its parents this measure is defined using a conditional probability function or a table (i.e., the Conditional Probability Tables (CPTs)). In Figure 1.4, the measure of dependency of x_1 on z is the probability $p(x_1|z)$. The graphical dependency structure and the local probability models completely specify a Bayesian network probabilistic model. Hence, Figure 1.4 defines $p(z, x_1, x_2, x_3)$ to be

$$p(z, x_1, x_2, x_3) = p(x_1|z)p(x_2|z)p(x_3|z)p(z). \tag{1.4}$$

To determine a Bayesian network from the data, one need to learn its structure (structural learning) and its conditional probability distributions (parameter learning) [64]. To determine the structure, the sampling methods based on Markov chain Monte Carlo (MCMC) or the variational methods are often adopted. The two key components of a structure learning algorithm are *searching* for "good" structures and *scoring* these structures. Since the number of model structures is large (super-exponential), a search method is required to decide which structures to score. Even with few nodes, there are too many possible networks to exhaustively score each one. When the number of nodes is large, the task becomes very

challenging. Efficient structure learning algorithm design is an active research area. For example, the K2 greedy search algorithm [39] starts with an initial network (possibly with no (or full) connectivity) and iteratively adding, deleting, or reversing an edge, measuring the accuracy of the resulting network at each stage, until a local maxima is found. Alternatively, a method such as simulated annealing guides the search to the global maximum [64][130]. There are two common approaches used to decide on a "good" structure. The first is to test whether the conditional independence assertions implied by the network structure are satisfied by the data. The second approach is to assess the degree to which the resulting structure explains the data. This is done using a score function which is typically based on approximations of the full posterior distribution of the parameters for the model structure is computed. In real applications, it is often required to learn the structure from incomplete data containing missing values. Several specific algorithms are proposed for structural learning with incomplete data, for instance, the AMS-EM greedy search algorithm proposed by Friedman [61], the evolutionary Algs and MCMC proposed by Myers [127], the Robust Bayesian Estimation proposed by Ramoni and Sebastiani [144], the Hybrid Independence Test proposed by Dash and Druzdzel [44], etc.

The second step of Bayesian network building consists of estimating the parameters that maximize the likelihood that the observed data came from the given dependency structure. To consider the uncertainty about parameters θ in a prior distribution $p(\theta)$, one use data d to update this distribution, and hereby obtain the posterior distribution $p(\theta|d)$ using Bayes' theorem as

$$p(\theta|d) = \frac{p(d|\theta)p(\theta)}{p(d)}, \quad \theta \in \Theta, \tag{1.5}$$

where Θ is the parameter space, d is a random sample from the distribution $p(d)$ and $p(d|\theta)$ is likelihood of θ. To maximize the posterior, the Expectation-Maximization (EM) algorithm is often used. The prior distribution describes one's state of knowledge (or lack of it) about the parameter values before examining the data. The prior can also be incorporated in structural learning. Obviously, the choice of the prior is a critical issue in Bayesian network learning, in practice, it rarely happens that the available prior information is precise enough to lead to an exact determination of the prior distribution. If the prior distribution is too narrow it will dominate the posterior and can be used only to express the precise knowledge. Thus, if one has no knowledge at all about the value of a parameter prior to observing the data, the chosen prior probability function should be very broad (non-informative prior) and at relatively to the expected likelihood function.

By far we have very briefly introduced the Bayesian networks. As probabilistic models, Bayesian networks provide a convenient framework for combination of evidences from multiple sources as a data fusion approach. The data can be integrated as full integration, partial integration and decision integration [64], which are briefly concluded as follows:

Full integration

In full integration, the multiple data sources are combined at the data level as one data set. In this manner the developed model can contain any type of relationship among the variables in different data sources [64].

Partial integration

In partial integration, the structure learning of bayesian network is performed separately on each data, which results in multiple dependency structures have only one variable (the outcome) in common. The outcome variable allows joining the separate structures into one structure. In the parameter learning step, the parameter learning proceeds as usual because this step is independent of how the structure was built. Partial integration forbids link among variables of multiple sources, which is similar to imposing additional restrictions in full integration where no links are allowed among variables across data sources [64].

Decision integration

The decision integration method learns a sperate model for each data source and the probabilities predicted for the outcome variable are combined using the weighted coefficients. The weighted coefficients are trained using the model building data set with randomizations [64].

1.2.5 Kernel based data fusion

In the learning phase of Bayesian networks, a set of training data is used either to obtain the point estimate of the parameter vector or to determine a posterior distribution over this vector. The training data is then discarded, and predictions for new inputs are based purely on the learned structure and parameter vector [21]. This approach is also used in nonlinear parametric models such as neural networks [21].

However, there is a set of machine learning techniques keep the training data points during the prediction phase. For example, the Parzen probability model, the nearest-neighbour classifier, the support vector machines, etc. These classifiers typically require a metric to be defined that measures the similarity of any two vectors in input space, as known as the “dual representation”.

Dural representation, kernel trick and Hilbert space

Many linear parametric models can be re-casted into an equivalent "dual representation" in which the predictions are also based on linear combinations of a *kernel function* evaluated at the training data points [21]. To achieve this, the data representations are embedded into a vector space $\mathcal{F}$, called feature space (Hilbert space) [178] [179] [41][152]. A key characteristic of this approach is that the embedding in Hilbert space is generally defined implicitly, by specifying an inner product in it. Thus, for a pair of data items, $\vec{x}_1$ and $\vec{x}_2$, denoting their embeddings as $\phi(\vec{x}_1)$ and $\phi(\vec{x}_2)$, the inner product of the embedded data $\langle \phi(\vec{x}_1), \phi(\vec{x}_2) \rangle$ is specified via a kernel function $K(\vec{x}_1, \vec{x}_2)$, known as the *kernel trick* or the *kernel substitution*, given by

$$K(\vec{x}_1, \vec{x}_2) = \phi(\vec{x}_1)^T \phi(\vec{x}_2). \tag{1.6}$$

From this definition, one of the most significant advantages is to handle symbolic objects (i.e., categorical data, string data, etc.), thereby greatly expanding the ranges of problems that can be addressed. Another important advantage is brought by the nonlinear high-dimensional feature mapping $\phi(\vec{x})$ from the original space $\mathbb{R}$ to the Hilbert space $\mathcal{F}$. By this mapping, the problems that are not separable by a linear boundary in $\mathbb{R}$ may become separable in $\mathcal{F}$ because according to the VC dimension theory [180], the capacity of linear classifiers is enhanced in the high dimensional space. The dual representation enables us to build interesting extensions of many well-known algorithms by making use of the kernel trick. For example, the kernel trick can be applied to principal component analysis in order to develop a nonlinear variant of PCA [153]. Other examples of algorithms extend by kernel trick include nearest-neighbour classifiers and the kernel Fisher discriminant [122][123].

Support Vector Classifiers

The problem of finding linear separating hyperplane on training data consists of N pairs $(\vec{x}_1, y_1), ..., (\vec{x}_N, y_N)$, with $\vec{x}_k \in \mathbb{R}^m$ and $y_k \in \{-1, +1\}$, the optimal separating hyperplane is formulated as

$$\min_{\vec{w},b} \ \frac{1}{2}\vec{w}^T\vec{w} \tag{1.7}$$

$$\text{s.t.} \quad y_k(\vec{w}^T\vec{x}_k + b) \geq 1, \quad k = 1, ..., N,$$

where $\vec{w}$ is the norm vector of the hyperplane, b is the bias term. The geometry meaning of the hyperplane is shown in Figure 1.5. Hence we are looking for the hyperplane that creates the biggest *margin* M between the training points for

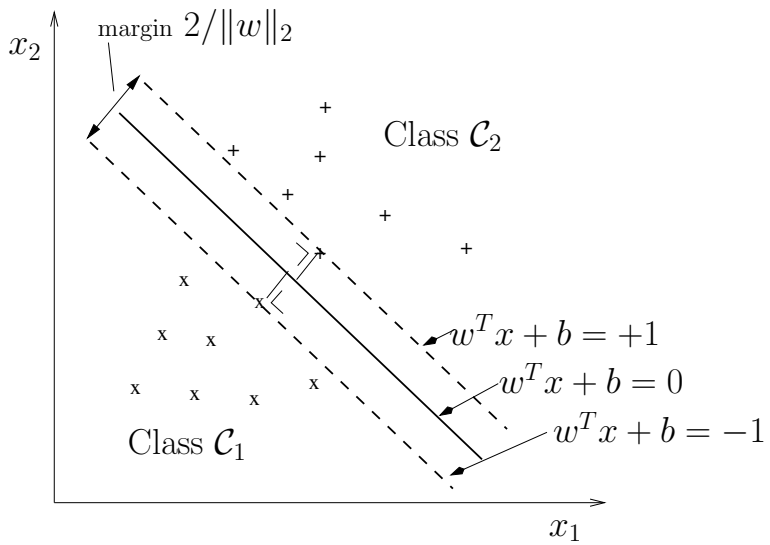


Figure 1.5: The geometry interpretation of a support vector classifier. Figure obtained from [166].

class 1 and -1. Note that $M = 2/||\vec{w}||$. This is a convex optimization problem (quadratic objective, linear inequality constraints) and the solution can be obtained as a QCLP problem.

In most cases, the training data represented by the two classes is not perfectly separable, so the classifier needs to tolerate some errors (allows some points to be on the wrong side of the margin). We define the slack variables $\vec{\xi} = [\xi_1, ..., \xi_N]^T$ and modify the constraints in (1.7) as

$$\min_{\vec{w},b} \quad \frac{1}{2}\vec{w}^T\vec{w} \tag{1.8}$$

$$\text{s.t.} \quad y_k(\vec{w}^T\vec{x}_k + b) \geq M(1 - \xi_k), \quad k = 1, ..., N$$

$$\xi_k \geq 0, \sum_{k=1}^{N} \xi_k = C, \quad k = 1, ..., N,$$

where $C \geq 0$ is the constant bounding the total misclassifications. The problem in (1.8) is still convex (quadratic objective, linear inequality constraints) and it corresponds to the "standard" support vector classifier [24][152] if we replace $\vec{x}_i$ with the embeddings $\phi(\vec{x}_i)$, given by

$$\min_{\vec{w},b,\vec{\xi}} \ \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{k=1}^{N}\xi_k \tag{1.9}$$

$$\text{s.t.} \quad y_k[\vec{w}^T\phi(\vec{x}_k) + b] \geq 1 - \xi_k, \quad k = 1, ..., N$$

$$\xi_k \geq 0, \quad k = 1, ..., N.$$

The Lagrange (primal) function is

$$\boxed{\text{P:}} \ \min_{\vec{w},b,\vec{\xi}} \ \frac{1}{2}\vec{w}^T\vec{w} + \lambda\sum_{k=1}^{N} - \sum_{k=1}^{N}\alpha_k\Big\{y_k\big[\vec{w}^T\phi(\vec{x}_k^T) + b\big] - (1 - \xi_k)\Big\} - \sum_{i=1}^{N}\beta_k\xi_k, \tag{1.10}$$

$$\text{s.t.} \ \ \alpha_k \geq 0, \quad \beta_k \geq 0, \quad k = 1, ..., N,$$

where α_i, β_i are Lagrange multipliers. The conditions of optimality are given by

$$\begin{cases} \frac{\partial}{\partial\vec{w}} = 0 \rightarrow \vec{w} = \sum_{k=1}^{N}\alpha_k y_k\phi(\vec{x}_k) \\ \frac{\partial}{\partial\vec{\xi}} = 0 \rightarrow 0 \leq \alpha_k \leq C, \quad k = 1, ..., N \\ \frac{\partial}{\partial b} = 0 \rightarrow \sum_{k=1}^{N}\alpha_k y_k = 0. \end{cases} \tag{1.11}$$

By substituting (1.11) in (1.10), we obtain the Lagrange dual objective function as

$$\boxed{\text{D:}} \ \max_{\vec{\alpha}} \ -\frac{1}{2}\sum_{k,l=1}^{N}\alpha_k\alpha_l y_k y_l\phi(\vec{x}_k)^T\phi(\vec{x}_l) + \sum_{k=1}^{N}\alpha_k \tag{1.12}$$

$$\text{s.t.} \quad 0 \leq \alpha_k \leq C, \quad k = 1, ..., N$$

$$\sum_{k=1}^{N}\alpha_k y_k = 0.$$

To maximize the dual problem in (1.12) is a simpler convex quadratic programming problem than the primal (1.10). Especially, the Karush-Kuhn-Tucker conditions including the constraints

$$\alpha_k\Big\{y_k\big[\vec{w}^T\phi(\vec{x}_k) + b\big] - (1 - \xi_k)\Big\} = 0,$$

$$\beta_k\xi_k = 0,$$

$$y_k\big[\vec{w}^T\phi(\vec{x}_k) + b\big] - (1 - \xi_k) \geq 0,$$

for $k = 1, ..., N$ characterize the unique solution to the primal and dual problem.

Support Vector Classifier for multiple sources and Kernel fusion

As discussed before, the additive expansions play a fundamental role in extending mono-source learning algorithms to multi-source learning cases. Analogously, to extend the support vector classifiers on multiple feature mappings, the objective in (1.9) can be rewritten as

$$\begin{aligned}
&\min_{\overrightarrow{\mathbf{w}},b,\vec{\theta},\vec{\xi}} \ \frac{1}{2}\overrightarrow{\mathbf{w}}^T\overrightarrow{\mathbf{w}} + C\sum_{k=1}^{N}\xi_k \qquad (1.13)\\
&\text{s.t.} \quad y_k\left[\overrightarrow{\mathbf{w}}^T\Phi(\vec{x}_k)+b\right] \geq 1-\xi_k, \quad k=1,...,N\\
&\qquad \overrightarrow{\mathbf{w}}^T\Phi(\vec{x}_k) = \sum_{j=1}^{p}\left(\sqrt{\theta_j}\vec{w}_j^T\phi_j(\vec{x}_k)\right)\\
&\qquad \xi_k \geq 0, \quad k=1,...,N\\
&\qquad \theta_j \geq 0, \ \sum_{j=1}^{p}\theta_j = 1, \quad j=1,...,p,
\end{aligned}$$

where $\phi_j(\vec{x}_k)$ are multiple feature mappings on $\vec{x}_k$, $\vec{w}_j$ is the norm vector of separating hyperplane corresponding to the j-th feature map, $\overrightarrow{\mathbf{w}}$ and $\Phi(\cdot)$ are the additive combined models implicitly representing the sum of $\left[\vec{w}_j^T\phi_j(\vec{x}_j)\right]_{j=1,...,p}$ and θ_j are the coefficients assigned on each feature mapping in the additive model. However, the inner product $\sqrt{\theta_j}\vec{w}_j$ makes the objective (1.13) non-convex so it needs to be replaced as a variable substitution $\vec{\eta}_j = \sqrt{\theta_j}\vec{w}_j$. Thus the objective is rewritten as

$$\begin{aligned}
&\boxed{\text{P:}} \quad \min_{\vec{\eta},b,\vec{\theta},\vec{\xi}} \ \frac{1}{2}\sum_{j=1}^{p}\vec{\eta}_j^T\vec{\eta}_j + C\sum_{k=1}^{N}\xi_k \qquad (1.14)\\
&\qquad \text{s.t.} \quad y_k\left[\sum_{j=1}^{p}\left(\vec{\eta}_j^T\phi_j(\vec{x}_k)\right)+b\right] \geq 1-\xi_k, \quad k=1,...,N\\
&\qquad\qquad \xi_k \geq 0, \quad k=1,...,N\\
&\qquad\qquad \theta_j \geq 0, \ \sum_{j=1}^{p}\theta_j = 1, \quad j=1,...,p,
\end{aligned}$$

where $\vec{\eta}_j$ are the scaled (by $\sqrt{(}\theta_j)$) norm vectors of the separating hyperplanes for the additive model of multiple feature mappings. In the formulations mentioned

above we assume that multiple feature mappings are created on a mono-source problem. It is analogous and straightforward to extend the same objective for multi-source problems. The investigation of this problem has been pioneered by Lanckriet *et al.* [105] and the solution is established in the dual representations as a min-max problem, given by

$$\boxed{\text{D:}} \quad \min_{\vec{\theta}} \max_{\vec{\alpha}} \quad -\frac{1}{2}\sum_{k,l=1}^{N} \alpha_k \alpha_l y_k y_l \sum_{j=1}^{p} \left(\theta_j K_j(\vec{x}_k, \vec{x}_l)\right) + \sum_{k=1}^{N} \alpha_k \tag{1.15}$$

$$\text{s.t.} \quad 0 \geq \alpha_k \geq C, \quad k = 1, ..., N$$

$$\sum_{k=1}^{N} \alpha_k y_k = 0,$$

$$\theta_j \geq 0, \ \sum_{j=1}^{p} \theta_j = 1, \quad j = 1, ..., p,$$

where $K_j(\vec{x}_k, \vec{x}_l) = \phi(\vec{x}_k)^T \phi(\vec{x}_l)$ are kernel tricks applied on multiple feature mappings. The symmetric, positive semidefinite kernel matrices K_j resolve the heterogeneities of genomic data sources (i.e., vectors, strings, trees, graphs, etc.) such that they can be merged additively as a single kernel, moreover, the coefficients of kernels θ_j are non-uniform, which means the information of multiple sources is leveraged adaptively. The technique of combining multiple support vector classifiers in the dual representations is also called *kernel fusion*.

Loss functions for Support Vector Classifiers

In Support Vector Classifiers, there are many criteria to assess the quality of an target estimation based on observations during the learning. These criteria are represented as different *loss functions* in the primal problem of Support Vector Classifiers, given by

$$\min_{\vec{w}} \frac{1}{2}\vec{w}^T\vec{w} + \lambda \sum_{k=1}^{N} L[y_k, f(\vec{x}_k)], \tag{1.16}$$

where $L[y_k, f(\vec{x}_k)]$ is the *loss function* of class label and prediction value penalizing the objective of the classifier. The examples shown above are all based on a specific loss function called "hinge" loss as $L[y_k, f(\vec{x}_k)] = |1 - y_k f(\vec{x}_k)|_+$, where the subscript "+" indicates positive part. The loss function is also related to the *risk* or *generalization error*, which is an important measure of the goodness of the classifier. The choice of the loss function is a non-trivial issue related to the estimating the joint probability distribution $p(x, y)$ on the data x and its label y,

which is general unknown because the training data only gives us an incomplete knowledge of $p(x, y)$. Table 1.1 presents several popular loss functions adopted in Support Vector Classifiers.

Table 1.1: Some popular loss functions for Support Vector Classifiers.

Loss Function	$L[y, f(\vec{x})]$	Classifier name
Binomial Deviance	$log[1 + e^{-yf(\vec{x})}]$	logistic regression
Hinge Loss	$\|1 - yf(\vec{x})\|_+$	SVM
Squared Error	$[1 - yf(\vec{x})]^2$ (equality constraints)	LS-SVM
L_2 norm	$[1 - yf(\vec{x})]^2$ (inequality constraints)	2-norm SVM
Huber's Loss	$\begin{cases} -4yf(\vec{x}), & yf(\vec{x}) < -1 \\ [1 - yf(x)]^2, & \text{otherwise} \end{cases}$	

Kernel based data fusion: a bioinformatics perspective

The kernel fusion framework has been originally proposed to solve the classification problems in computational biology. In the past decade, most of the efforts invented on this topic have been focused on supervised learning. As the matter of fact, this framework is also useful to solve a wide range of machine learning problems, such as one class learning (novelty detection), unsupervised learning (clustering), regression and others (as shown in Figure 1.6). Therefore, many opportunities exist in the advances of kernel fusion, however, there are also many challenges ahead. One of the main challenge is that kernel fusion is not established yet for unsupervised learning, and this will be one of the main topics addressed in this thesis.

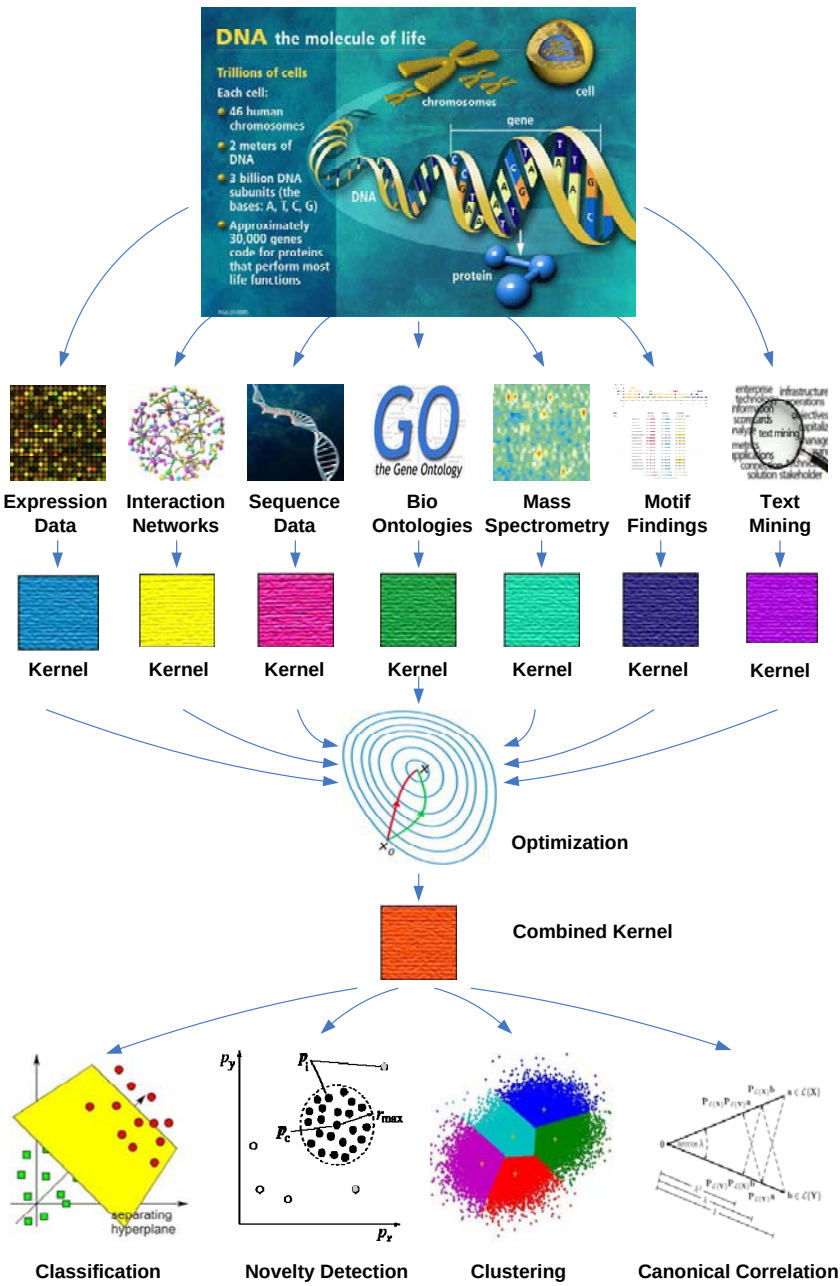


Figure 1.6: Conceptual map of kernel based genomic data fusion

1.3 Objectives and Challenges

The main objective of this thesis is to extend existing technique in supervised learning and develop new unsupervised learning methods on the basis of kernel fusion. Unfortunately, unsupervised learning is generally a harder problem then the supervised case so there are many challenges ahead.

Non-convex problem on unlabeled data

In supervised learning, the data samples are usually labeled or partially labeled whereas in unsupervised problem the samples are totally unlabeled. To optimize an objective with unlabeled data is a difficult problem, which often results in a non-convex solution where the global optimality is hard to determine. For example, K-means clustering is solved as a non-convex stochastic process and it has lots of local minima. When incorporating a non-convex unsupervised learning problem with the convex kernel fusion method, the issues of convexity and convergence are critical problems.

Large scale data and computational complexity

Unsupervised learning usually deals with large amount of data, which also increases the computational burden of kernel fusion. In the supervised case, the model is often trained on a small number of labeled data and then generalized on the test data. Therefore, the major computational burden is determined by the training process. The complexity of model generalization on the test data is often linear. For example, given N training data and M test data, the computational complexity of the SVM training using a single kernel ranges from $O(N^2)$ to $O(N^3)$; the complexity of predicting labels on the test data is $O(M)$. In contrast, one cannot split the data for unsupervised learning into training and test parts. The popular K-means clustering has the complexity of $O(k(N+M)dl)$, where k is the number of clusters, d is the complexity to compute the distance, and l is the number of iterations. The kernel fusion procedure involving both training and test data has much larger computational burden than the supervised case. For instance, the semi-definite programming (SDP) solution of MKL proposed by Lanckriet *et al.* [106] has the complexity up to $O((p+N+M)^2(k+N+M)^{2.5})$ [191], which makes it almost infeasible on large scale problem. So, the computational burden of kernel fusion on large scale data is also a critical issue.

Model evaluation and data collection

In data fusion based supervised learning, model selection is more challenging. For instance, in clustering problem the model evaluation usually relies on the statistical validation, which is often measured as various internal indices, such as Silhouette index, Jaccard index, Modularity, and so on. However, these indices are mostly data dependent thus they are not consistent with each other among heterogeneous data sources, which makes the model selection problem more difficult. In contrast, evaluating models by external evaluations using the ground truth labels, such as Rand Index, Normalized Mutual Information, etc., is more reliable to select the optimal model. Unfortunately, the ground truth labels may not always be available for clustering problem. Therefore, how to select unsupervised learning model in data fusion applications is also one of the main challenges.

In machine learning, most benchmark data sets were proposed for single source learning. In many data fusion approaches, people generate multiple data sources artificially using different distance measures or kernel functions on the same data set. Thus, the information to integrate is usually highly redundant, which makes the approach less meaningful and the performance not significant. To demonstrate the true merit of data fusion, we should consider the real applications using genuine heterogeneous data sources.

1.4 Chapter by Chapter Overview

Chapter 2 investigates several unsupervised learning problems and summarizes their objectives as a common (generalized) Rayleigh quotient form. In particular, it shows the relationship between the Rayleigh quotient and the Fisher Discriminant Analysis (FDA), which serves as the basis of many supervised learning methodologies. The FDA is also related to the kernel fusion approach formulated in the LS-SVM. Clarifying this connection provides the theoretical grounding for us to combine kernel fusion methods with concrete unsupervised algorithms.

Chapter 3 reviews kernel fusion, as known as the MKL, in one class and classification problems. Moreover, it proposes several novel results. Firstly, we generalizes the L_∞ MKL formulation proposed by Lanckriet *et al.* to a novel L_2 formulation. The L_∞ and L_2-norm differ at the optimization of different norms in terms of multiple kernels in the dual problem. Secondly, we introduce the MKL formulation in the LS-SVM and show the proof about the equivalence between the FDA and the Ridge Regression.

Chapter 4 proceeds the topic of Chapter 3 and investigates kernel fusion on large scale data. We compare two approaches to reduce the computational burden of the MKL on large scale problem. Firstly, we reduce the scale of optimization problem

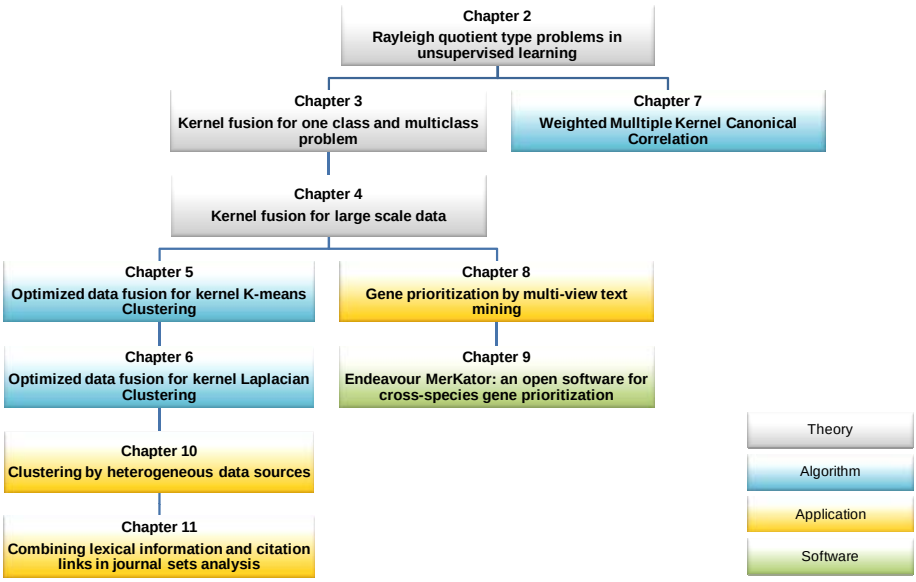


Figure 1.7: Overview of the relationships between the different chapters of this thesis.

by low rank approximations of kernel matrices. Secondly, we compare various optimization techniques, such as Quadratic Constrained Quadratic Programming (QCQP), SOCP, separable QP and SIP, to transform the large scale problems into new tasks of small scales. The main finding of this chapter is that LS-SVM MKL can be solved very efficiently in the SIP formulation. The SIP LS-SVM MKL also serves as the workhorse of the algorithms and the applications to be discussed in the remaining chapters.

Chapter 5 proposes a new Optimized data fusion for Kernel K-means Clustering (OKKC) algorithm [193] as a kernel fusion based clustering algorithm. This algorithm is non-convex but is proved to converge locally. Comparing with some related work, the proposed algorithm has much simpler objective and procedure, moreover, it performs better on synthetic and benchmark data sets.

Chapter 6 proceeds the topic of Chapter 5 and considers the integration of kernels with Laplacians in clustering. We propose a new algorithm, called OKLC [192], to combine the attribute representations with the graph representation for clustering.

Chapter 7 discusses Canonical Correlation Analysis, a different unsupervised learning problem than clustering. A new method called WMKCCA is proposed to leverage the importance of different data sources in the CCA objective. Beside the derivation of mathematical models, we present some preliminary results of using

the mappings obtained by WMKCCA as the common information extracted from multiple data sources.

Chapter 8 starts to present several real applications about kernel fusion. Firstly, an application about biomedical literature mining is introduced. This approach combines several Controlled Vocabularies (CVs) using ensemble methods and kernel fusion methods to improve the accuracy of gene prioritization. Experimental result shows that the combination of multiple CVs in text mining outperforms over the various individual Controlled Vocabulary (CV) methods thus provides an interesting approach to exploit information provided by the myriad of different bio-ontologies.

Chapter 9 continues to discuss the gene prioritization problem. To further exploits the information among genomic data sources and the phylogenetic evidences among different species, we design and develop an open software, Endeavour MerKator [195], to perform cross-species gene prioritization by genomic data fusion.

Chapter 10 considers the problem of clustering by heterogeneous data fusion and concludes various methodologies in a unified framework. This framework consists of two general approaches, ensemble clustering and kernel fusion, and provides a set of solutions for data fusion applications. In this chapter, the clustering framework is demonstrated on two applications: multi-source digit recognition and disease gene clustering by multi-view text mining.

Chapter 11 demonstrates a Scientometrics application to combine attribute based lexical similarities with graph based citation links for journal mapping. The attribute information is transformed as kernels while the citations are represented as Laplacian matrices, then are all combined by OKLC to construct journal mapping by clustering. The merit of this approach is illustrated in a systematic evaluation with many comparing approaches and the proposed algorithm is shown outperforming over all other methods.

Chapter 12 summarizes the thesis and mentions several topics that worth further investigation.

1.5 Relevant research topics in ESAT-SCD, K.U.Leuven

This thesis is related to several research topics in SCD-SISTA, ESAT, K.U.Leuven, which is shown pictorially in Figure 1.8. Each research topic is represented by one or several recently-finished Ph.D dissertations.

- *Kernel fusion by convex optimization*

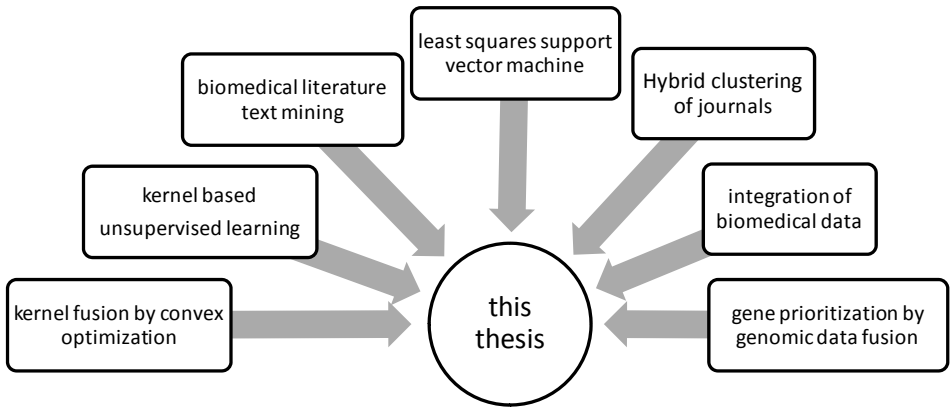


Figure 1.8: Overview of the connections of this thesis with other research topics in SCD-SISTA,ESAT,K.U.Leuven

De Bie T., Semi-supervised learning based on Kernel methods and Graph cut algorithms, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2005.

- *Kernel based unsupervised learning*

  Alzate C., Support Vector Methods for Unsupervised Learning, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2009.

- *Biomedical literature text mining*

  Glenisson P., Integrating scientific literature with large scale gene expression analysis, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2004.

  Van Vooren S., Data Mining for Molecular Karyotyping: Linked Analysis of Array-CGH Data and Biomedical Text, SCD-SISTA,ESAT, K.U.Leuven, 2009.

- *Support vector machines and Least squares support vector machines*

  Pelckmans K., Primal-Dual Kernel Machines, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2005.

- *Hybrid clustering of journals*

  Janssens F., Clustering of scientific fields by integrating text mining and bibliometrics, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2007.

- *Data fusion of Genetics, Molecular Biology, and Biomedical sources*

  Coessens B., Data Integration Techniques for Molecular Biology Research, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2006.

Gevaert O., A Bayesian network integration framework for modeling biomedical data, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2008.

- *Gene prioritization by genomic data fusion*

  Durinck S., Microarray compendia and their implications for bioinformatics software development, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2006.

  Van Loo P., Systems biology: identification of regulatory regions and disease causing genes and mechanisms, Ph.D thesis, SCD-SISTA,ESAT, K.U.Leuven, 2008.

1.6 Contributions of the thesis

Personal contributions

This thesis is mainly composed of original and independent work of the author in several aspects. Chapter 2 to Chapter 7, and Chapter 10 represent the author's personal contribution in kernel fusion theory, algorithmic design and innovation and they are all based on first-author publications. Though Chapter 8, Chapter 9 and Chapter 11 are based on collaborated work, the author contributed to 90% of the work mentioned in this thesis. In Chapter 8, the author collected the corpus data, investigated the bio-ontologies, performed text mining, designed and programmed all the algorithms and evaluated the performance. The set of disease benchmark genes and the biological interpretation of prioritization results are collaborated with the co-authors. In Chapter 11, the author applied the proposed method on the experimental data set, which is obtained through collaboration. The author performed the clustering process and performance evaluation. **In conclusion, 95% of the work presented in this thesis is based on the author's independent research and contributions.**

Main contributions

- **Efficient clustering algorithms based on LS-SVM kernel fusion.** We propose several novel algorithms to solve the novel and emerging problem about clustering using kernel fusion. From Chapter 2 to Chapter 4, we elaborate some fundamental issues about the mathematical models and the efficiency in large scale problems. The key observation is that the LS-SVM formulation of kernel fusion optimizes the Rayleigh quotient form, which is similar to the objectives of many unsupervised problems. Moreover, we show that the LS-SVM MKL is solved more efficiently in large problems.

In Chapter 5 and Chapter 6, we introduce the algorithms and demonstrate some relevant case studies. Chapter 2, Chapter 3, Chapter 4, and Chapter 5.

S. Yu, L.-C. Tranchevent, X. Liu, W. Glänzel, J. Suykens, B. De Moor and Y. Moreau, "Optimized data fusion for kernel K-means clustering", *Internal Report 08-200, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2008, Lirias number: 242275, submitted for publication.*

S. Yu, B. De Moor and Y. Moreau, "Clustering by heterogeneous data fusion: framework and applications", *in Workshop of Learning from Multiple Sources, NIPS 2008*, 2008.

- **Non-sparse L_2-norm kernel fusion.** Current kernel fusion methods mostly optimize the L_∞-norm of kernel fusion in terms of the dual problem. This method is characterized as the sparse solution, which assigns dominant coefficients on one or two kernels. The sparse solution is useful to distinguish the relevant sources from irrelevant ones. However, in biomedical applications, most of the data sources are well selected and processed, so they often have high relevance to the problem. In these cases, sparse solution may be too selective to thoroughly combine the complementary information in the data. Moreover, data sets in medical research are often collected at high cost, so the sparse solution discarding data sources is often less favorable and acceptable. To address this problem, we propose a new kernel fusion scheme to optimize the L_2-norm of kernel integration. The L_2-norm leads to non-sparse solution, which smoothly distributes the coefficients on multiple kernels, and at the same time, leverages the effects of kernels in the objective optimization. Empirical results show that the L_2-norm kernel fusion may lead to better performance in biomedical applications. Chapter 3 and Chapter 4.

  S. Yu, T. Falck, A. Daemen, J. Suykens, B. De Moor and Y. Moreau, "Non-sparse kernel fusion and its applications in genomic data integration", *Internal Report, K.U.Leuven (Leuven, Belgium), 2009, Lirias number: 248322, submitted for publication.*

- **Weighted multiple kernel canonical correlation analysis.** CCA is a classical statistical method to analysis relationship among multiple data sources. Recently, CCA has attracted lots of interests and leaded to interpretations in many aspects, i.e., dimensionality reduction, independent measure, correlation measure, etc. It has also been applied in many data fusion applications ranging from statistical analysis, information retrieval, to robotic vision. In this thesis, we generalizes CCA by adding additional weight coefficients on data sets. Chapter 7.

S.Yu, B. De Moor and Y. Moreau. "Learning with heterogeneous data sets by Weighted Multiple Kernel Canonical Correlation Analysis", *in Proc. of the IEEE Machine Learning for Signal Processing (IEEE MLSP) 2007*, 2007.

- **Multi-view text mining for gene prioritization and clustering.** Equipped by the workhorse of data fusion, we investigate a real application in biomedical text mining. We retrieve and combine the information in biomedical literature from multiple perspectives using multiple controlled vocabularies. We show that single text perspective is useful in biomedical knowledge discovery, however, a single CV text model is too fragile with respect to the uncertainty that a gene may not be annotated or indexed by the terms of a specific CV. Therefore, we propose a multi-view approach to combine multiple perspectives together and validate the approach on biomedical benchmark data. Chapter 8.

  S. Yu, S. Van Vooren, L.C. Tranchevent, B. De Moor, Y. Moreau, "Comparison of vocabularies, representations and ranking algorithms for gene prioritization by text mining", *Bioinformatics*, vol. 24, no. 16, pp. i119-125, 2008.

  S. Yu, L.-C. Tranchevent, B. De Moor, Y. Moreau, "Gene prioritization and clustering by multi-view text mining", *BMC Bioinformatics*, in publication, 2009.

- **Software of cross-species gene prioritization by genomic data fusion.** To practise kernel methods in real applications, we design and develop a kernel based software to solve real bioinformatics problems. Based on the main methodology proposed in the previous work [46], we create an open online software for cross-species gene prioritization. Chapter 9.

  S. Yu, L.-C. Tranchevent, S. Leach, R. Barriot, T. De Bie, B. De Moor, Y. Moreau, "Cross-species gene prioritization by genomic data fusion", *Internal Report (in preparation)*, 2009.

- **Combining lexical information and citation links in journal sets analysis.** We also apply our methods in a different domain as journal set analysis by hybrid clustering. Preliminary observations find that the accuracies of clustering and classification to map the scientific fields is enhanced by the combination of text mining information and bibliometrics [84]. We focus our efforts on two issues. Firstly, we present a "kernelized" approach for hybrid journal clustering. Secondly, we refine the hybrid clustering paradigm by distinguishing the attribute-based data from the interaction-based data, and combine them separately in different algorithmic models. Chapter 5, Chapter 6, Chapter 11, and Chapter 12.

X. Liu, S. Yu, Y. Moreau, W. Glänzel, B. De Moor, F. Janssens, "Hybrid Clustering of Text Mining and Bibliometrics Applied to Journal Sets", *in Proceedings of SIAM Data Mining conference 2009*, Sparks, Nevada, USA.

X. Liu, S. Yu, Y. Moreau, W. Glänzel, B. De Moor, F. Janssens, "Weighted Hybrid Clustering by Combining Text Mining and Bibliometrics on Large-Scale Journal Database", *Internal Report, submitted to the journal of American Information and Science Technology, in publication*, 2009.

X. Liu, S. Yu, Y. Moreau, W. Glänzel, B. De Moor, F. Janssens, "Hybrid Clustering by Integrating Text and Citation based Graphs in Journal Database Analysis", *Accepted for publication in ICDM-09 Workshop on Mining Multiple Information Sources*, 2009.

Novelty in the thesis

In particular, we highlight the novelty in this thesis as follows:

- **Three novel algorithms appear in this thesis have never been reported in the literature before:** The L_2-norm LS-SVM MKL classifier, the OKLC algorithm to incorporate multiple kernels and Laplacians in clustering, the WMKCCA algorithm to compute weighted multiple kernel canonical correlation. There is no other similar algorithm available.
- **The OKKC algorithm presented in this thesis provides more efficient solution than the existing methods.**
- **Multi-view text mining for disease gene prioritization and clustering:** Though pieces of similar works appear in the literature, our approach is characterized as one of the most comprehensive applications. It consists of several key components in the text mining pipeline and a systematic evaluation of existing methods for text mining model integration.
- **Cross-species genomic data fusion:** To combine evidences from multiple species to prioritize disease candidate genes is a new topic appears in recent two years. Hopefully, the software mentioned in this thesis will be the first real genome-wide bioinformatics application using kernel fusion techniques.
- **Multi-source and multi-structure hybrid clustering for journal set analysis:** For the first time, we optimize the combination of multiple kernels and Laplacians to cluster journal sets. The similar approach has not been reported before either in machine learning community or in scientometrics community.

Chapter 2

Rayleigh quotient-type problems in unsupervised learning

2.1 Optimization of Rayleigh quotient

Rayleigh quotient and its optimization

For real matrices and vectors, given a positive definite matrix Q and a nonzero vector $\vec{w}$, a Rayleigh quotient (also known as *Rayleigh-Ritz ratio*) is defined as

$$\rho = \rho(\vec{w}; Q) = \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T \vec{w}}. \tag{2.1}$$

It was originally proposed in the theorem to approximate eigenvalues of Hermitian matrix. The theorem is mentioned as as follows:

Theorem 2.1 (Rayleigh-Ritz). Let $Q \in M_n$ be Hermitian, and let the eigenvalues of Q be ordered as $\lambda_{min} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n-1} \leq \lambda_n = \lambda_{max}$. Then

$$\lambda_1 \vec{w}^T \vec{w} \leq \vec{w}^T Q \vec{w} \leq \lambda_n \vec{w}^T \vec{w} \quad \text{for all} \quad \vec{w} \in \mathbb{C}^n,$$

$$\lambda_{max} = \lambda_n = \max_{\vec{w} \neq 0} \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T \vec{w}} = \max_{\vec{w}^T \vec{w} = 1} \vec{w}^T Q \vec{w},$$

$$\lambda_{min} = \lambda_1 = \min_{\vec{w} \neq 0} \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T \vec{w}} = \min_{\vec{w}^T \vec{w} = 1} \vec{w}^T Q \vec{w}.$$

Proof: Omitted, available in e.g. See Theorem 4.2.2 in [76].

The Rayleigh quotient is used to calculate the eigenvalues of a matrix. As a matter of fact, many machine learning problems can be simplified as the minimization or maximization of the Rayleigh quotient as

$$\max_{\vec{w}} \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T \vec{w}}, \quad \text{or} \quad \min_{\vec{w}} \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T \vec{w}}. \tag{2.2}$$

Notice that the quotient is invariant to the magnitude of $\vec{w}$, one can reformulate the problem in terms of QCQP, given by

$$\begin{aligned} \max_{\vec{w}} \quad & \vec{w}^T Q \vec{w} \\ \text{s.t.} \quad & \vec{w}^T \vec{w} = 1. \end{aligned} \tag{2.3}$$

The Lagrangian $\mathcal{L}(\vec{w}, \lambda)$ is

$$\mathcal{L}(\vec{w}, \lambda) = \max_{\vec{w}, \lambda} \left(\vec{w}^T Q \vec{w} - \alpha(\vec{w}^T \vec{w} - 1) \right), \tag{2.4}$$

where $\alpha \in \mathbb{R}$ is the Lagrangian multiplier (dual variable). Taking the conditions for optimality from the Lagrangian $\partial \mathcal{L}/\partial \vec{w} = 0$, $\partial \mathcal{L}/\partial \alpha = 0$, (2.4) leads to the eigenvalue problem as

$$Q\vec{w} = \alpha \vec{w}. \tag{2.5}$$

Obviously, the optimal solution of the maximization problem in (2.2) is obtained by $\vec{w}_{max}$ in the maximal eigenvalue pair $(\alpha_{max}, \vec{w}_{max})$ of Q. Similarly, the solution of minimization is corresponding to $\vec{w}_{min}$ in the minimal eigenvalue pair $(\alpha_{min}, \vec{w}_{min})$.

Generalized Rayleigh quotient

The Generalized Rayleigh quotient involves two quadratic forms, given by

$$\rho = \rho(\vec{w}; Q, P) = \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T P \vec{w}}, \tag{2.6}$$

where Q and P are positive definite matrices, $\vec{w} \neq 0$. The problem is to maximize or minimize the Generalized Rayleigh quotient as follows:

$$\max_{\vec{w}} \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T P \vec{w}}, \quad \text{or} \quad \min_{\vec{w}} \frac{\vec{w}^T Q \vec{w}}{\vec{w}^T P \vec{w}}. \tag{2.7}$$

Taking the conditions for optimality from the Lagrangian, one obtains

$$Q\vec{w} = \alpha P\vec{w}. \tag{2.8}$$

Thus the optimum of (2.7) can be obtained by exploiting the eigenvalue pair of the generalized eigenvalue problem in (2.8).

Trace optimization of Generalized Rayleigh quotient-type problems

We now consider a more general problem than (2.7) by replacing $\vec{w}$ with a matrix $W = [\vec{w}_1, ..., \vec{w}_k]$. Firstly, we quote the Ky-Fan theorem as

Theorem 2.2 (Ky Fan). Let $Q \in \mathbb{R}^{n\times n}$ be a positive definite matrix, and let the eigenvalues of Q be ordered as $\lambda_{min} = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{n-1} \leq \lambda_n = \lambda_{max}$ and the corresponding eigenvectors as $\vec{u}_1, ..., \vec{u}_n$. Then

$$\lambda_1 + \cdots + \lambda_k = \max_{W^TW=I} \text{trace}(W^TQW). \tag{2.9}$$

Moreover, the optimal W^* is given by $W^* = [\vec{u}_1, ..., \vec{u}_k]U$, where U is an arbitrary orthogonal matrix.

Proof: Omitted, available in e.g., [19, 161].

To consider two positive definite matrices Q and P, one generalizes the Ky-Fan theorem as follows:

Theorem 2.3 Given a real value matrix $W = [\vec{w}_1, ..., \vec{w}_k] \in \mathbb{R}^{n\times k}$, and two positive definite matrices $Q, P \in \mathbb{R}^{n\times n}$, the optimal solution of the trace maximization problem as

$$\begin{aligned} &\max_{W} \ \text{trace}\frac{W^TQW}{W^TPW} \\ &\text{s.t.} \ \ W^TW = I \end{aligned} \tag{2.10}$$

is obtained by the k vectors corresponding to the largest k eigenvalue pairs solved from the generalized eigenvalue problem

$$Q\vec{\xi} = \lambda P\vec{\xi}. \tag{2.11}$$

Simplified Proof: Denote

$$F = P^{\frac{1}{2}}W, \tag{2.12}$$

then

$$F^T F = W^T P W. \tag{2.13}$$

Denote

$$P = U \Lambda U^T \tag{2.14}$$

as the eigenvalue decomposition of P where Λ is the diagonal matrix of the eigenvalues, given by

$$(F^T F)^{-\frac{1}{2}} = U \Lambda^{-\frac{1}{2}} U^T. \tag{2.15}$$

Since $W^T B W$ is symmetric, the quotient term in (2.10) can be equivalently rewritten as

$$\begin{aligned} \frac{W^T Q W}{W^T P W} &= (W^T P W)^{-\frac{1}{2}} (W^T Q W)(W^T P W)^{-\frac{1}{2}} \\ &= (F^T F)^{-\frac{1}{2}} (W^T Q W)(F^T F)^{-\frac{1}{2}} \\ &= (F^T F)^{-\frac{1}{2}} F^T P^{-\frac{1}{2}} Q P^{-\frac{1}{2}} F (F^T F)^{-\frac{1}{2}}. \end{aligned} \tag{2.16}$$

Let us denote that $B = F(F^T F)^{-\frac{1}{2}}$ and replace (2.15) in (2.16), moreover, since $B^T B = I$, thus (2.16) is equal to

$$\frac{W^T A W}{W^T B W} = B^T P^{-\frac{1}{2}} Q P^{-\frac{1}{2}} B. \tag{2.17}$$

According to the Ky-Fan, the optimal of maximizing (2.17) is given by the eigenvectors corresponding to the largest eigenvalues of $P^{-\frac{1}{2}} Q P^{-\frac{1}{2}}$, which is equivalent to the solution of the generalized eigenvalue problem in (2.9). □

The formal proof of Theorem 2.3 is given in [98, 129].

We have now introduced the basic formulations of Rayleigh quotient-type problems and their optimal solutions. Next, we will show that many unsupervised learning problems can be simplified to the Rayleigh quotient-type forms.

2.2 Rayleigh quotient-type problem in unsupervised learning

Principal Component Analysis (PCA)

PCA considers a given set of zero mean data $X \in \mathbb{R}^{N \times d}$, where $X = \{\vec{x}_i\}_{i=1}^N$, $\vec{x}_i \in \mathbb{R}^d$. The objective is to find a direction vector $\vec{w}$ on which the variance of the

projection $\vec{w}^T\vec{x}_i$ is maximized. Since the variance is invariant to the magnitude of $\vec{w}$, the objective of PCA is equivalently formulated as

$$\begin{aligned} \max_{\vec{w}} \quad & \vec{w}^T C_{xx}\vec{w} \\ \text{s.t.} \quad & \vec{w}^T\vec{w} = 1, \end{aligned} \tag{2.18}$$

where C_{xx} is the sample covariance matrix of X. Obviously, (2.18) is a Rayleigh quotient optimization and the optimal $\vec{w}$ is given by the eigenvector from the largest eigenvalue pair of C_{xx}.

CCA

CCA finds linear relations between two sets of variables [77]. For two zero mean data sets $X \in \mathbb{R}^{N\times d_1}$ and $Y \in \mathbb{R}^{N\times d_2}$, the objective is to identify vectors $\vec{w}_1$ and $\vec{w}_2$ such that the correlation between the projected variables $\vec{w}_1^T X$ and $\vec{w}_2^T Y$ is maximized, given by

$$\max_{\vec{w}_1,\vec{w}_2} \rho = \frac{\vec{w}_1^T C_{xy}\vec{w}_2}{\sqrt{\vec{w}_1^T C_{xx}\vec{w}_1}\sqrt{\vec{w}_2^T C_{yy}\vec{w}_2}}, \tag{2.19}$$

where $C_{xx} = \mathcal{E}[XX^T], C_{yy} = \mathcal{E}[YY^T], C_{xy} = \mathcal{E}[XY^T]$. The problem in (2.19) is usually formulated as the optimization problem, given by

$$\begin{aligned} \max_{\vec{w}_1,\vec{w}_2} \quad & \vec{w}_1^T C_{xy}\vec{w}_2 \\ \text{s.t.} \quad & \vec{w}_1^T C_{xx}\vec{w}_1 = 1, \\ & \vec{w}_2^T C_{yy}\vec{w}_2 = 1. \end{aligned} \tag{2.20}$$

Taking the conditions for optimality from the Lagrangian

$$\mathcal{L}(\vec{w}_1,\vec{w}_2;\lambda_1,\lambda_2) = \vec{w}_1^T C_{xy}\vec{w}_2 - \lambda_1(\vec{w}_1^T C_{xx}\vec{w}_1 - 1) - \lambda_2(\vec{w}_2^T C_{yy}\vec{w}_2 - 1), \tag{2.21}$$

one has

$$\begin{cases} C_{xy}\vec{w}_2 = \lambda_1 C_{xx}\vec{w}_1 \\ C_{yx}\vec{w}_1 = \lambda_2 C_{yy}\vec{w}_2 \end{cases}. \tag{2.22}$$

Since we have

$$\begin{cases} \vec{w}_1^T C_{xy}\vec{w}_2 = \lambda_1\vec{w}_1^T C_{xx}\vec{w}_1 \\ \vec{w}_2^T C_{yx}\vec{w}_1 = \lambda_2\vec{w}_2^T C_{yy}\vec{w}_2 \\ \vec{w}_1^T C_{xy}\vec{w}_2 = \vec{w}_2^T C_{yx}\vec{w}_1 \\ \vec{w}_1^T C_{xx}\vec{w}_1 = \vec{w}_2^T C_{yy}\vec{w}_2 \end{cases}, \tag{2.23}$$

we find that $\lambda_1 = \lambda_2 = \lambda$, thus we obtain a generalized eigenvalue problem, given by

$$\begin{bmatrix} 0 & C_{xy} \\ Cyx & 0 \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix} = \lambda \begin{bmatrix} C_{xx} & 0 \\ 0 & C_{yy} \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix}. \tag{2.24}$$

Analogously, the objective function of CCA can be also rewritten in a generalized Rayleigh quotient form as

$$\max_{\vec{w}_1, \vec{w}_2} \frac{\begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix}^T \begin{bmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix}}{\begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix}^T \begin{bmatrix} C_{xx} & 0 \\ 0 & C_{yy} \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix}}. \tag{2.25}$$

FDA

FDA optimizes the discriminating direction for classification, which is also expressed as a form similar to the Generalized Rayleigh quotient, where A is the measure of the separability of class (between class variance) and B is the measure of within class variance, given by

$$\max_{\vec{w}} \frac{\vec{w}^T S_B \vec{w}}{\vec{w}^T S_W \vec{w}}, \tag{2.26}$$

where

$$S_B = (\vec{\mu}_2 - \vec{\mu}_1)(\vec{\mu}_2 - \vec{\mu}_1)^T,$$

$$S_W = \sum_{i=1,2} \sum_{\vec{x} \in \mathcal{X}_i} (\vec{x} - \mu_i)(\vec{x} - \mu_i)^T,$$

and $\vec{\mu}_i$ denotes the sample mean for class i [123].

K-means clustering

It has been shown that, e.g., See [52], the principal components in PCA are equivalent to the continuous solutions of the cluster membership indicators in the K-means clustering method. K-means uses K-prototypes to characterize the data and the partitions are determined by minimizing the distortion as

$$J_{K-means} = \sum_{k=1}^{K} \sum_{i \in \mathcal{X}_i} (\vec{x}_i - \vec{\mu}_k)(\vec{x}_i - \vec{\mu}_k)^T. \tag{2.27}$$

For a given data set X and a cluster number K, the summation of all the pairwise distances is a constant value hence minimizing the distortion is equivalent to maximizing the *between clusters distortion*, given by

$$J_{K-means} = \sum_{k=1}^{K} (\vec{\mu}_k - \hat{\mu})(\vec{\mu}_k - \hat{\mu})^T. \tag{2.28}$$

where $\hat{\mu}$ is the global sample mean of X.

Let us denote P as the weighted cluster indicator matrix for k classes, given by

$$A = F(F^T F)^{-\frac{1}{2}}, \tag{2.29}$$

where F is the $n \times k$ binary cluster indicator matrix as

$$F = f_{i,j}{}_{n\times k}, \text{where } f_{i,j} = \begin{cases} 1 & \text{if } x_i \in l_j \\ 0 & \text{if } x_i \notin l_j \end{cases}. \tag{2.30}$$

Assume X has zero mean, without losing generality, (2.28) can be re-written in the matrix form as

$$J_{K-means} = \max_A \operatorname{trace}\left(A^T X^T X A\right). \tag{2.31}$$

Because the construction of P in (2.29) and (2.30) ensures that $P^T P = I$, the objective of K-means is exactly the maximization of a Rayleigh quotient. When $K = 2$, P reduces to vector $\vec{p}$, and leads to the PCA problem. When $K > 2$, the cluster indicators f_k can be recovered by exploiting the $K-1$ principal components of $X^T X$, for instance, by QR decomposition proposed in [198]. This PCA based approach of K-means clustering is also known as *the spectral relaxation* of K-means.

Spectral clustering

Spectral clustering models the data as graphs where the data samples are represented as vertices connected by non-negative weighted undirected edges. The clustering problem is then restated as to find a partition of the graph that the edges between different groups have a very low weight [181]. Different criteria have been applied to model the objective of cut, i.e., the RatioCut, the normalized cut, Markov Random Walks, the min-cut, etc. In this paper, the discussions about spectral clustering are all based on the normalized cut objective.

Let us denote $G = (V, E)$ as an undirected graph with vertex set $V = \{v_1, ..., v_n\}$, W as the weighted adjacency matrix of the graph $W = \{w_{ij}\}_{i,j=1,...,n}$, $d_i = \sum_{j=1}^{n} w_{ij}$ as the degree of a vertex $v_i \in V$, and D as the diagonal matrix with the

degrees $d_1, ..., d_n$ on the diagonal. Given a subset of vertices $\mathcal{X} \subset V$, we denote its complement $V \setminus \mathcal{X}$ as $\bar{\mathcal{X}}$. For two disjoint subsets $M, N \subset V$, the cut is defined as

$$\text{cut}(M, N) = \sum_{i \in M, j \in N} w_{ij} \quad . \tag{2.32}$$

The size of a subset is defined as

$$\text{vol}(M) = \sum_{i \in M} d_i. \tag{2.33}$$

The normalized cut criterion optimizes the partition $\mathcal{X}_1, \ldots, \mathcal{X}_k$ to minimize the objective as

$$\text{Ncut}(\mathcal{X}_1, \ldots, \mathcal{X}_k) = \sum_{i=1}^{k} \frac{\text{cut}(\mathcal{X}_i, \bar{\mathcal{X}_i})}{\text{vol}(\mathcal{X}_i)}. \tag{2.34}$$

Unfortunately, to obtain the exact solution of (2.34) is NP hard [182]. To solve it, the discrete constraint of clustering indicators is usually relaxed to real values thus the approximated solution of spectral clustering can be obtained from the eigenspectrum of the graph Laplacian matrix. For k-way clustering ($k > 2$), the weighted cluster indicator matrix P is defined in the same way as (2.29) and (2.30), the problem of minimizing the normalized cut is equivalently expressed as

$$\begin{aligned} &\min_{A} \ \text{trace}(A^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} A), \\ &\text{s.t.} \ \ A^T A = I. \end{aligned} \tag{2.35}$$

This is again the optimization of a Rayleigh quotient problem which can be solved by eigenvalue decomposition. The optimal P^* corresponds to the first k eigenvectors of the normalized Laplacian $\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$.

Kernel-Laplacian clustering (KL)

Let us assume that the attribute based data X and the graph affinity matrix W are representations of the same sets of samples, the objective function of Kernel-Laplacian clustering can be defined as

$$J_{KL} = \kappa J_{Ncut} + (1 - \kappa) J_{k-means} \tag{2.36}$$

where κ is the weight adjusting the effect of K-means and spectral clustering objectives. P is the weighted cluster indicator matrix as defined before. Replace (2.31) and (2.35) in (2.36), the objective of KL clustering becomes

$$\begin{aligned} J_{KL} \quad &= \kappa \min_{A} \text{trace}\left(A^T \tilde{L} A\right) + (1 - \kappa) \max_{A} \text{trace}\left(A^T X^T X A\right) \\ \text{s.t.} \ \ &A^T A = I. \end{aligned} \tag{2.37}$$

To solve the optimization problem without tuning the hyperparameter κ, Wang *et al.* propose a solution to optimize the trace quotient of the two sub-objectives [183]. The trace quotient formulation is then further relaxed as a maximization of the quotient trace, given by

$$\begin{aligned} J_{KL} &= \max_{A} \operatorname{trace}\left((A^T \tilde{L} A)^{-1}(A^T X^T X P)\right) \\ \text{s.t. } & A^T A = I. \end{aligned} \tag{2.38}$$

The objective in (2.38) is again a generalized Rayleigh quotient problem and the optimal solution P^* is obtained by solving the generalized eigenvalue problem. To maximize the objective with k clusters, P^* is approximated as the largest k eigenvectors of $\tilde{L}^+ \left(X^T X\right)$, where $\tilde{L}^+$ is the pseudo inverse of $\tilde{L}$.

One class support vector machine (1-SVM)

The 1-SVM method transforms the binary SVM classification task as one class learning problem. The method transforms the training data of one class into a high dimensional Hilbert space by the feature map, and iteratively finds the maximal margin hyper-plane that best separates the training data from the origin. The solution for the hyper-plane is found by solving the objective as follows:

$$\begin{aligned} &\min_{\vec{w},\xi,\rho} \left(\frac{1}{2}\vec{w}^T\vec{w} - \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \right) \\ &\text{s.t. } \vec{w}^T\phi(\vec{x}_i) \geq \rho - \xi_i, \quad i = 1, ..., n \\ &\qquad \xi_i \geq 0, \end{aligned} \tag{2.39}$$

where $\vec{w}$ is the vector perpendicular to the separating hyper-plane (norm vector), N is the number of training data, ρ is the bias value parameterizes the hyper plane, ν is a regularization variable penalizing the outliers in the training data, ξ_i are the slack variables. Taking the conditions for optimality from the Lagrangian as

$$L(\vec{w},\xi,\rho;\vec{\alpha},\vec{\beta}) = \frac{1}{2}\vec{w}^T\vec{w} + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho - \sum_{i=1}^{N}\alpha_i^T\left(w^T\phi(\vec{x}_i) - \rho + \xi_i\right) - \sum_{i=1}^{n}\beta_i^T\xi_i. \tag{2.40}$$

one obtains the dual problem as

$$\min_{\vec{\alpha}} \ \vec{\alpha}^T K \vec{\alpha} \tag{2.41}$$

$$\text{s.t.} \ \ 0 \leq \alpha_i \leq \frac{1}{\nu N},$$

$$\sum_{i=1}^{n} \alpha_i = 1,$$

where K is obtained by applying the kernel trick $K(x, y) = \phi(\vec{x})^T \phi(\vec{y})$, the dual problem of 1-SVM is again a problem similar to the Rayleigh quotient-type form. However, the optimal $\vec{\alpha}$ cannot be solved as the eigenvalue problem because the constraint $\vec{\alpha}^T \vec{\alpha} = 1$ does not hold in (2.41).

2.3 Summary

In this chapter we made a survey of several popular unsupervised learning algorithms. The main finding was that these problems can all be simplified as the Generalized Rayleigh quotient form, given by

$$\min_W \frac{W^T Q W}{W^T P W} \ \text{ or } \ \max_W \frac{W^T Q W}{W^T P W}. \tag{2.42}$$

In Table 2.1 we summarize the objectives, the mappings and the constraints corresponding of these algorithms in terms of Generalized Rayleigh quotient.

Table 2.1: Summary of several unsupervised algorithms and the mappings to the Generalized Rayleigh quotient form

algorithm	**objective**	**mapping**	**constraints**
PCA	$\max_{\vec{w}} \vec{w}^T C_{xx} \vec{w}$	$Q = C_{xx},\ P = I,\ W = \vec{w}$	$\vec{w}^T \vec{w} = 1$
CCA	$\max_{\vec{w}_1, \vec{w}_2} \frac{\begin{bmatrix}\vec{w}_1 \\ \vec{w}_2\end{bmatrix}^T \begin{bmatrix}0 & C_{xy} \\ C_{yx} & 0\end{bmatrix} \begin{bmatrix}\vec{w}_1 \\ \vec{w}_2\end{bmatrix}}{\begin{bmatrix}\vec{w}_1 \\ \vec{w}_2\end{bmatrix}^T \begin{bmatrix}C_{xx} & 0 \\ 0 & C_{yy}\end{bmatrix} \begin{bmatrix}\vec{w}_1 \\ \vec{w}_2\end{bmatrix}}$	$Q = \begin{bmatrix}0 & C_{xy} \\ C_{yx} & 0\end{bmatrix}, P = \begin{bmatrix}C_{xx} & 0 \\ 0 & C_{yy}\end{bmatrix}, W = \begin{bmatrix}\vec{w}_1 \\ \vec{w}_2\end{bmatrix}$	$\begin{bmatrix}\vec{w}_1 \\ \vec{w}_2\end{bmatrix}^T \begin{bmatrix}\vec{w}_1 \\ \vec{w}_2\end{bmatrix} = I$
FDA	$\max_{\vec{w}} \left(\frac{\vec{w}^T S_B \vec{w}}{\vec{w}^T S_W \vec{w}} \right)$	$Q = S_B,\ P = S_W, W = \vec{w}$	$\vec{w}^T \vec{w} = I$
K-means	$\max_A \text{trace}\left(A^T X^T X A\right)$	$Q = X^T X,\ P = I,\ W = A$	$A^T A = I$
Spectral Clustering	$\min_A \text{trace}(A^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} A)$	$Q = L,\ P = D,\ W = A$	$A^T A = I$
KL	$\max_A \text{trace}\left(\frac{A^T X^T X A}{A^T \tilde{L} A} \right)$	$Q = X^T X,\ P = L,\ A = P$	$A^T A = I$
1-SVM	$\min_{\vec{\alpha}} \vec{\alpha}^T K \vec{\alpha}$	$Q = K,\ P = I,\ A = \vec{\alpha}$	$\sum \alpha_i = 1$

Chapter 3

Kernel fusion for one class and multiclass problem

3.1 Introduction

Kernel based data fusion, denoted as *kernel fusion*, has been pioneered by Lanckriet *et al.* as a statistical learning framework for genomic data fusion [106] and it has been applied widely in various applications. In particular, kernel representation resolves the heterogeneities of genomic data sources by transforming different data structures (i.e., vectors, strings, trees, graphs, etc.) into kernel matrices. These kernels are combined in a single kernel, moreover, the coefficients of the kernels are leveraged adaptively to optimize the algorithmic objective. The notion of kernel fusion was originally proposed to solve classification problems in computational biology, but current efforts have also invented analogous solutions in one class and unsupervised learning problems. Most of the existing kernel fusion methods are based on the formulation of Lanckriet *et al.*, which is represented in this paper as the optimization of the infinite norm (L_∞) of kernel fusion in the dual problem. The L_∞-norm of the dual problem is corresponds to L_1 regularization in the primal problem characterized by sparseness in the kernel coefficients [135]. The sparse solution assigns dominant coefficients to only one or two kernels, which is useful to distinguish relevant from irrelevant data sources. However, in biomedical applications, most of the data sources are carefully selected and preprocessed so they often have strong relevance to the problem. In these cases, sparse solution may be too selective to thoroughly combine the complementary information in the data sources. Moreover, data sets in medical research are often collected at high cost, so the sparse solution discarding data sources is often less favorable and acceptable. To address this problem, we propose a new kernel fusion scheme

by optimizing the L_2-norm of multiple kernels. The L_2-norm leads to non-sparse solution, which smoothly distributes the coefficients on multiple kernels and, at the same time, leverages the effects of kernels in the objective optimization. Empirical results show that the L_2-norm kernel fusion may lead to better performance in genomic data fusion.

The issue of using different norms in kernel fusion is a recent topic, e.g., see the work of Kloft *et al.* [97] and Kowalski *et al.* [99]. The main feature of this paper is in terms of the dual space solution, which differs from considering the norm of the coefficient regularization term in the primal space. Clarifying the dual problem enabled us to directly solve the L_2-norm problem as a convex SOCP directly. Moreover, as one of the novel contributions, we introduced L_2-norm LS-SVM classifier in the MKL context. Empirical results on real applications and numerical experiments on large scale data have shown that the L_2-norm LS-SVM is an efficient and promising approach for genome-wide data fusion applications.

3.2 Problem Definition

We consider the problem of minimizing the quadratic term of a real vector in function of $\vec{\alpha}$ and a real positive semi-definite (PSD) matrix Q

$$\min_{\vec{\alpha}} \ \vec{\alpha}^T Q \vec{\alpha} \tag{3.1}$$

$$\text{s.t.} \ \ Q \succeq 0,$$

where the Positive Semi-definite (PSD) property of Q is set in the constraint. Also, PSD implies that $\forall \vec{\alpha}$, $\vec{\alpha}^T Q \vec{\alpha} \geq 0$. We will show that many machine learning problems simplify to the form (3.1) with additional constraints on $\vec{\alpha}$. In particular, if we restrict that $\vec{\alpha}^T \vec{\alpha}$ to 1, the problem becomes a Rayleigh quotient and leads to the eigenvalue problem.

Now we consider a convex parametric linear combination of a set of p number of PSD matrices Q_j, given by:

$$\Omega = \left\{ \sum_{j=1}^{p} \theta_j Q_j \middle| \forall j, \ \theta_j \geq 0, \ Q_j \succeq 0 \right\}. \tag{3.2}$$

To bound the coefficients θ_j, we restrict that, for example, $||\theta_j||_1 = 1$, and (3.1) is equivalently rewritten as

$$\min_{\vec{\alpha},\theta} \ \vec{\alpha}^T \left(\sum_{j=1}^{p} \theta_j Q_j \right) \vec{\alpha} \tag{3.3}$$

$$\text{s.t.} \ \ Q_j \succeq 0, \quad j = 1, ..., p$$

$$\theta_j \geq 0, \quad j = 1, ..., p$$

$$\sum_{j=1}^{p} \theta_j = 1.$$

To solve (3.3), one can formulate it as a quadratic constrained linear programming (QCLP), given by:

$$\min_{\vec{\alpha},t} \ \ t \tag{3.4}$$

$$\text{s.t.} \ \ Q_j \succeq 0, \quad j = 1, ..., p$$

$$t \geq \vec{\alpha}^T Q_j \vec{\alpha}, \quad j = 1, ..., p.$$

The optimal solution θ^* in (3.4) is obtained from the dual variable corresponding to the quadratic constraints. The optimal t^* is equivalent to the *Chebyshev* or L_∞ norm of the vector of quadratic terms, given by:

$$t^* = ||\vec{\alpha}^T Q_j \vec{\alpha}||_\infty = \max\{\alpha^T Q_1 \vec{\alpha}, ..., \alpha^T Q_p \vec{\alpha}\}. \tag{3.5}$$

The L_∞ norm is the upper bound w.r.t. the constraint $\sum_{j=1}^{p} \theta_j = 1$ because

$$\sum_{j=1}^{p} \theta_j Q_j \leq t^*. \tag{3.6}$$

Apparently, suppose the optimal $\vec{\alpha}^*$ is given, optimizing the L_∞ norm in (3.5) will pick the single term with the maximal value, and the optimal solution of the coefficients is more likely to be sparse. An alternative solution to (3.3) is to introduce a different constraint on the coefficients, for example, $||\theta_j||_2 = 1$, thus

we propose a new extension of the problem in (3.1), given by:

$$\min_{\vec{\alpha},\theta} \ \vec{\alpha}^T \left(\sum_{j=1}^{p} \theta_j Q_j \right) \vec{\alpha} \tag{3.7}$$

$$\text{s.t.} \ \ Q_j \succeq 0, \quad j = 1, ..., p$$

$$\theta_j \geq 0, \quad j = 1, ..., p$$

$$||\theta_j||_2 = 1.$$

This new extension is analogously solved as a Quadratic Constrained Linear Programming (QCLP) problem, given by:

$$\min_{\vec{\alpha},\eta} \ \eta \tag{3.8}$$

$$\text{s.t.} \ \ Q_j \succeq 0, \quad j = 1, ..., p$$

$$s_j \geq \vec{\alpha}^T Q_j \vec{\alpha}, \quad j = 1, ..., p$$

$$\eta \geq ||s_j||_2, \quad j = 1, ..., p.$$

The proof that the QCLP problem in (3.8) is the solution of (3.7) is given in the following theorem.

Theorem 3.1 (L_2-norm extension of MKL) The QCLP problem in (3.8) equivalently solves the problem in (3.7)

Proof: The Cauchy-Schwarz inequality states that

$$0 \leq \left(\sum_{j=1}^{p} x_j y_j \right)^2 \leq \sum_{j=1}^{p} x_j^2 \sum_{j=1}^{p} y_j^2, \tag{3.9}$$

one has the following equivalent form:

$$0 \leq \left[\left(\sum_{j=1}^{p} x_j y_j \right)^2 \right]^{\frac{1}{2}} \leq \left[\sum_{j=1}^{p} x_j^2 \sum_{j=1}^{p} y_j^2 \right]^{\frac{1}{2}}. \tag{3.10}$$

Denoting $x_j = \theta_j$ and $y_j = \vec{\alpha}^T Q_j \vec{\alpha}$, (3.10) becomes

$$0 \leq \sum_{j=1}^{p} \left(\theta_j \vec{\alpha}^T Q_j \vec{\alpha} \right) \leq \left[\sum_{j=1}^{p} \theta_j^2 \sum_{j=1}^{p} \left(\vec{\alpha}^T Q_j \vec{\alpha} \right)^2 \right]^{\frac{1}{2}}. \tag{3.11}$$

Since $||\theta_j||_2 = 1$, (2.11) is equivalent to

$$0 \leq \sum_{j=1}^{p} \left(\theta_j \vec{\alpha}^T Q_j \vec{\alpha}\right) \leq \left[\sum_{j=1}^{p} \left(\vec{\alpha}^T Q_j \vec{\alpha}\right)^2\right]^{\frac{1}{2}}. \tag{3.12}$$

Therefore, the additive term $\sum_{j=1}^{p} \left(\theta_j \vec{\alpha}^T Q_j \vec{\alpha}\right)$ is bounded by the L_2 norm $||\vec{\alpha}^T Q_j \vec{\alpha}||_2$. □

Moreover, it is easy to prove that when $\theta_j^* = \vec{\alpha}^T Q_j \vec{\alpha} / ||\vec{\alpha}^T Q_j \vec{\alpha}||_2$, the parametric combination reaches the upperbound and the equality holds. Apparently, optimizing this L_2-norm results in a non-sparse solution in θ_j. In order to distinguish this from the solution obtained by (3.3) and (3.4), we denote it as the L_2-norm approach. It can also easily be seen (not shown here) that the L_1-norm approach is simply averaging the quadratic terms with uniform coefficients.

In this section, we have explained the L_∞, L_1, and L_2-norm approaches to extend the basic problem in (2.1) to multiple matrices Q_j. These approaches differed mainly in constraints applied on the coefficients. To clarify the difference of notations used in this paper with the common interpretations about L_1 and L_2 regularization, we illustrate the mapping of our L_∞, L_1, and L_2 notations between the common interpretations of coefficient regularization. As shown in Table 3.1, the notations used in this paper are interpreted in the dual space and are equivalent to regularization of kernel coefficients in the primal space. The dual space interpretation enables us to derive the solution of the optimization problem, moreover, it becomes more easier to extend the analogue solution to various machine learning algorithms. To keep the discussion concise, we will from now on mainly focus on the dual problems and present the solutions in the dual space.

Table 3.1: The notations used in this paper are based on the dual problem and they are linked to the equivalent notations in the primal problem

	primal problem	dual problem
variable	θ_j	$\vec{\alpha}^T K_j \vec{\alpha}$
$\mathbf{L_\infty}$	$\lvert\theta_j\rvert = 1$	$\max \lvert\lvert\vec{\alpha}^T K_j \vec{\alpha}\rvert\rvert_\infty$
$\mathbf{L_1}$	$\theta_j = \bar{\theta}$	$\max \lvert\lvert\vec{\alpha}^T K_j \vec{\alpha}\rvert\rvert_1$
$\mathbf{L_2}$	$\lvert\lvert\theta_j\rvert\rvert_2 = 1$	$\max \lvert\lvert\vec{\alpha}^T K_j \vec{\alpha}\rvert\rvert_2$

Next, we will investigate several concrete kernel fusion algorithms and will derive the corresponding non-sparse solutions.

3.3 One class SVM kernel fusion for ranking

The algorithm of one class SVM (1-SVM) is defined by Tax and Duin [168] and Schölkopf *et al.* [151] as

$$\min_{\vec{w},\xi,\rho} \ \frac{1}{2}\vec{w}^T\vec{w} - \frac{1}{\nu l}\sum_{k=1}^{l}\xi_k - \rho \tag{3.13}$$

$$\text{s.t.} \ \ \vec{w}^T\phi(\vec{x}_k) \geq \rho - \xi_k, \ k = 1, \ldots, N$$

$$\xi_k \geq 0, \ k = 1, ..., N.$$

where $\vec{w}$ is the norm vector of the separating hyperplane, $\vec{x}_k$ are the training samples, ν is the regularization term penalizing the outliers in the training samples, $\phi(\cdot)$ denotes the feature map, ρ is the bias term, ξ_k are the slack variables, and N is the number of training samples. Taking the conditions for optimality from the Lagrangian, one obtains the dual problem, given by:

$$\min_{\vec{\alpha}} \ \vec{\alpha}^T K \vec{\alpha} \tag{3.14}$$

$$\text{s.t.} \ \ 0 \leq \alpha_k \leq \frac{1}{\nu N}, \quad k = 1, ..., N$$

$$\sum_{k=1}^{N}\alpha_k = 1,$$

where α_k are dual variables, K represents the kernel matrix obtained by the inner product between any pair of samples specified by a kernel function $K(\vec{x}_k, \vec{x}_l)$. To incorporate multiple kernels in (3.14), De Bie *et al.* proposed a solution [46] with the dual problem formulated as

$$\min_{\vec{\alpha}} \ \ t \tag{3.15}$$

$$\text{s.t.} \ \ t \geq \vec{\alpha}^T K_j \vec{\alpha}, \ j = 1, ..., p$$

$$0 \leq \alpha_k \leq \frac{1}{\nu N}, \quad k = 1, ..., N$$

$$\sum_{k=1}^{N}\alpha_k = 1,$$

where p is the number of data sources and K_j is the j-th kernel matrix. The formulation exactly corresponds to the L_∞-norm solution of the problem defined in the previous section (the PSD constraint is implied in the kernel matrix) with

additional constraints imposed on $\vec{\alpha}$. The optimal coefficients θ_j are used to combine multiple kernels as

$$\Omega = \left\{ \sum_{j=1}^{p} \theta_j K_j \middle| \sum_{j=1}^{p} \theta_j = 1, \ \ \forall j, \ \theta_j \geq 0 \right\}, \tag{3.16}$$

and the ranking function is given by

$$f(\vec{y}) = \frac{1}{\sqrt{\vec{\alpha}^T \Omega(\vec{x}_k, \vec{x}_k)\vec{\alpha}}} \sum_{k=1}^{N} \alpha_k \Omega(\vec{x}_l, \vec{x}_k), \tag{3.17}$$

where $\{\vec{x}_k\}_{k=1}^{N}$ are the training samples and $\vec{x}_l$ is the test sample to be scored. De Bie *et al.* applied the method in the application of disease gene prioritization, where multiple genomic data sources are combined to rank a large set of test genes using the 1-SVM model trained from a small set of training genes which are known to be relevant for certain diseases. The L_∞-norm formulation in their approach results in a sparse solution when integrating genomic data sources (see Figure 2 of [46]). To avoid this disadvantage, they proposed a regularization method by restricting the minimal boundary on the kernel coefficients, notated as θ_{min}, to ensure the minimal contribution of each genomic data source to be θ_{min}/p. According to their experiments, the regularized solution performed the best, being significantly better than the sparse integration and the average combination of kernels.

Instead of setting the ad hoc parameter θ_{min}, one can also straightforwardly propose an L_2-norm approach to solve the identical problem, given by

$$\begin{aligned} \min_{\vec{\alpha}} \ \ & t \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.18) \\ \text{s.t.} \ \ & t \geq ||s_j||_2, \ j = 1, ..., p \\ & s_j \geq \vec{\alpha}^T K_j \vec{\alpha}, \ j = 1, ..., p \\ & 0 \leq \alpha_k \leq \frac{1}{\nu N}, \ k = 1, ..., N \\ & \sum_{k=1}^{N} \alpha_k = 1. \end{aligned}$$

The problem above is a QCLP problem and can be solved by conic optimization solvers such as Sedumi [164]. In (3.18), the first constraint represents a Q cone and the second constraint corresponds to p number of R cones when the kernel matrices are decomposed as $K_j = C_j C_j^T$, such that $s_j \geq ||\vec{\alpha}^T C_j||_2^2$. The optimal kernel coefficients θ_j correspond to the dual variables of the R cones with $||\theta_j||_2 = 1$.

In this L_2-norm approach, the integrated kernel Ω is combined by different θ_j^* and the same scoring function as in (3.17) is applied on the different solutions of $\vec{\alpha}$ and Ω.

3.4 Kernel fusion for classification

3.4.1 Support Vector Machine

The notion of kernel fusion is originally proposed in binary SVM classification, where the primal objective of SVM [24] is

$$\begin{aligned} \min_{\vec{w},b,\xi} \quad & \frac{1}{2}\vec{w}^T\vec{w} + \lambda\sum_{k=1}^{N}\xi_k \\ \text{s.t. } & y_k[\vec{w}^T\phi(\vec{x}_k) + b] \geq 1 - \xi_k,\ k = 1, ..., N \\ & \xi_k \geq 0,\ k = 1, ..., N, \end{aligned} \tag{3.19}$$

where $\vec{x}_k$ are data samples, $\phi(\cdot)$ is the feature map, y_k are class labels, $\lambda > 0$ is a positive regularization parameter, ξ_k are slack variables, $\vec{w}$ is the norm vector of the separating hyperplane, and b is the bias. This problem is convex and can be solved as a dual problem, given by

$$\begin{aligned} \min_{\vec{\alpha}} \quad & \frac{1}{2}\vec{\alpha}^T Y K Y \vec{\alpha} - \vec{\alpha}^T\vec{1} \\ \text{s.t. } & (Y\vec{\alpha})^T\vec{1} = 0, \\ & 0 \leq \alpha_k \leq C, \quad k = 1, ..., N, \end{aligned} \tag{3.20}$$

where $\vec{\alpha}$ are the dual variables, $Y = diag(y_1, ..., y_N)$, K is the kernel matrix, and C is the upperbound of the box constraint on dual variables. To incorporate multiple kernels in (3.20), Lanckriet *et al.* [106] and Bach *et al.* [13] proposed the multiple kernel learning (MKL) problem as follows:

$$\begin{aligned} \min_{\gamma,\vec{\alpha}} \quad & \frac{1}{2}\gamma - \vec{\alpha}^T\vec{1} \\ \text{s.t. } & (Y\vec{\alpha})^T\vec{1} = 0, \\ & 0 \leq \alpha_k \leq C,\ k = 1, ..., N \\ & \gamma \geq \vec{\alpha}^T Y K_j Y \vec{\alpha}, \quad j = 1, ..., p, \end{aligned} \tag{3.21}$$

where p is the number of kernels. Apparently, (3.21) optimizes the L_∞-norm of the set of kernel quadratic terms. Based on previous discussions, the L_2-norm solution is analogously given by

$$\begin{aligned} \min_{\eta,\vec{\alpha}} \quad & \frac{1}{2}\eta - \vec{\alpha}^T\vec{1} \\ \text{s.t.} \quad & (Y\vec{\alpha})^T\vec{1} = 0, \\ & 0 \le \alpha_k \le C, \quad k = 1, ..., N \\ & \eta \ge ||\gamma_j||_2, \quad j = 1, ..., p \\ & \gamma_j \ge \vec{\alpha}^T Y K_j Y \vec{\alpha}, \quad j = 1, ..., p. \end{aligned} \tag{3.22}$$

Both formulations in (4.21) and (4.22) can be efficiently solved as second order cone programming (SOCP) problems by a conic optimization solver (i.e., Sedumi) or as QCQP problems by a general QP solver (i.e., MOSEK [8]). It is also known that a binary MKL problem can be formulated as Semi-definite Programming (SDP), as proposed by Lanckriet *et al.* [104] and Kim *et al.* [95]. However, SDP is computationally prohibitive in multi-class problems due to the presence of PSD constraints and can only be solved approximately by relaxation [191]. On the contrary, the QCLP and QCQP formulations of binary classification problems can be easily extended to a multi-class case using the one-versus-all (1vsA) coding, i.e., solving the problem of q classes as q numbers of binary problems. The L_∞-norm solution for a multi-class problem is then formulated as follows:

$$\begin{aligned} \min_{\rho,\vec{\alpha}} \quad & \frac{1}{2}\rho - \sum_{i=1}^{q} \vec{\alpha}_i^T\vec{1} \\ \text{s.t.} \quad & (Y_i\vec{\alpha}_i)^T\vec{1} = 0, \quad i = 1, ..., q \\ & 0 \le \alpha_{ki} \le C, \quad k = 1, ..., N, \ i = 1, ..., q \\ & \rho \ge \sum_{i=1}^{q} (\vec{\alpha}_i^T Y_i K_j Y_i \vec{\alpha}_i), \quad j = 1, ..., p. \end{aligned} \tag{3.23}$$

The L_2-norm solution is given by

$$\min_{\delta,\vec{\alpha}} \ \frac{1}{2}\delta - \sum_{i=1}^{q} \vec{\alpha}_i^T \vec{1} \tag{3.24}$$

$$\text{s.t. } (Y_i\vec{\alpha}_i)^T\vec{1} = 0, \quad i = 1, ..., q$$

$$0 \leq \alpha_{ki} \leq C, \quad k = 1, ..., N, \quad i = 1, ..., k$$

$$\rho_j \geq \sum_{i=1}^{q} (\vec{\alpha}_i^T Y_i K_j Y_i \vec{\alpha}_i), \quad j = 1, ..., p$$

$$\delta \geq ||\rho_j||_2.$$

3.4.2 Least Squares Support Vector Machine.

In LS-SVM, the cost function of the classification error is replaced by a least squares term and the inequality constraints are replaced as equality constraints [166], given by

$$\min_{\vec{w},b,\vec{e}} \ \frac{1}{2}\vec{w}^T\vec{w} + \frac{1}{2}\lambda\vec{e}^T\vec{e} \tag{3.25}$$

$$\text{s.t. } y_k[\vec{w}^T\phi(\vec{x}_k) + b] = 1 - e_k, \ k = 1, ..., N,$$

where most of the variables are defined in the similar way as (3.19). This problem is also convex, moreover, as a unconstrained quadratic programming problem it has an analytical solution. Taking the conditions for optimality from the Lagrangian, eliminating $\vec{w}, \vec{e}$, defining $\vec{y} = [y_1, ..., y_N]^T$ and $Y = diag(y_1, ..., y_N)$, one obtains the following linear system [166]:

$$\left[\begin{array}{c|c} 0 & \vec{y}^T \\ \hline \vec{y} & YKY + I/\lambda \end{array}\right] \left[\begin{array}{c} b \\ \hline \vec{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \vec{1} \end{array}\right], \tag{3.26}$$

where $\vec{\alpha}$ are unconstrained dual variables. Without loss of generality, we denote $\vec{\beta} = Y\vec{\alpha}$ and rewrite (3.26) as

$$\left[\begin{array}{c|c} 0 & \vec{1}^T \\ \hline \vec{1} & K + Y^{-2}/\lambda \end{array}\right] \left[\begin{array}{c} b \\ \hline \vec{\beta} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y^{-1}\vec{1} \end{array}\right]. \tag{3.27}$$

In (3.27), we add an additional constraint as $Y^{-2} = I$ then the coefficient becomes a static value in the multi-class case. In 1vsA coding, solving (3.26) requires solving k number of linear problems whereas in (3.27), the coefficient matrix is only factorized once such that the solution of $\vec{\beta}_i$ w.r.t. the multi-class label vectors

$\vec{y}_i$ is very efficient to obtain. The constraint $Y^{-2} = I$ can be simply satisfied by assuming the class labels to be -1 and +1. Thus, from now on, we assume $Y^{-2} = I$ in the discussion.

To incorporate multiple kernels in LS-SVM classification, the L_∞-norm approach is a QP problem, given by (assuming $Y^{-2} = I$)

$$\begin{aligned} \min_{\vec{\alpha},t} \quad & \frac{1}{2}t + \frac{1}{2\lambda}\vec{\beta}^T\vec{\beta} - \vec{\beta}^T Y^{-1}\vec{1} \\ \text{s.t.} \quad & \sum_{k=1}^{N} \beta_k = 0, \\ & t \geq \vec{\beta}^T K_j \vec{\beta}, \ j = 1, ..., p. \end{aligned} \tag{3.28}$$

The L_2-norm approach is analogously formulated as

$$\begin{aligned} \min_{\vec{\alpha},\eta} \quad & \frac{1}{2}\eta + \frac{1}{2\lambda}\vec{\beta}^T\vec{\beta} - \vec{\beta}^T Y^{-1}\vec{1} \\ \text{s.t.} \quad & \sum_{k=1}^{N} \beta_k = 0, \\ & s_j \geq \vec{\beta}^T K_j \vec{\beta}, \ j = 1, ..., p, \\ & \eta \geq ||s_j||_2, \ j = 1, ..., p. \end{aligned} \tag{3.29}$$

The λ parameter regularizes the squared error term in the primal objective in (3.25) and the quadratic term $\vec{\beta}^T\vec{\beta}$ in the dual problem. Usually, the optimal λ needs to be selected empirically by cross-validation. In the kernel fusion of LS-SVM, we can alternatively transform the effect of regularization as an identity kernel matrix in $\frac{1}{2}\vec{\beta}^T\left(\sum_{j=1}^{p} K_j + \theta_{p+1}I\right)\vec{\beta}$, where $\theta_{p+1} = 1/\lambda$. Then the MKL problem of combining p kernels is equivalent to combining $p+1$ kernels where the last kernel is an identity matrix with the optimal coefficient corresponding to the λ value. This method has been mentioned by Lanckriet *et al.* to tackle the estimation of the regularization parameter in the soft margin SVM [104]. It has also been used by Ye *et al.* to jointly estimate the optimal kernel for discriminant analysis [191]. Saving the effort of validating λ may significantly reduce the model selection cost in complicated learning problems. By this transformation, the objective of LS-SVM MKL becomes similar to that of SVM MKL with as main difference that the dual variables are unconstrained. Though (3.28) and (3.29) can in principle both be solved as QP problems by a conic solver or a QP solver, the efficiency of a linear solution of the LS-SVM is lost.

3.5 Case studies of genomic data fusion

In this section we present two real applications to combine multiple genomic data sources. We focus the discussion on comparing the effect of sparse kernel fusion with the proposed non-sparse approach.

3.5.1 Disease relevant gene prioritization by genomic data fusion

The computational definition of gene prioritization is mentioned in our earlier work [3][46][196]. We applied the 1-SVM MKL to combine kernels derived from 9 heterogeneous genomic sources to prioritize 620 genes which are known to be relevant for 22 diseases. The performance was evaluated by Leave-one-out (LOO) validation: For each disease which contains K relevant genes, one gene, termed the "defector" gene, was deleted from the set of training genes and added to 99 randomly selected test genes (test set). We used the remaining $K-1$ genes (training set) to build our prioritization model. Then, we prioritized the test set of 100 genes with the trained model and determined the ranking of that defector gene in test data. The prioritization function in (3.17) scores the more relevant genes higher and others lower, thus, by labeling the "defector" gene as class "+1" and the random candidate genes as class "-1", we could plot the Receiver operating characteristic (ROC) curves to compare different models by the values of area under the ROC curve (AUC).

We compared four different 1-SVM MKL approaches for ranking: L_∞, L_1, L_2-norm and the regularized L_∞-norm approach by setting the minimal coefficient θ_{min} to 0.5, notated as $L_\infty(0.5)$. To make the problem simple and obvious, the kernels were all constructed with linear functions (except sequence data where 2-mer string kernel was applied) and in total nine kernel matrices were combined. The regularization parameter ν was set to 0.5. As there is no hyper-parameter that needs to be tuned in LOO validation, we report the LOO results as the performance of generalization. For each disease relevant gene, the 99 test genes were randomly selected in each LOO validation from the whole human protein-coding genome. We repeated the experiment 20 times and the performance is shown in Table 3.2. The L_2-norm approach performed best. The sparse L_∞ and the naive L_1 approaches performed not as good as the L_2-norm one. Though the regularized kernel fusion $L_\infty(0.5)$ improved the performance of sparse solution from 0.9045 to 0.9176, the free parameter θ_{min} was selected arbitrarily in this problem and the performance was not comparable to the non-sparse L_2 approach.

In Figure 3.1 we illustrate the average kernel coefficients determined by different approaches in one repetition. The L_∞-norm approach assigned dominant coefficients to Text mining and Gene Ontology data, whereas other data sources were almost discarded from integration. In contrast, the L_2-norm approach

Table 3.2: AUC values of LOO performance evaluated from 20 random repetitions. The paired Spearman correlation scores indicate the similarities of rankings obtained by different approaches compared with the target rankings (denoted as -).

	AUC	corr	corr	corr	corr
L_∞	0.9045(0.0043)	-	0.94	0.66	0.82
$L_\infty(0.5)$	0.9176(0.0040)	0.94	-	0.82	0.92
L_1	0.9103(0.0035)	0.66	0.82	-	0.90
L_2	**0.9219(0.0034)**	0.82	0.92	0.90	-

smoothly distributed the coefficients over all data sources and thoroughly combined them in integration. The performance of sparse kernel integration relied only on one or two kernels, making the combined model fragile with respect to the uncertainty (the relevance of a gene to a certain disease is not mentioned in Text mining and GO annotation). The non-sparse integration showed the same effect as regularized integration (as automatic estimated minimal boundaries for each data source), with the main advantage that there is no need to tune the regularization parameter θ_{min} which is highly dependent on problems.

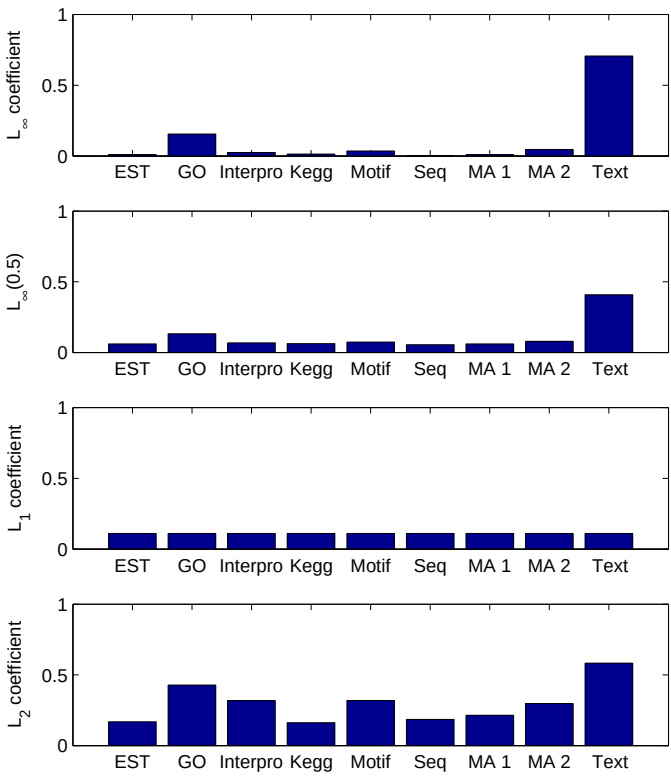


Figure 3.1: Kernel coefficients learned by different kernel fusion approaches. Nine data sources are combined: Annotation data of expressed sequence tagging (EST), annotation data of Gene Ontology (GO), annotation data of Interpro (Interpro), annotation data of KEGG (KEGG), annotation data of Motif findings (Motif), string data of Sequence (Seq), two microarray expression data sets (MA1 and MA2), and vector based text mining data (Text).

3.5.2 Clinical decision support by integrating microarray and proteomics data

The second case study is taken from the kernel-based integration of genome-wide data for clinical decision support in cancer diagnosis [43]. Thirty-six patients with rectal cancer were treated by combination of cetuximab, capecitabine and external beam radiotherapy and their tissue and plasma samples were gathered at three time points: before treatment (T_0); at the early therapy treatment (T_1) and at the moment of surgery (T_2). The tissue samples were hybridized to gene chip arrays and after processing, the expression was reduced to 6,913 genes. Ninety-six proteins known to be involved in cancer were measured in the plasma samples, and the ones that had absolute values above the detection limit in less than 20% of the samples were excluded for each time point separately. This resulted in the exclusion of six proteins at T_0 and four at T_1. "Responders" were distinguished from "non-responders" according to the pathologic lymph node stage at surgery (pN-STAGE). The "responder" class contains 22 patients with no lymph node found at surgery whereas the "non-responder" class contains 14 patients with at least 1 regional lymph node. Only the two array-expression data sets (MA) measured at T_0 and T_1 and the two proteomics data sets (PT) measured at T_0 and T_1 were used to predict the outcome of cancer at surgery.

Similar to the original method applied on the data [43], we used R BioConductor DEDS as feature selection techniques for microarray data and the Wilcoxon rank sum test for proteomics data. The statistical feature selection procedure was independent to the classification procedure, however, the performance varied widely with the number of selected genes and proteins. We considered the relevance of features (genes and proteins) as prior knowledge and systematically evaluated the performance using multiple numbers of genes and proteins. According to the ranking of statistical feature selection, we gradually increased the number of genes and proteins from 11 to 36, and combined the linear kernels constructed by these features. The performance was evaluated by LOO method, where the reason was two folded: Firstly, the number of samples was small (36 patients); Secondly, the kernels were all construct by linear function, moreover, in LS-SVM classification we proposed the strategy to estimate the regularization parameter λ in kernel fusion. Therefore, no hyperparameter was needed to be tuned so we reported the LOO validation result as the performance of generalization.

We applied L_∞, L_1, and L_2-norm kernel fusion in SVM and LS-SVM formulations to classify the patients according to lymph node stage. Since the performance of the classifier was greatly dependent on the selected features, for each classifier we compared 25 results (a grid of 5 numbers of genes multiplied by 5 numbers of proteins). As shown in Table 3.3, the best performance was obtained by L_1-norm LS-SVM (AUC=0.9675) using 25 genes and 15 proteins. The L_2-norm LS-SVM classifier also performed well and was comparable to the best result.

In particular, for the two compared classifiers (LS-SVM and SVM), the L_1 and L_2-norm approaches all outperformed the L_∞-norm approach, which also substantiates the advantage of non-sparse kernel fusion to integrate well processed biomedical data.

We also tried to regularize the integration of kernels in the L_∞-norm approach by setting the θ_{min} parameter. Nine different θ_{min} were tried uniformly from 0.1 to 0.9 and the influence on performance is shown in Figure 3.2. As shown, to select the optimal θ_{min} was non-trivial in the present case study because the effect was not significant when the values were smaller than 0.6. Especially, even when we increased θ_{min} to 0.9, which was almost averagely combining the kernels, the performance was still lower than the L_2-norm approach.

In LS-SVM kernel fusion, we estimated the λ jointly as a coefficient assigned to an identify matrix. To validate whether the estimation is optimal, we set λ to 51 different values uniformly sampled on the log_2 scale from -10 to 40. We compared the joint estimation result with the optimal classification performance among the sampled λ values and it was proved as optimal for most of the results. An example is illustrated in Figure 3.3 to combine the kernels composed by 27 genes and 17 proteins. The coefficients estimated by the L_∞-norm were almost 0 thus the λ values were very big. In contrast, the λ values estimated by the non-sparse L_2 method were at reasonable scales.

Table 3.3: Overall results of patient classification. In L_∞ and L_2-norm LS-SVM, the λ was estimated jointly as kernel coefficient. In L_1-norm LS-SVM , λ was set to 1. In all SVM approaches, the C parameter of box constraint equated to 1. In the table, the row and column labels respectively represent the numbers of genes (g) and proteins (p) used to construct the kernel matrices. To evaluate the AUC of LOO validation, we ignored the bias term b (as the implicit bias approach) because its value varied by each left out sample. In our problem, considering the bias term decreased the AUC performance. The performance was compared among six algorithms at the same number of genes and proteins, where the best values are represented in bold, the second best ones in italic. The best performance of all the feature selection results is underlined. The complete experimental results containing 26 different numbers of genes and 26 numbers of proteins is available at `http://homes.esat.kuleuven.be/~syu/l2lssvm.html`

	LS-SVM(L_∞)					SVM(L_∞)				
	14p	15p	16p	17p	18p	14p	15p	16p	17p	18p
24g	0.9416	0.9481	*0.9253*	0.9188	0.9188	0.8669	0.8896	0.8669	0.8669	0.8636
25g	*0.9610*	*0.9610*	0.9481	0.9383	0.9351	0.8864	0.8896	0.8766	0.8799	0.8766
26g	0.9513	0.9513	0.9188	0.9156	0.9123	0.8734	0.8864	0.8766	0.8701	0.8636
27g	0.9383	0.9351	0.9188	0.9123	0.9058	0.8571	0.8636	0.8636	0.8669	0.8539
28g	0.9448	0.9513	0.9383	0.9253	0.9286	0.8571	0.8669	0.8669	0.8636	0.8604

	LS-SVM(L_1)					SVM(L_1)				
	14p	15p	16p	17p	18p	14p	15p	16p	17p	18p
24g	**0.9513**	**0.9513**	**0.9318**	**0.9318**	0.9253	0.9253	0.9416	0.9286	**0.9318**	0.9253
25g	**0.9643**	**0.9675**	**0.9578**	**0.9545**	**0.9545**	0.9416	0.9481	0.9351	0.9286	0.9286
26g	**0.9643**	**0.9643**	**0.9545**	**0.9545**	**0.9545**	0.9416	0.9481	0.9318	0.9318	0.9318
27g	**0.9643**	**0.9643**	**0.9545**	**0.9513**	**0.9481**	0.9383	0.9416	0.9286	0.9318	0.9318
28g	**0.9578**	**0.9675**	**0.9513**	**0.9513**	**0.9481**	0.9416	0.9416	0.9351	0.9351	0.9318

	LS-SVM(L_2)					SVM(L_2)				
	14p	15p	16p	17p	18p	14p	15p	16p	17p	18p
24g	*0.9448*	**0.9513**	*0.9253*	*0.9221*	**0.9286**	0.9091	0.9123	0.9026	0.9058	0.8994
25g	*0.9610*	*0.9610*	*0.9513*	*0.9448*	*0.9448*	0.9253	0.9351	0.9188	0.9156	0.9156
26g	*0.9610*	*0.9545*	*0.9448*	*0.9351*	*0.9351*	0.9253	0.9416	0.9188	0.9221	0.9221
27g	*0.9578*	*0.9513*	*0.9448*	*0.9416*	*0.9351*	0.9221	0.9188	0.9156	0.9188	0.9188
28g	*0.9545*	**0.9675**	**0.9513**	*0.9416*	*0.9448*	0.9188	0.9286	0.9188	0.9221	0.9188

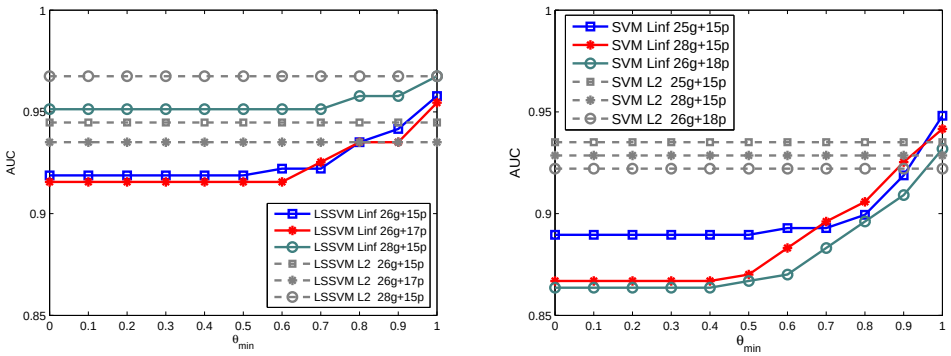


Figure 3.2: Comparison of the L_2-norm approach with regularized L_∞-norm MKL approach in patient classification. Left: the effect of θ_{min} in L_∞-norm LS-SVM MKL, the grey dash lines illustrate the L_2-norm performance; Right: the effect of θ_{min} in L_∞-norm SVM MKL and the corresponding L_2-norm results.

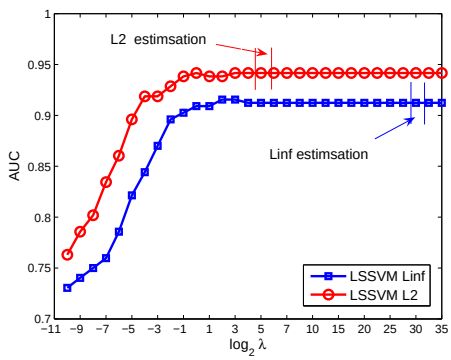


Figure 3.3: The effect of predefined λ values in LS-SVM MKL when combining 4 kernels constructed by 27 genes and 17 proteins. The jointly estimated results of λ from the L_∞-norm and L_2-norm approaches are also shown.

3.6 Summary

In this chapter we analyzed the problem of combing multiple kernels in SVM based novelty detection and classification. The main contributions are concluded as follows:

- We reviewed kernel fusion methods for one class, binary class and multi-class learning problems.
- For binary class and multi-class classification, we compared the mathematical formulations between the SVM and the LS-SVM and derived their MKL solutions in the forms of QCQP.
- We proposed a new kernel fusion scheme by optimizing the L_2-norm of the multiple kernels. It is shown that the L_2-norm fusion may result in a non-sparse combination of kernels. In two real biomedical applications, the non-sparse data integration is shown performing better than the sparse solution to thoroughly combine data sources.

All the MKL formulations presented in this chapter are in principle solvable as SOCP, QCQP, or QCLP. However, for large-scale problem, quadratic solvers and conic servers may still have huge computational burden and become very inefficient. In the forthcoming chapter, we will focus on the issue about how to reduce the complexity of the MKL problem on large scale data.

Chapter 4

Kernel fusion for large scale data

4.1 Introduction

In the preceding chapter we discussed the issue of MKL for one class, binary class and multi-class problems. Though kernel learning problem is convex, it is non-smooth, making the direct application of the simple local descent algorithms such as Sequential Minimization Optimization (SMO) infeasible [14]. The Semi-definite Programming (SDP) formulation proposed by Lanckriet *et al.* [106] and Kim *et al.* [95] is computationally prohibitive in the multi-class problem due to the presence of positive semi-definite constraints and thus can only be solved approximately by relaxation [191]. Fortunately, all these problems can be formulated as QCQP [26], which can be solved in principle by general purpose interior point based optimization toolboxes such as Mosek [8]. Theoretical results on interior point method [131] and experimental comparisons have shown that the QCQP is solved more efficiently than the SDP, and it seems to scale up well on large-scale problems. Unfortunately, the MKL problem becomes challenging when it scales up in three dimensions: the number of samples, the number of classes and the number of kernels. When these scales are all large, memory issues may arise as the kernel matrices need to be allocated. Kernel matrices may have full rank, but their eigenvalues may decay (at least) exponentially [186]. One possible solution is to reduce the memory burden by transforming QP as conic optimization, e.g., SOCP, thus one can approximate the large kernels using low-rank decompositions. However, to precisely approximate the kernel matrices in low rank is not always feasible, especially when the intrinsic dimensions of the kernel matrices are large. To tackle the computational burden of QP with high precision, Sonnenburg *et*

al. propose the SIP [74] method to approximate the SVM MKL solution as the iterations of a linear programming and a single kernel SVM [160]. Consider the single kernel SVM, it is known that the LS-SVM [166] is solved as a set of linear equations, which is more efficient than the QP solution of SVM. Therefore, the SIP formulation of the L_∞-norm LS-SVM MKL is basically the iterations of two linear problems and can be solved much more efficiently than the standard SVM MKL. Ye *et al.* [191] formulate the discriminant problem by the FDA, which is equivalent to the LS-SVM, and obtain an unconstrained solution of MKL.

In this chapter, our novel contribution is the systematic evaluation of convex optimization variants for large scale MKL problem. In particular, for the first time in the literature, we propose the SIP formulation to solve the L_2-norm LS-SVM MKL problem.

4.2 Low rank kernel fusion for 1-SVM

In the 1-SVM MKL (3.15), the kernel matrix can be decomposed as $K_j = C_j C_j^T$ with constraints $K_j \succeq 0, C_j \succeq 0$. The symbol $\succeq$ denotes the PSD property and the decomposition C_j may not be unique. The problem can be reformulated as a conic optimization in the form of C_j, given by

$$\begin{aligned} \min_{\vec{\alpha}} \quad & t \qquad (4.1) \\ \text{s.t.} \quad & t \geq ||\vec{\alpha}^T C_j||^2, \quad j = 1, ..., p \\ & 0 \leq \alpha_k \leq \frac{1}{\nu N}, \quad k = 1, ..., N \\ & \sum_{k=1}^{N} \alpha_k = 1. \end{aligned}$$

The conic form in (4.1) enables us to approximate C_j by Incomplete Cholesky Decomposition (ICD), given by $K_j \approx \mathscr{C}_j \mathscr{C}_j^T$, where $\mathscr{C} \in \mathbb{R}^{N \times d} (N \gg d)$ is the low rank decomposed matrix. Obviously, to replace C with $\mathscr{C}$ in (4.2) will significantly reduce the allocated memory because $\mathscr{C}$ often has much smaller dimensions than C. When applying ICD, the number (d) of ICD components is usually predefined so we denote the formulations as "Fixed-Size 1-SVM MKL".

4.2.1 Fixed-Size 1-SVM MKL: SOCP formulation

We denote the new formulation of the 1-SVM MKL using $\mathscr{C}_j$ as a *Fixed-Size* 1-SVM MKL because the low rank d is pre-selected for optimization. The dual problem

becomes

$$\begin{aligned} \min_{\vec{\alpha}} \quad & \eta \\ \text{s.t.} \quad & \eta \geq ||\vec{\alpha}^T \mathscr{C}_j||_2, \quad j = 1, ..., p \\ & 0 \leq \alpha_k \leq \frac{1}{\nu N}, \quad k = 1, ..., N \\ & \sum_{k=1}^{N} \alpha_k = 1, \end{aligned} \tag{4.2}$$

where $\mathscr{C}_j \in \mathbb{R}^{n \times d}$ is equal to the first d dimensions of C_j by taking the most important ICD components. The implementation of the Fixed-Size 1-SVM MKL is described as follows:

1. Consider the set of full size kernels $K_j \in \mathbb{R}^{N \times N}, \quad j = 1, ..., p$, where N is the number of samples and p is the number data sources.
2. Apply ICD on K_j with pre-specified precision, obtain $\hat{C}_j \in \mathbb{R}^{N \times l}$. To ensure the resulting decomposition $l \leq d$, the precision of ICD is over-estimated.
3. Truncate $\hat{C}_j$ by keeping the d number of major ICD components, obtain $\mathscr{C}_j \in \mathbb{R}^{N \times d}$.
4. Use $\mathscr{C}_j$ as the approximation of kernel decomposition to solve the SOCP formulated in (4.2).

The objective value in (4.2) is the square root of objective value in (4.1) thus $\eta = \sqrt{t}$. The SOCP problem can be solved as the conic optimization solver (i.e., Sedumi [164]) or the general QP solver (i.e., Mosek [8]).

4.2.2 Fixed-Size 1-SVM MKL: Separable QP formulation

The Fixed-Size 1-SVM MKL can also be formulated as a separable QP and solved by a QP solver. The separable QP is given by

$$\begin{aligned} \min_{\vec{\alpha}} \quad & t \\ \text{s.t.} \quad & \vec{\beta}_j = \vec{\alpha}^T \mathscr{C}_j, \quad j = 1, ..., p \\ & t \geq \vec{\beta}_j^T \vec{\beta}_j, \quad j = 1, ..., p \\ & 0 \leq \alpha_k \leq \frac{1}{\nu N}, \quad k = 1, ..., N \\ & \sum_{k=1}^{N} \alpha_k = 1, \end{aligned} \tag{4.3}$$

where $\vec{\beta}_j$ is a d dimensional vector. The objective in (4.3) is equivalent to the objective of (4.1) and can be solved by a QP solver (i.e., Moesk [8]).

4.2.3 Numerical experiment

The experiment adopts two UCI [22] digit recognition data sets, pen-digit and optical-digit, to compare the efficiency of the SOCP and the separable QP. In each data source we sample 3750 digits (375 samples for each class) and construct the kernel matrices by linear and Radial Basis Function (RBF) kernel functions. To simulate the problem of one class learning, we assume that 3000 digits are used for training and the remaining 750 digits as test data used for ranking. The kernel matrices of 3750 samples are centered and we retrieve 3000 samples for training. In experiment, in each data source we take one kernel matrix and combine the two kernel matrices by L_∞ 1-SVM MKL. The 1-SVM MKL problem is solved using the full size kernel matrix. The benchmark is carried on the same hardware platform using two optimization solvers: Sedumi and Mosek. Since 1-SVM MKL can either be solved as a SOCP or as a QCLP, we compare 3 different approaches: to solve it as a QCLP by Mosek, to solve it as a SOCP by Sedumi, and to solve it as a SOCP by Mosek. The comparison of the computational time is shown in Figure 4.1. Moesk is much more efficient than Sedumi, in particular, solving the problem as a QP is more preferable than as a SOCP because the SOCP often contains a large number of variables and constraints.

Based on the solutions obtained on the full rank kernel matrices, we evaluate the effect of using low rank kernel approximation. Firstly, the two linear kernel

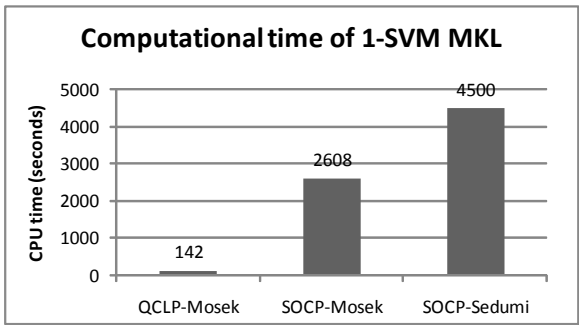


Figure 4.1: Comparison of different solvers and formulation in a large scale 1-SVM MKL problem

matrices are combined. The original pen-digit data has 16 features and the optical-digit data has 64 features. The kernel matrices are created on 3000 samples by the linear kernel function and then are decomposed using the ICD. The results solved in the 1-SVM MKL using the complete kernel matrices are denoted the solutions as $\vec{\theta}$, $\vec{\alpha}$, and the optimal value of objective function o. Then we solve the SOCP formulated as (4.2) and the separable QP formulated as (4.3) using the low rank decompositions. The obtained solutions are denoted as $\vec{\mu}'$, $\vec{\alpha}'_q$, and o'. The fixed size parameter d is varied from 2000 to 100 and the error of low rank approximation is measured by three indices, e_o, e_μ, and e_α, given by

$$e_o = \frac{|o - o'|}{|o|},$$

$$e_\mu = 1 - \frac{\vec{\theta} \cdot \vec{\theta}'}{||\vec{\theta}||_2 ||\vec{\theta}'||_2},$$

$$e_\alpha = \frac{||\vec{\alpha} - \vec{\alpha}'||_2}{||\vec{\alpha}||_2}.$$

The computational time of the SOCP (solved by Sedumi) and the separable QP (solved by Mosek) is also compared with the original QP solution. As shown in Table 4.1, the separable QP is solved more efficiently than the SOCP. The intrinsic dimensions of the kernel matrix constructed by linear function are small thus the $\vec{\alpha}$ is approximated with high precision using the low rank matrices with 500 ICD components.

Next, we evaluate the low rank approximation approach to combine two RBF kernel matrices using the same error indices and procedure as the linear case. As shown in Table 4.2, the error observed on the combination of RBF kernel matrices is larger than the linear ones, but the precision obtained on 500 ICD

Table 4.1: Using low rank approximation of kernel matrices to combine two linear kernel matrices in 1-SVM MKL. The regularization parameter is set as $\nu = 0.5$. The computational time is measured on a dual Opteron 250 Unix system with 16Gb memory

dimension	SOCP(sec)	separable QP(sec)	e_o	e_θ	e_α
2000	2768	1777	0	0.0000	0
1500	2098	1158	8.3E-05	0.0000	0.0002
1000	1472	579	2.2E-04	0.0000	0.0050
800	1092	399	3.9E-04	0.0000	0.0128
500	**262**	**184**	**0.0556**	**0.0000**	**0.0563**
300	224	71.4	0.2778	0.0000	0.1799
100	18.2	11.4	0.9556	0.0000	0.3267

Table 4.2: Using low rank approximation to combine two RBF kernel matrices in 1-SVM MKL. The kernel width is selected empirically as twice of the average pairwise distances of all the samples. The regularization parameter is set as $\nu = 0.5$. The computational time is measured on a dual Opteron 250 Unix system with 16Gb memory

dimension	SOCP(sec)	separable QP(sec)	e_o	e_θ	e_α
2000	3039	1552	0	1.8E-06	0.0002
1500	2157	1066	2.4E-04	1.7E-05	0.0011
1000	1531	544	3.2E-03	1.0E-04	0.0057
800	932	378	7.1E-03	2.9E-04	0.0115
500	**295**	**165**	**0.0714**	**6.6E-04**	**0.0322**
300	248	70.6	0.1429	6.3E-04	0.0757
100	19	11.3	0.8571	0.0141	0.2265

components is still good. As a conclusion of the results in Fig 4.1, Table 4.1, and Table 4.2, to solve the SOCP problem in Sedumi is less efficient than to solve the equivalent QP in Mosek. Transforming the large scale QP problem into a SOCP and a separable QP is often not preferable, since it may involve a large number of additional constraints and variables to slow down the computation. Therefore, low-rank approximation is generally not advisable on a QP solver. For a conic solver, the SOCP solution is inefficient on large scale problems.

4.3 Large scale multi-class MKL

In the previous section, we have investigated one class kernel fusion problem. In this section, we will explore efficient methods to solve the large scale MKL for binary class and multi-class classification problems.

4.3.1 Low rank approximation for SVM MKL: conic formulation

According to (3.23), the QCQP formulation of the L_∞-norm SVM MKL is given by

$$\begin{aligned}
\min_{\rho,\vec{\alpha}_i} \quad & \frac{1}{2}\rho - \sum_{i=1}^{q} \vec{\alpha}_i^T \vec{1} \\
\text{s.t.} \quad & (Y_i\vec{\alpha}_i)^T\vec{1} = 0, \quad i = 1, ..., q \\
& 0 \le \alpha_{ki} \le C, \quad k = 1, ..., N, \quad i = 1, ..., q \\
& \rho \ge \sum_{i=1}^{q} (\vec{\alpha}_i^T Y_i K_j Y_i \vec{\alpha}_i), \quad j = 1, ..., p.
\end{aligned}$$

The formulation can be equivalently reformulated as the form of a SOCP by decomposing the kernel matrix $K_j = C_j C_j^T$, given by

$$\begin{aligned}
\min_{\psi,\vec{\alpha}_i} \quad & \frac{1}{2}\psi - \sum_{i=1}^{q} \vec{\alpha}_i^T \vec{1} && (4.4) \\
\text{s.t.} \quad & (Y_i\vec{\alpha}_i)^T\vec{1} = 0, \quad i = 1, ..., q \\
& 0 \le \alpha_{ki} \le C, \quad k = 1, ..., N, \quad i = 1, ..., q \\
& v_{ji} \ge ||C_j^T Y_i \vec{\alpha}_i||_2, \quad j = 1, ..., p, \quad i = 1, ..., q && (4.5) \\
& u_j \ge ||v_{ji}||_2, \quad j = 1, ..., p, \quad i = 1, ..., q && (4.6) \\
& \psi \ge u_j, \quad j = 1, ..., p.
\end{aligned}$$

In the formulation above, (4.5) is the Q-cone constraint posed on the j-th kernel matrix and the i-th class, (4.6) is the Q-cone constraint posed on the j-th kernel matrix. Combining (4.5) and (4.6), we have

$$u_j \ge \left(\sum_{i=1}^{q} (\vec{\alpha}_i^T Y_i G_j Y_i \vec{\alpha}_i) \right)^{\frac{1}{2}}, \qquad (4.7)$$

thus the last constraint in (4.4) minimizes ψ, which is equivalent to minimizing $\sqrt{\rho}$ in (3.23) and the optimal solutions of $\vec{\alpha}_i^*$ are the same. We can analogously approximate C_j by a low rank matrix $\mathscr{C}_j$ in (4.5) and obtain a smaller size SOCP problem. According to the benchmark result (not shown in the thesis), the precision of the approximation is mainly determined by the choice of low-rank d. In the SOCP, to gain much efficient solution over the original QP problem, one has to choose a very small d, which may be risked by a large approximation error. In summary, it is only preferable to transform the QCQP into a SOCP problem when there is no QP solver available. Otherwise, to solve the QCQP problem in a QP solver is often much more efficient than to solve the equivalent SOCP problem.

4.3.2 Low rank kernel fusion for SVM MKL: separable QP formulation

To take the advantages of the both sides, the efficiency of the QP solver and the low-rank approximation, one may reformulate the MKL as a separable QP, given by

$$\begin{aligned} \min_{\vec{\beta}_i, \vec{\alpha}_i} \quad & \frac{1}{2} \sum_{i=1}^{q} \vec{\beta}_i^T \vec{\beta}_i - \sum_{i=1}^{q} \vec{\alpha}_i^T \vec{1} \\ \text{s.t.} \quad & (Y_i \vec{\alpha}_i)^T \vec{1} = 0, \quad i = 1, ..., q \\ & 0 \le \alpha_{ki} \le C, \quad k = 1, ..., N, \quad i = 1, ..., k \\ & \vec{\beta}_i = \mathscr{C}_j^T Y_i \vec{\alpha}_i, \quad j = 1, ..., p, \quad i = 1, ..., q. \end{aligned} \tag{4.8}$$

Though the separable QP for multi-class may also involve more variables and constraints than the unseparated QP in (3.23), the computational burden may be reduced if the rank d of $\mathscr{C}_j$, comparing to N, is smaller enough. Of course, there is still a trade-off because the smaller d will probably lead to a larger approximation error.

Numerical experiment

In the experiment, we apply the L_∞ SVM MKL to combine kernel matrices obtained from the pen-digit and optical-digit recognition data as used in the previous experiment. To construct the kernel matrix, 2000 samples (200 for each digit) are retrieved from each data source. The kernel matrices are decomposed by the ICD. We first use Mosek to solve the original QP problem as (3.23) using the complete kernel matrices and denote the solution as $\vec{\theta}$, $\vec{\alpha}_i$, and the value of objective as o. Next we apply Mosek to solve the separable QP problem as (4.8)

using the low rank decompositions of kernel matrices. The approximated solutions are denoted as $\vec{\theta}'$, $\vec{\alpha}'_i$, and o'. We vary the number of low rank d from 10 to 50 and evaluate the error of the solutions by three indices, e_o, e_θ, and e_α, given by

$$e_o = \frac{|\mathcal{J}_{(4.8)} - \mathcal{J}_{(3.23)}|}{|\mathcal{J}_{(3.23)}|},$$

$$e_\theta = 1 - \frac{\vec{\theta} \cdot \vec{\theta}'}{||\vec{\theta}||_2 ||\vec{\theta}'||_2},$$

$$e_\alpha = \frac{||A - A'||_2}{||A||_2}, \quad A = [\vec{\alpha}_1, ..., \vec{\alpha}_q], A' = [\vec{\alpha}'_1, ..., \vec{\alpha}'_q].$$

The computational time of the separable QP is also compared with the original problem. The result of combining two linear kernel matrices is illustrated in Table 4.3. Because the inherent dimension of a linear kernel matrix is bounded by the dimension of the original data, the approximated solution is reliable when $d = 60$. The low rank approximation also improves the computational efficiency: the separable QCQP is solved much faster than the standard QCQP.

Table 4.3: Using low rank approximation to combine two linear kernel matrices as the L_∞ SVM MKL. The regularization parameter is set as $\lambda = 0.001$.

	dimension	time(sec)	e_o	e_μ	e_α
QCQP	2000	35.6	0	0	0
Separable QP	**60**	**1.3743**	**0**	**0**	**3.70E-06**
	50	1.1098	0.0014	6.44E-07	0.0376
	40	0.8786	0.0140	1.35E-05	0.1443
	30	0.7505	0.0585	2.13E-04	0.3601
	20	0.5035	0.1530	2.10E-03	0.5963
	10	0.3181	0.7666	4.58E-02	1.0225

We aware that to approximate the kernel matrices mapped to infinite dimensional space, e.g., the ones constructed by RBF function, might cause large approximation error when the low-rank is too small. To demonstrate this, we next combine the two kernel matrices constructed by RBF functions, where the kernel width parameters are selected as twice of the average distance of all pairwise data samples. As shown in Table 4.4, since the inherent dimension of the RBF kernel matrices is infinite, the low rank approximation leads to a large approximation error. To reduce this error smaller than 5%, we have to increase the rank to 900. At this rank, the separable QP contains a large amount of additional variables and constraints, which makes the optimization process very inefficient.

Table 4.4: Using low rank approximation to combine two RBF kernel matrices as the L_∞ SVM MKL. The regularization parameter is set as $\lambda = 0.001$.

	dimension	time(sec)	e_o	e_μ	e_α
QCQP	2000	39.2356	0	0	0
Separable QP	**900**	**292.0419**	**0.0063**	**3.56E-05**	**0.0453**
	800	243.0001	0.0083	2.79E-05	0.0533
	700	192.2551	0.0136	1.26E-04	0.0726
	600	148.7579	0.0246	4.60E-04	0.1391
	500	104.8798	0.0351	5.01E-04	0.2301
	400	71.9904	0.0737	2.46E-04	0.3378
	300	46.6797	0.126	1.51E-04	0.5443
	200	22.9864	0.2752	1.09E-04	0.7289
	100	8.3359	0.7434	0.0262	1.2202

4.3.3 SVM MKL: SIP formulation

Though not shown in this thesis, it is known that the binary class MKL problem can also be formulated and solved as a SDP [106]. In SDP, the convex constraint $K \succeq 0$ is equivalent to

$$\vec{\alpha}^T K \vec{\alpha} = \text{trace}\left(\vec{\alpha}\vec{\alpha}^T K\right) \geq 0 \quad \forall d \in \mathbb{R}^n \tag{4.9}$$

These constraints are linear inequalities in the matrix variable K, but there is an infinite number of them [100]. Thus SDP is naturally a *semi-infinite linear programming* (SIP) in $\mathbb{R}^{\frac{n(n+1)}{2}}$ [100]. It is also shown in the previous discussion that when using a low rank kernel approximations in SOCP, the problem often results in a large number of constraints. So people are interested in formulating QP as SIP to reduce the computational burden. The standard form of a SIP is given by

$$\max_{\vec{y}} \quad \vec{b}^T \vec{y} \tag{4.10}$$

$$\text{s.t.} \quad f_t(\vec{y}) \leq 0, \quad \forall t \in \Upsilon$$

where all the constraint functions in $f_t(y), \forall t \in \Upsilon$ are linear or nonlinear. Since there are a infinite number of constraints, to solve it, a *discretization* method is usually applied [74, 145, 89] as follows:

1. Choose a finite subset $\mathcal{N} \subset \Upsilon$.
2. Solve the convex programming problem

$$\arg\max_{\vec{y}} \{ \ \vec{b}^T \vec{y} \mid f_t(\vec{y}) \leq 0, \quad t \in \mathcal{N} \}. \tag{4.11}$$

3. If the solution of (2) is not satisfactorily close to the original problem then choose a larger, but still finite subset $\mathcal{N}$ and repeat from Step 2.

The convergence of SIP and the accuracy of the discretization algorithm have been extensively studied, e.g., see [74, 145, 89]. The accuracy of discretization method is mainly determined by the tolerance value defined in the stopping criterion. Kaliski *et al.* state that solving the problem to machine precision is not always possible and they propose the tolerance value as $1.0e-14$ for implementation [89]. However, we find that in MKL problem the tolerance value can be much looser than that value, and sometimes an extremely small tolerance value can lead to larger approximation error. Based on the work of [160][191], the SIP formulation for standard SVM multi-class MKL is summarized in the following theorem.

Theorem 4.1 (SIP-SVM-MKL). Given a set of p kernel matrices $K_1, ..., K_p$, the optimal kernel matrix, in the form of a convex linear combination of the given kernel matrices, that minimizes the MKL objective in (3.23) is solved by the following SIP problem:

$$\max_{\vec{\theta},\rho} \quad \rho \tag{4.12}$$

$$\text{s.t.} \quad \vec{\theta} \geq 0,$$

$$\sum_{j=1}^{p} \theta_j = 1,$$

$$\sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}) \geq \rho, \quad \forall \vec{\alpha}, \tag{4.13}$$

$$f_j(\vec{\alpha}) = \sum_{i=1}^{q} \left(\frac{1}{2} \vec{\alpha}_i^T Y_i K_j Y_i \vec{\alpha}_i - \vec{\alpha}_i^T \vec{1} \right), \tag{4.14}$$

$$0 \leq \alpha_{ki} \leq C, \quad k = 1, ..., N, \quad i = 1, ..., q$$

$$(Y_i \vec{\alpha}_i)^T \vec{1} = 0, \quad i = 1, ...q.$$

Proof: The objective in (3.23) can be equivalently reformulated as

$$\max_{\vec{\theta},\vec{\alpha}} \quad \sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}) \tag{4.15}$$

$$\text{s.t.} \quad \theta_j \geq 0, \quad j = 1, ..., p$$

$$\sum_{j=1}^{p} \theta_j = 1,$$

$$f_j(\vec{\alpha}) = \sum_{i=1}^{q} \left(\frac{1}{2} \vec{\alpha}_i^T Y_i K_j Y_i \vec{\alpha}_i - \vec{\alpha}_i^T \vec{1} \right),$$

$$0 \leq \alpha_{ki} \leq C, \quad k = 1, ..., N, \quad i = 1, ..., q$$

$$(Y_i \vec{\alpha}_i)^T \vec{1} = 0, \quad i = 1, ..., q.$$

Denote $\vec{\alpha}^*$ as the optimal solution of the problem in (4.15), the optimal is given by $\rho^* = \sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}^*)$, we have

$$\sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}) \geq \rho^* \quad \forall \vec{\alpha}. \tag{4.16}$$

Let us define ρ as

$$\rho = \min_{\vec{\alpha}} \sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}) \tag{4.17}$$

and replace ρ in the objective of (4.15), the problem is equivalent as maximizing the minimal boundary of ρ. □

The SIP problem described above can be solved by the *discretization* method, which is equivalent to the *column generation* method proposed by Sonnenburg *et al.* [160] and Ye *et al.* [191]. According to their results, we implement SIP as a

bi-level algorithm and the pseudo code is illustrated in Algorithm 4.3.1.

Algorithm 4.3.1: SIP-SVM-MKL($K_1, K_2, ..., K_p, F$)

Obtain the initial guess $\vec{\alpha}^{(0)} = [\vec{\alpha}_1^{(0)}, ..., \vec{\alpha}_q^{(0)}]$ by randomly generated $\vec{\alpha}_1^{(0)}, ..., \vec{\alpha}_k^{(0)}$
Compute $f_j(\vec{\alpha}^{(0)})$ as (4.14) for each kernel matrixK_j
while $(\Delta\rho > \varepsilon)$
do
- $step1$: Fix $\vec{\alpha}$, solve $\vec{\theta}^{(\tau)}$then obtain $\rho^{(\tau)}$
- $step2$: Compute the optimal kernel matrix$\mathbf{\Omega}^{(\tau)}$ in the τ-th loop by $\vec{\theta}^{(\tau)}$ and $K_1, ..., K_p$
- $step3$: Minimize $f_j(\vec{\alpha})$ with single kernel matrix$\mathbf{\Omega}^{(\tau)}$ and obtain the optimal $\vec{\alpha}^{(\tau)}$
- $step4$: Compute $f_1(\vec{\alpha}^{(\tau)}), ..., f_p(\vec{\alpha}^{(\tau)})$ as (4.14) using $\vec{\alpha}^{(\tau)}$ and $K_1, K_2, ..., K_p$
- $step5$: Compute the update of ρ as $\Delta\rho = |1 - \frac{\sum_{j=1}^{p} \theta_i^{(\tau-1)} f_j(\vec{\alpha}^{(\tau)})}{\rho^{(\tau-1)}}|$
- $step6$: $\tau = \tau + 1$

comment: τ is the indicator of the current loop
return $(\vec{\theta}^{(\tau)} = [\theta_1^{(\tau)}, ..., \theta_p^{(\tau)}]^T, \vec{\alpha}^{(\tau)} = [\alpha_1^{(\tau)}, ..., \alpha_q^{(\tau)}]^T)$

In Algorithm 4.3.1, the optimal θ and ρ are computed for a restricted subset of constraints in (4.10) as a restricted master problem in step 1. Then a second problem generates a new, yet unsatisfied constraint determined by $\vec{\alpha}$ in step 3. If $\vec{\alpha}$ is not satisfied by the current θ and ρ, it will be added successively to the restricted master problem until all the constraints are satisfied. This makes the algorithm converges faster because the $\vec{\alpha}$ obtained by solving

$$\min_{\vec{\alpha}_i} \sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}) \tag{4.18}$$

maximizes the violation for current $\vec{\theta}$ and ρ. If $\sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}) \geq \rho$, then all the constraints are satisfied thus $\vec{\theta}$ and ρ reach the optimal values. Otherwise, this constraint is added to the restricted master problem and the iteration continues until convergence. In step 3, the problem of minimizing (4.14) on given $\mathbf{\Omega}^{(\tau)}$ is equivalent to solving the dual problem of a single kernel matrix SVM, which can be generally solved as QP problem [160].

Algorithm 4.3.1 is also applicable to the L_2-norm SVM MKL, whereas the non-convex constraint $||\vec{\theta}||_2 = 1$ in Step 1 needs to be relaxed as $||\vec{\theta}||_2 \leq 1$, and the $f_j(\vec{\alpha})$ term in (4.14) is modified to only containing the quadratic term. The SIP

formulation of the L_2-norm SVM MKL is given by

$$\begin{aligned}
&\max_{\vec{\theta},\rho} \ \rho \qquad (4.19)\\
&\text{s.t.} \ \ \vec{\theta}_j \geq 0, \quad j = 1, ..., p,\\
&\qquad ||\vec{\theta}||_2 \leq 1,\\
&\qquad \sum_{j=1}^{p} \theta_j f_j(\vec{\alpha}) - \sum_{i=1}^{q} \vec{\alpha}^T \vec{1} \geq \rho, \quad \forall \vec{\alpha}\\
&\qquad f_j(\vec{\alpha}) = \frac{1}{2} \sum_{i=1}^{q} \left(\vec{\alpha}_i^T Y_i K_j Y_i \vec{\alpha}_i \right), j = 1, ..., p\\
&\qquad 0 \leq \alpha_{ki} \leq C, \quad k = 1, ..., N, \quad i = 1, ..., q\\
&\qquad (Y_i \vec{\alpha}_i)^T \vec{1} = 0, \quad i = 1, ..., q.
\end{aligned}$$

With the modifications, Step 1 of Algorithm 4.3.1 becomes a QCLP problem, given by

$$\begin{aligned}
&\max_{\vec{\theta},\rho} \ \rho \qquad (4.20)\\
&\text{s.t.} \ \ \frac{1}{2} \sum_{j=1}^{p} \theta_j Q_j - \sum_{i=1}^{q} \vec{\alpha}_i^T \vec{1} \geq \rho\\
&\qquad 1 \geq \theta_1^2 + ... + \theta_p^2.
\end{aligned}$$

where $Q_j = \sum_{i=1}^{q} \left(\vec{\alpha}_i^T Y_i K_j Y_i \vec{\alpha}_i \right)$ and $\vec{\alpha}_i$ are given values. Moreover, the PSD property of kernel matrices ensures that $Q_j \geq 0$, thus the optimal solution always satisfies that $||\vec{\theta}||_2 = 1$.

4.3.4 LS-SVM MKL: SIP formulation

We have shown that the QP problem of the standard SVM MKL can be approximated by a SIP which consists of iterations of a single kernel learning. It is known that the LS-SVM is solved as a set of linear equations, which is much more efficient than the QP of SVM. This has inspired us to adopt the LS-SVM MKL solution using SIP formulations.

The L_∞-norm approach of multi-class LS-SVM MKL is formulated as

$$\begin{aligned}
&\max_{\vec{\theta},u} \; u \qquad (4.21)\\
&\text{s.t. } \theta_j \geq 0,\ j = 1, ..., p+1\\
&\quad \sum_{j=1}^{p+1} \theta_j = 1,\\
&\quad \sum_{j=1}^{p+1} \theta_j f_j(\vec{\beta}) \geq u, \quad \forall \vec{\beta},\\
&\quad f_j(\vec{\beta}) = \sum_{i=1}^{q} \left(\frac{1}{2} \vec{\beta}_i^T K_j \vec{\beta}_i - \vec{\beta}_i^T Y_i^{-1} \vec{1} \right), \quad j = 1, ..., p+1.
\end{aligned}$$

In the formulation above, K_j represents for a set of $p+1$ kernel matrices with the $(p+1)$-th kernel matrix being the identity matrix. In the similar way, the L_2-norm LS-SVM MKL is formulated as

$$\begin{aligned}
&\max_{\vec{\theta},u} \; u \qquad (4.22)\\
&\text{s.t. } \theta_j \geq 0,\ j = 1, ..., p+1\\
&\quad \sum_{j=1}^{p+1} \theta_j^2 \leq 1,\\
&\quad \sum_{j=1}^{p+1} \theta_j f_j(\vec{\beta}) - \sum_{i=1}^{q} \vec{\beta}_i^T Y_i^{-1} \vec{1} \geq u, \quad \forall \vec{\beta},\\
&\quad f_j(\vec{\beta}) = \sum_{i=1}^{q} \left(\frac{1}{2} \vec{\beta}_i^T K_j \vec{\beta}_i \right), \quad j = 1, ..., p+1.
\end{aligned}$$

The pseudocode of the L_∞-norm and the L_2-norm LS-SVM MKL is presented in Algorithm 4.3.2.

Algorithm 4.3.2: SIP-LS-SVM-MKL$(K_1, ..., K_p, F)$

Obtain the initial guess $\vec{\beta}^{(0)} = [\vec{\beta}_1^{(0)}, ..., \vec{\beta}_q^{(0)}]$
$\tau = 0$
while $(\Delta u > \varepsilon)$
do
- $step1$: Fix $\vec{\beta}$, solve $\vec{\theta}^{(\tau)}$then obtain $u^{(\tau)}$
- $step2$: Compute kernel combination $\Omega^{(\tau)}$
- $step3$: Solve the single LS-SVM and obtain the optimal $\vec{\beta}^{(\tau)}$
- $step4$: Compute $f_1(\vec{\beta}^{(\tau)}), ..., f_{p+1}(\vec{\beta}^{(\tau)})$
- $step5$: $\Delta u = |1 - \frac{\sum_{j=1}^{p+1} \theta_j^{(\tau-1)} f_j(\vec{\beta}^{(\tau)})}{u^{(\tau-1)}}|$
- $step6$: $\tau = \tau + 1$

comment: τ is the indicator of the current loop
return $(\vec{\theta}^{(\tau)}, \vec{\beta}^{(\tau)})$

Step 1 optimizes $\vec{\theta}$ as a linear programming in the L_∞-norm approach and as a QCLP problem in the L_2-norm approach. Since the regularization coefficient is automatically estimated as θ_{p+1}, Step 3 simplifies to a linear problem as

$$\left[\begin{array}{c|c} 0 & \vec{1}^T \\ \hline \vec{1} & \Omega^{(\tau)} \end{array}\right] \left[\begin{array}{c} b^{(\tau)} \\ \hline \vec{\beta}^{(\tau)} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y^{-1}\vec{1} \end{array}\right], \tag{4.23}$$

where $\Omega^{(\tau)} = \sum_{j=1}^{p+1} \theta_j^{(\tau)} K_j$.

4.3.5 Numerical Experiments

Comparison of the QCQP and the SIP on various number of data samples

In this experiment, the MKL is respectively solved as the QCQP and the SIP on the benchmark data set consists of two linear kernel ms and 10 digit classes. The number of the data samples is increased from 1000 to 3750 and the computational time (seconds) is illustrated in Figure 4.2. The experiment is carried on a dual Opteron 250 Unix system with 16Gb memory.

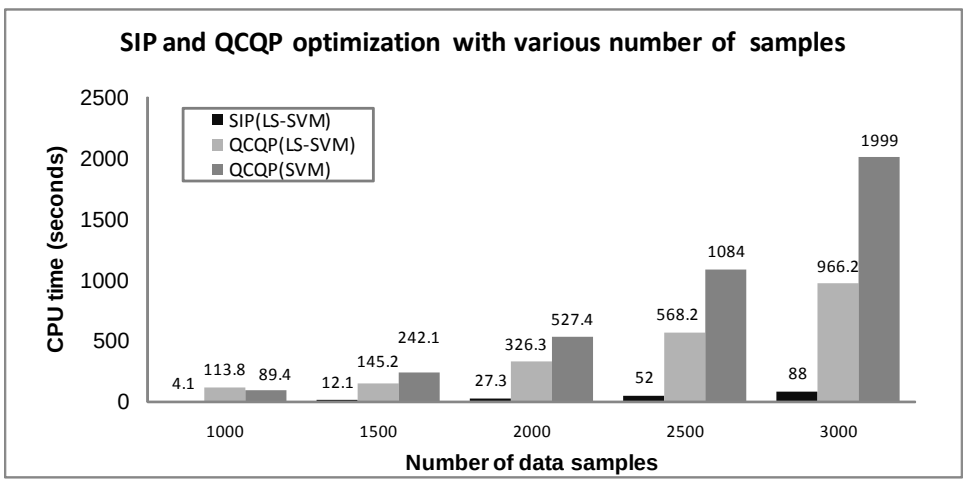


Figure 4.2: Comparison of the QCQP and SIP solution in the SVM MKL and the LS-SVM MKL on different sizes of samples

Comparison of the L_∞ QCQP MKL and the L_∞ SIP MKL on various number of classes

In this experiment, the MKL is respectively solved as the QCQP and the SIP on the benchmark data set consists of two linear kernel matrices and 2000 samples (200 samples from each digit). The number of classes is increased from 2 to 20 and the computational time (seconds) is plotted in Figure 4.3. As shown, in the QCQP the computational time increases steadily with respect to the class number. On the contrary, in SIP-LS-SVM the coefficient of the linear system is identical, thus the computational time is almost invariant with respect to the number of classes. We also try to increase the class number to 2000, and the computational time is still similar to the binary class problem.

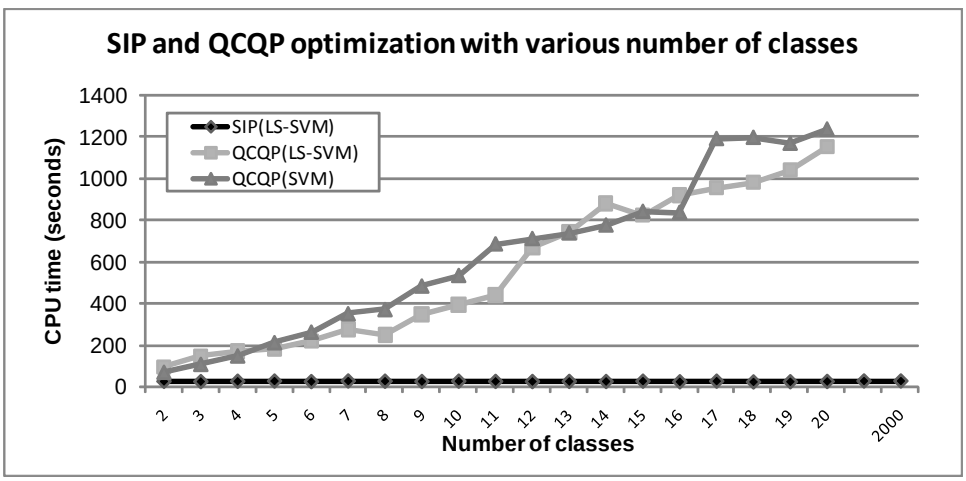


Figure 4.3: Comparison of the QCQP and the SIP solution for the SVM MKL and the LS-SVM MKL on different numbers of classes. The experiment is carried on a dual Opteron 250 Unix system with 16Gb memory.

Comparison of the L_∞ QCQP MKL and the L_∞ SIP MKL on various number of kernel matrices

In this experiment, the MKL is respectively solved as the QCQP and the SIP on a benchmark data set consists of 2000 samples and 2 classes. We use different kernel widths to construct the RBF kernel matrices and increase the number of kernel matrices from 2 to 200. Four MKL approaches are compared and the results are shown in Figure 4.4.

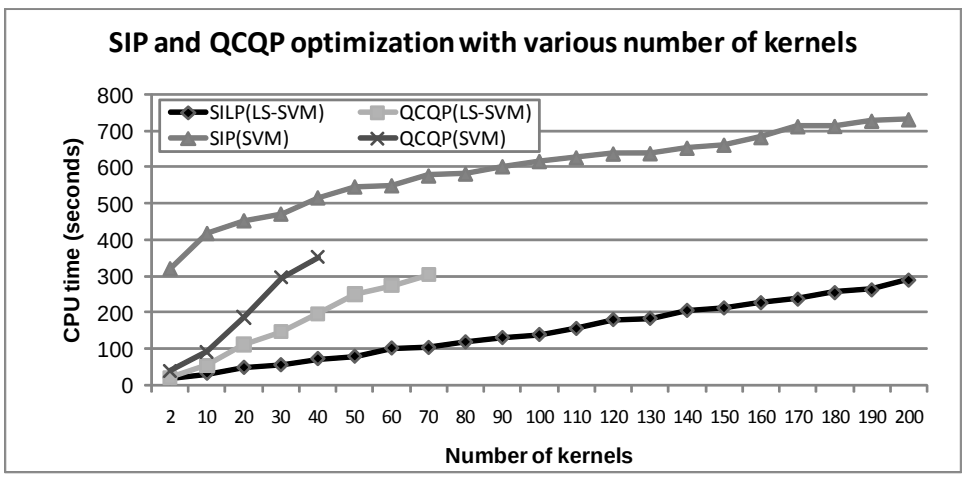


Figure 4.4: Comparison of the QCQP and the SIP solution for the SVM MKL and the LS-SVM MKL on different numbers of kernel matrices. The experiment is carried on a dual Opteron 250 Unix system with 16Gb memory.

Stopping criterion in the L_∞ SIP LS-SVM MKL

In this experiment, we benchmark the effect of the stopping criterion in the SIP LS-SVM MKL using two 2000×2000 kernel matrices and binary class labels. We assume that the $\vec{\beta}$ and θ obtained by the QCQP formulation are the accurate solutions. To produce the accurate results, the relative gap tolerance of Mosek QCQP interior point solver is set to 1.00E-14. We gradually decrease the tolerance value of SIP from 1E-01 to 1E-20. The error indices e_β and e_θ are defined in the similar way as in the previous experiments. The computational time of SIP is also compared at different tolerance values. As shown in Table 4.5, the SIP solution reaches the highest precision at the tolerance value of 1.00E-05. When we further reduce the tolerance, the solution error starts to increase.

Table 4.5: The effect of the stopping criterion in the L_∞ SIP LS-SVM MKL and the approximation error compared with the QCQP solution. The experiment is carried on a dual Opteron 250 Unix system with 16Gb memory.

tolerance	e_β	e_θ	time(seconds)
1.00E-01	0.167498580835201	0.151388760408573	17.6
1.00E-02	0.023159768542745	0.115179353972694	25.0
1.00E-03	0.004082310659036	0.047865486210002	29.1
1.00E-04	0.001608750132959	0.008790394600066	31.8
1.00E-05	0.000007007287276	0.000000173825726	35.87
1.00E-06	0.000352123042239	0.000432788118725	43.75
1.00E-07	0.000293107997199	0.000300838246191	45.91
1.00E-08	0.000248635066674	0.000216945883773	50.88
1.00E-09	0.000237497840066	0.000198047851896	54.19
1.00E-10	0.000235650319233	0.000194995025856	58.75
1.00E-11	0.000236127822591	0.000195781806772	60.91
1.00E-12	0.000235791257203	0.000195227086420	69.11
1.00E-13	0.000235791257203	0.000195227086420	72.15
1.00E-14	0.000235788051441	0.000195221806478	76.84
1.00E-15	0.000235820180235	0.000195274726294	119.96
1.00E-16	0.000230755680228	0.000187020276172	2569.3
1.00E-17	0.000230755680228	0.000187020276172	2532.5
1.00E-18	0.000230755680228	0.000187020276172	2541.9
1.00E-19	0.000230755680228	0.000187020276172	2591.5
1.00E-20	0.000230755680228	0.000187020276172	2561.1

4.4 Summary

In this chapter, we investigated the issue of kernel fusion on large scale data. The main contributions are concluded as follows:

- We compared the computational performance of convex quadratic programming on large scale problems and its equivalent conic optimization problem. We found that the quadratic problem in quadratic solver was solved more efficiently than the conic form in conic solver.
- The low rank kernel approximation was able to improve the computational efficiency of the conic optimization and the separable QP, however, the rank has to be very small. In general, for nonlinear RBF kernel matrices, the original QP was solved more efficiently than the conic problem and the separable QP on a QP solver.
- SIP was a promising method to tackle the computational burden of MKL, in particular, the SIP LS-SVM MKL problem is solved much efficiently because it only consists of linear programming. It also performed well on large scale problems. In a extreme test, the SIP-LS-SVM-MKL solved a huge problem which has 8000 samples, 2000 classes and 200 kernel matrices in about 4100 seconds.
- We derived the L_2-norm MKL for SVM and LS-SVM and proposed the solutions in SIP formulations.

Chapter 5

Optimized data fusion for Kernel K-means Clustering

5.1 Introduction

This chapter presents a novel OKKC to combine multiple data sources for clustering analysis. The objective of K-means clustering is formulated as a Rayleigh quotient function of the between-cluster scatter matrix and the cluster membership matrix. To incorporate multiple data sources, the between-cluster matrix is calculated in the high dimensional feature space where the information of heterogeneous data sources can be easily combined as kernel matrices. The objective to optimize the kernel combination and the cluster memberships on unlabeled data is non-convex, to solve it, we apply an alternative minimization method to optimize the cluster memberships and kernel coefficients iteratively. When the cluster membership is given, we optimize the kernel coefficients as LS-SVM MKL. It is known that the LS-SVM optimizes the Fisher discriminants in the feature space (KFDA) thus it optimizes towards the same objective as the K-means. Therefore, the proposed alternative minimization algorithm converges locally. The proposed OKKC algorithm is related some other algorithms proposed in the literature, however, it has a simpler structure such that it executes more efficiently than other existing methods.

5.2 Objective of K-means clustering

In K-means clustering, a number of k prototypes are used to characterize the data and the partitions C_j are determined by minimizing the distortion as

$$\min \sum_{\vec{x}_i \in C_j} \sum_{j=1}^{k} ||\vec{x}_i - \vec{\mu}_j||^2, \tag{5.1}$$

where $\vec{x}_i$ is the i-th data point, $\vec{\mu}_j$ is the prototype (mean) of the j-th partition C_j, k is the number of partitions (usually predefined). It is known (i.e., Section 14.5.1 in [169]) that (5.1) is equal to the trace maximization of the between scatter matrix S_b as

$$\max_{u_{ij}} \text{trace } S_b, \tag{5.2}$$

where u_{ij} is the hard cluster assignment as $u_{ij} \in \{0, 1\}, \sum_{j=1}^{k} u_{ij} = 1$ and

$$S_b = \sum_{j=1}^{k} n_j (\vec{\mu}_j - \vec{\mu}_0)(\vec{\mu}_j - \vec{\mu}_0)^T, \tag{5.3}$$

where $\vec{\mu}_0$ is the global mean, $n_j = \sum_{i=1}^{N} u_{ij}$ is the number of samples in C_j. Moreover, the within-cluster scatter S_w and total scatter matrix S_t are respectively given by

$$S_w = \sum_{j=1}^{k} \sum_{\vec{x}_i \in C_j} (\vec{x}_i - \vec{\mu}_j)(\vec{x}_i - \vec{\mu}_j)^T, \tag{5.4}$$

$$S_t = \sum_{i=1}^{N} (\vec{x}_i - \vec{\mu}_0)(\vec{x}_i - \vec{\mu}_0)^T \tag{5.5}$$

and

$$S_t = S_w + S_b \tag{5.6}$$

Without loss of generality, we assume that the data $X \in \mathbb{R}^{M \times N}$ has been centered such that the global mean is $\mu_0 = 0$. To express $\vec{\mu}_j$ in terms of X, we denote the scalar cluster membership matrix $A \in \mathbb{R}^{N \times k}$ as

$$A_{ij} = \begin{cases} \frac{1}{\sqrt{n_j}} & \text{if } \vec{x}_i \in C_j \\ 0 & \text{if } \vec{x}_i \notin C_j \end{cases}, \tag{5.7}$$

then $A^T A = I_k$ and the objective of K-means in (5.2) can be equivalently written as

$$\max_A \text{trace}\{A^T X^T X A\} \tag{5.8}$$

In literature, various solutions have been proposed to optimize the objective of K-means, such as the iterative descent method [72], the expectation-maximization method [21], the spectral relaxation method [52], and many others. In particular, the spectral relaxation method relaxes the discrete cluster memberships of L to numerical values, since by construction we have $A^T A = I$, then (5.8) becomes a problem similar to the Rayleigh quotient, given by

$$\begin{aligned} \max_A \quad & \text{trace}\frac{A^T X^T X A}{A^T A}, \\ \text{s.t.} \quad & A^T A = I, \\ & X^T X \succeq 0, \end{aligned} \tag{5.9}$$

where $\succeq$ denotes for the positive semi-definitiveness (PSD). If A reduces to a one column vector, the problem in (5.9) is exactly a Rayleigh quotient and the optimal A^* is given by the eigenvector $\vec{u}_{max}$ in the largest eigenvalue pair $\{\lambda_{max}, \vec{u}_{max}\}$ of $X^T X$. If A represents the relaxed assignment of multi-cluster memberships, according to the Ky Fan [52] (more formal mathematical proofs available in [19, 161]), let the eigenvalues of $X^T X$ be ordered as $\lambda_{max} = \lambda_1 \geq, ..., \geq \lambda_N = \lambda_{min}$ and the corresponding eigenvectors as $\vec{u}_1, ..., \vec{u}_N$, then the optimal A^* subject to $A^{*T} A^* = I_k$ is given by $A^* = U_k Q$, where $U_k = [\vec{u}_1, ..., \vec{u}_k]$, and Q is an arbitrary $k \times k$ orthogonal matrix, and $\max \text{trace}\{Q^T X^T X Q\} = \lambda_1 + .. + \lambda_k$. Thus, for a given cluster number k, the K-means can be solved as an eigenvalue problem and the discrete cluster memberships of the original A can be recovered using the iterative descend K-means method on U_k or using the spectral ordering proposed by Ding and He [53].

5.3 Optimizing multiple kernel matrices for K-means

We further generalize the objective in (5.9) by applying the feature map $\phi(\cdot) : \mathbb{R} \to \mathcal{F}$ on X, then the centered data in $\mathcal{F}$ is denoted as X^Φ, given by

$$X^\Phi = [\phi(\vec{x}_1) - \vec{\mu}^\Phi, \phi(\vec{x}_2) - \vec{\mu}^\Phi, ..., \phi(\vec{x}_N) - \vec{\mu}^\Phi], \tag{5.10}$$

where $\phi(\vec{x}_i)$ is the feature map applied on the column vector of the i-th data point in $\mathcal{F}$, μ_r^Φ is the global mean in $\mathcal{F}$. The inner product $X^T X$ in (5.9) can be combined using the kernel trick $K(\vec{x}_u, \vec{x}_j) = \phi(\vec{x}_u)^T \phi(\vec{x}_v)$, where $K(\cdot, \cdot)$ is a

Mercer kernel. We denote G as the centered kernel matrix as $G = PKP$, where P is the centering matrix $P = I_N - (1/N)\vec{1}_N\vec{1}_N^T$, I_N is the $N \times N$ identity matrix, $\vec{1}N$ is a column vector of N ones.

Now let's assume that $X_1, ..., X_p$ as p different representations of the same N objects. We are motivated to extend the clustering problem of a single data set to multiple data sets, thus we can easily combine the centered kernel matrices G_r, $(r = 1, ..., p)$ in an parametric linear additive manner as

$$\mathbf{\Omega} = \left\{ \sum_{r=1}^{p} \theta_r G_r \;\middle|\; \forall \theta_r \geq 0,\ \sum_{r=1}^{p} \theta_r^{\delta} = 1 \right\}, \tag{5.11}$$

where θ_r are coefficients of the kernel matrices, δ is the parameter to determine the bound of θ_r which is usually set as 1. We will show later in this paper that δ can be set to different values than 1 and leads to different solutions of θ_r.

We first start with $\delta = 1$, the K-means objective in (5.9) is extended to $\mathcal{F}$ to incorporate multiple data sets, given by

$$\boxed{\text{Q5.1:}} \qquad \max_{A,\vec{\theta}} \mathcal{J}_1 = \text{trace}\frac{A^T \mathbf{\Omega} A}{A^T A}, \tag{5.12}$$

$$\text{s.t.} \quad A^T A = I_k,$$

$$\mathbf{\Omega} = \sum_{r=1}^{p} \theta_r G_r,$$

$$\theta_r \geq 0,\ r = 1, ..., p,$$

$$\sum_{r=1}^{p} \theta_r = 1.$$

The idea of combining multiple kernel matrices in K-means clustering is also mentioned in some related work, for instance, the NAML algorithm proposed by Chen *et al.* [32]. In their approach, the objective of K-means is formulated as (5.8) using the Mahalanobis distance. Following their mathematical derivations, the K-means objective on a single data source is given by

$$\max_{A} \quad \mathcal{J} = \text{trace}\left(A^T X_G^T \left(X_G X_G^T + \lambda I_{inf} \right)^{-1} X_G A \right), \tag{5.13}$$

$$\text{s.t.} \quad A^T A = I_k,$$

where A is defined as same as (5.7), λ is the regularization parameter on the covariance matrix, X_G denotes for the centered data in $\mathcal{F}$ obtained by the feature

map $\phi(\cdot)$ as $X_G = [\phi(\vec{x}_1) - \vec{\mu}^{\Phi}, \phi(\vec{x}_2) - \vec{\mu}^{\Phi}, ..., \phi(\vec{x}_N) - \vec{\mu}^{\Phi}]$, where $G = PKP$, $P = I_N - (1/N)\vec{1}_N\vec{1}_N^T$ and $K(\vec{x}_u, \vec{x}_v) = \phi(\vec{x}_u)^T\phi(\vec{x}_v)$. Unfortunately, the objective in (5.13) is not explicitly computable because $\phi(\vec{x})$ may have infinite dimensions, thus the covariance matrix $X_G X_G^T$ is a infinite size matrix. To tackle this flaw, NAML algorithm involves linear discriminant analysis (LDA) to find a linear projection $P \in \mathbb{R}^{m\times d}$ that maps $\phi(\vec{x})$ from the infinite dimensional space in $\mathcal{F}$ to a d-dimensional space $\mathcal{D}$. According to the Representer Theorem [150], the optimal projection P is in the span of the images of data points in $\mathcal{F}$ as $P = X_G Q$, where $Q \in \mathbb{R}^{N\times d}$. Incorporating P in the objective of K-means, (5.13) is extended as

$$\max_{A,Q} \quad \mathcal{J} = \text{trace}\left(A^T GQ\left(Q^T(GG + \lambda G)Q\right)^{-1} Q^T GA\right), \tag{5.14}$$

$$\text{s.t.} \quad A^T A = I_k,$$

which now becomes explicitly computable because X_G is transformed as G on the basis of kernel trick. However, to solve (5.14) one needs to optimize A and Q iteratively. To incorporate multiple centered kernel matrices $G_1, ..., G_p$, in NAML, the objective in (5.14) is further extended as

$$\max_{A,Q,\vec{\theta}} \quad \mathcal{J} = \text{trace}\left(A^T \mathbf{\Omega} Q\left(Q^T(\mathbf{\Omega}\mathbf{\Omega} + \lambda\mathbf{\Omega})Q\right)^{-1} Q^T \mathbf{\Omega} A\right), \tag{5.15}$$

$$\text{s.t.} \quad A^T A = I_k,$$

$$\mathbf{\Omega} = \sum_{r=1}^{p} \theta_r G_r,$$

$$\theta_r \geq 0, \quad \sum_{r=1}^{p} \theta_r = 1, \quad r = 1, ..., p,$$

which is a highly non-convex objective to optimize A, Q, and $\vec{\theta}$ simultaneously. To solve this, Chen *et al.* apply an alternative minimization framework [42] to optimize A, Q, and $\vec{\theta}$ iteratively in a three-level optimization procedure. Comparing with NAML, the proposed objective in (12) only optimize A and $\vec{\theta}$, which is also a non-convex problem to solve A and $\vec{\theta}$ simultaneously. To solve this, we can also apply the alternative minimization technique to solve A and $\vec{\theta}$ iteratively as a bi-level optimization procedure.

5.4 Bi-level optimization of K-means on multiple kernel matrices

To solve the objective in the problem Q5.1, in the first phase we minimize $\mathcal{J}_1$ with respect to A, keeping $\vec{\theta}$ fixed. In the second phase we minimize $\mathcal{J}_1$ with respect to $\vec{\theta}$, keeping A fixed. The two-stage optimization is then repeated until convergence (the proof will be shown later). When $\vec{\theta}$ is fixed, the problem is exactly the K-means clustering problem as discussed before. When A is fixed, the problem of maximizing $\mathcal{J}_1$ reduces to the optimization of the coefficients θ_r assigned on kernel matrices with the given cluster memberships. We will show that when the memberships are given, the problem in Q_1 can be formulated as Fisher discriminant analysis in $\mathcal{F}$ (KFDA) [123].

5.4.1 The role of cluster assignment

In problem Q5.1, when we minimize $\mathcal{J}_1$ with respect to A with the fixed $\vec{\theta}$, the obtained $N \times k$ weighted cluster indicator matrix A can also be regarded as the one-versus-all coding of the cluster assignments, where the number of columns is equal to the cluster number. Each element at i, j in A represents whether the i-th data belongs to the j-th cluster. For example, if ten data samples $X = \{\vec{x}_i\}_{i=1}^{10}$ are clustered into three classes and the labels are $Y = [1, 1, 1, 2, 2, 2, 3, 3, 3, 3]^T$, the indicator matrix is given by

$$A = \begin{bmatrix} 0.5774 & 0 & 0 \\ 0.5774 & 0 & 0 \\ 0.5774 & 0 & 0 \\ 0 & 0.5774 & 0 \\ 0 & 0.5774 & 0 \\ 0 & 0.5774 & 0 \\ 0 & 0 & 0.5000 \\ 0 & 0 & 0.5000 \\ 0 & 0 & 0.5000 \\ 0 & 0 & 0.5000 \end{bmatrix},$$

where the coding of the three classes is equivalent to

$$Y(1) = \begin{bmatrix} 0.5774 & 0 & 0 \end{bmatrix},$$

$$Y(2) = \begin{bmatrix} 0 & 0.5774 & 0 \end{bmatrix},$$

$$Y(3) = \begin{bmatrix} 0 & 0 & 0.5000 \end{bmatrix}.$$

The elements in A is similar to the coding scheme of labeled data in the multi-class learning because each column of A actually distinguishes one cluster from other clusters. When A is given, the between-scatter matrix S_b^Φ and within-scatter matrix S_w^Φ are fixed, thus the problem of optimizing the coefficients of multiple kernel matrices is equivalent to optimizing the kernel Fisher discriminant using multiple kernel matrices. The scatter matrices of KFDA is determined by the cluster assignments of data points, which is obtained via an affinity function using A as the input as

$$F_{ij} == \begin{cases} +1 & \text{if } A_{ij} > 0 \\ -1 & \text{if } A_{ij} = 0 \end{cases}. \tag{5.16}$$

In the second iteration of problem Q5.1, to minimize $\mathcal{J}_1$ with respect to $\vec{\theta}$ with the fixed A, we formulate it as the optimization of KFDA [123] on multiple kernel matrices using the transformed cluster assignment matrix F as the input parameter.

5.4.2 Optimizing the kernel coefficients as KFDA

As known, for a single data set, to optimize the linear discriminant in $\mathcal{F}$ for binary cluster assignments, one need to maximize

$$\max_{\vec{w}} \mathcal{J} = \frac{\vec{w}^T S_b^\Phi \vec{w}}{\vec{w}^T (S_t^\Phi + \kappa I)\vec{w}}, \tag{5.17}$$

where $\vec{w}$ is the norm vector of the separating hyperplane in $\mathcal{F}$, S_b^Φ and S_t^Φ are respectively the between-class and the total scatter matrices in $\mathcal{F}$, κ is the regularization term to ensure the positive definiteness of the denominator. It is also possible to use the within-class scatter matrix S_w^Φ in the denominator, but we will soon show that the formulation based on S_t^Φ can bring us computational convenience. For k clusters, denote $W = [\vec{w}_1, ..., \vec{w}_k]$ as the matrix where each column corresponds to the discriminative direction of one-vs-others (1vsA) clusters, the objective can be extended as [117][55]

$$\max_{W} \mathcal{J} = \text{trace} \frac{W^T S_b^\Phi W}{W^T (S_t^\Phi + \kappa I) W}. \tag{5.18}$$

Since the discriminant analysis is invariant to the magnitude of $\vec{w}$, we assume $\vec{w}^T\vec{w} = 1$ or $W^T W = I$, the problem in (5.18) leads to a generalized eigenvalue problem and the columns of the optimal W^* are the generalized eigenvectors that correspond to the largest eigenvalues of $(S_t^\Phi + \kappa I)^{-1} S_b^\Phi$ [55][117] (more formal mathematical proofs available in [98, 129]). In particular, in clustering problem if

the kernel matrices are all centered in $\mathcal{F}$, the elements in S_t^Φ are all 0, the term $S_t^\Phi + \kappa I$ becomes κI, thus the KFDA problem in (5.18) is simplified as

$$\begin{aligned} \max_W \mathcal{J} = \ & \text{trace}\frac{W^T S_b^\Phi W}{W^T W}, \\ \text{s.t.} \ & W^T W = I, \end{aligned} \tag{5.19}$$

which is exactly a Rayleigh quotient problem as the objective of K-means in (5.9). Collectively, the objective of K-means using multiple kernel matrices is formulated as

$$\boxed{\text{Q5.2:}} \qquad \begin{aligned} \max_{A,W,\vec{\theta}} \mathcal{J} = \ & \text{trace}\frac{W^T A^T \mathbf{\Omega} A W}{W^T A^T L W}, \\ \text{s.t.} \ & A^T A = I, \\ & W^T W = I, \\ & \mathbf{\Omega} = \sum_{r=1}^{p} \theta_r G_r, \\ & \theta_r \geq 0, \ r = 1, ..., p, \\ & \sum_{r=1}^{p} \theta_r = 1. \end{aligned} \tag{5.20}$$

In the K-means step, we set $W = I$ and optimize the L; In the KFDA step, we fix the L and optimize W and $\vec{\theta}$. Therefore, the two components optimize towards the same objective as a Rayleigh quotient in $\mathcal{F}$, which also guarantee that the iterative optimization of these two steps will necessary converge to a local optimum. Moreover, in the KFDA step of the present problem, we are not interested in W which determines the separating hyperplane, instead, we only need the optimal coefficients θ_r assigned on the multiple kernel matrices. The problem in of optimizing W and $\vec{\theta}$ in KFDA is related to the *multiple kernel learning* (MKL) approach proposed for supervised learning. In particular, it is known that FDA is related to the least squares approach [55], and the KFDA is actually a LS-SVM [166] problem as discussed in the previous chapters.

5.4.3 Regularization term

As shown in the LS-SVM MKL, the regularization term can be estimated jointly as a kernel coefficient assigned to an identify matrix. Analogously, we can also

estimate the λ value in clustering as

$$\tilde{\mathbf{\Omega}} = \left\{ \sum_{j=1}^{p+1} \theta_i G_j \middle| \sum_{j=1}^{p+1} \theta_j = 1,\ \theta_j \geq 0,\ G_{p+1} = I \right\}. \tag{5.21}$$

In this way, we avoid the challenge of estimating the hyper-parameter on unlabeled data.

5.4.4 Soft clustering

For LS-SVM MKL in (3.28) and (3.29), the Lagrangian parameters β are unconstrained and the target labels in Y can be positive or negative values. The coding scheme of Y only influences the value of β and the bias term b, but has no effect on the kernel coefficient θ. Therefore, the optimization solutions is independent to the scales and the signs of codings in A. The restriction $A^T A = I$ can be relaxed to that $A^T A$ only needs to be a diagonal matrix as

$$A = \begin{bmatrix} y_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & y_k \end{bmatrix}, \tag{5.22}$$

where the functions of class labels $y_1 \neq y_2, ..., \neq y_k$ should be different. Moreover, (5.22) can be further relaxed to allow $A^T A$ as non-diagonal, which is equivalent to consider the "overlapping" or "soft" label assignments in clustering. To extend the objective in (5.20) for soft clustering, one may replace the hard K means clustering (KM) clustering component with a soft clustering algorithm.

5.5 Optimized data fusion for kernel K-means clustering

Now we have clarified the two algorithmic components to optimize the K-means objective as the function of multiple kernel matrices. The main advantage is that the coefficients of kernel matrices are optimized automatically which leverages the effect of different kernel matrices in data integration. Collectively, we call the

propose algorithm as OKKC and its pseudocode is presented in Algorithm 5.5.1 as follows.

Algorithm 5.5.1: OKKC$(G_1, G_2, ..., G_p, K, \varepsilon)$

comment: Obtain the $\mathbf{\Omega^{(0)}}$ by the initial guess of $\theta_1^{(0)}, ..., \theta_p^{(0)}$

$A^{(0)} \leftarrow$ KERNEL K-MEANS $(\mathbf{\Omega^{(0)}}, K)$

$\tau = 0$

while $(\Delta A > \varepsilon)$

do $\begin{cases} step1 : F^{(\tau)} \leftarrow A^{(\tau)} \\ step2 : \mathbf{\Omega}^{(\tau+1)} \leftarrow \text{SIP-LS-SVM-MKL}(G_1, G_2, ..., G_p, F^{(\tau)}) \\ step3 : A^{(\tau+1)} \leftarrow \text{KERNEL K-MEANS}(\mathbf{\Omega}^{(\tau+\mathbf{1})}, K) \\ step4 : \Delta A = ||A^{(\tau+1)} - A^{(\tau)}||^2 / ||A^{(\tau+1)}||^2, \quad \tau = \tau + 1 \end{cases}$

return $(A^{(\tau+1)}, \theta_1^{(\tau+1)}, ..., \theta_p^{(\tau+1)})$

In our implementation, the Kernel K-means clustering component is based on Girolami's algorithm [65]. To make the algorithm applicable on large scale data, the LS-SVM MKL component is based on SIP formulation. The iteration in algorithm terminates when the cluster membership matrix A stops changing. For large scale data, the K-means is not stable thus A may change even when the identical $\mathbf{\Omega}$ is used for clustering. We can set a tolerance value ε to determine the convergence of A. In our implementation, ε is set to 5E-2. In specific cases, we can also replace the Kernel K-means with some enhanced versions to improve the stability of clustering.

5.6 Experimental result

5.6.1 Synthetic data sets

The experiment is carried on three synthetic data sources with intuitive patterns. As shown in Figure 5.1, we assume that the 400 samples are represented in three different data sources. In the normal data, the two clusters form two normal distributions with a certain degree of overlapping. In the ring data, the two clusters show highly nonlinear distributions. In the random data, the labels are randomly assigned to simulate irrelevant information. Apparently, the normal data contains the most informative partition in linear space while the ring data contains obvious nonlinear partitions. We apply K-means algorithm (set $K = 2$) to partition on the three data sources in the linear space. To obtain the nonlinear clustering, we construct kernel matrices on the data sources using two Radial Basis

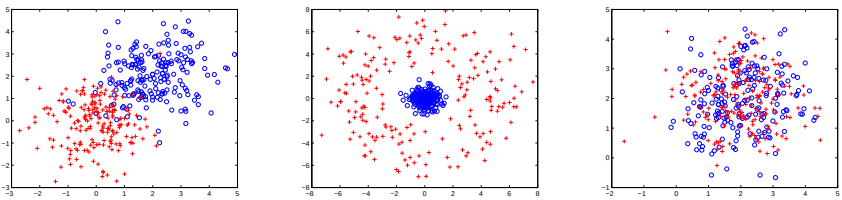


Figure 5.1: Synthetic data sets used in the experiments. The data is visualized in a 2-D space, where the horizontal labels represent x_1 and the vertical labels represent x_2. From the left to the right: Normal, Ring and Random data. Each data source contains 400 samples which are equally divided into 2 groups.

Functions (RBF) functions with two different widths. The width of the first RBF function is selected empirically as twice of the average distance of a data point to its nearest neighbor in the whole data source (denoted as RBF×1). For the second RBF function, twice of the width is taken [46] (denoted as RBF×2). Then we apply the kernel K-means algorithm [65] to obtain the partitions from the kernel matrices. The clustering results are validated by the Rand Index (RI) [82] and the Normalized Mutual Information (NMI) [162] by comparing the automatic clustering partitions with the labels of the samples.

We fuse the data sources using OKKC algorithm, firstly in the linear space then in the nonlinear space. In the linear space fusion, we combine the three data sources as three linear kernel matrices. Since the synthetic data is generated without natural dependency among the samples, we assume that all the samples within the same group are equivalent with each other. Thus, in each run of the clustering, we randomly permute the samples within each label and repeat the clustering 20 times. The clustering performance obtained from the individual data source and the data fusion approach is shown in Figure 5.2 (figures on the top). The OKKC is almost the same as the best individual data set. The nice performance is attributed to the dominant weights assigned on the normal data (0.9999).

Next, we combine the RBF kernel matrices and the linear kernel matrices. Nine kernel matrices (one linear and two RBF for each data set) are combined for clustering. The performance obtained by the individual data and the OKKC is shown in Fig 5.2 (figures on the bottom). The OKKC still performs well because the good RBF kernel matrices are assigned with dominant weights (normal linear: 0.0002, normal RBF1: 0.0000, normal RBF2: 0.0065, ring linear: 0.0060, ring RBF1: 0.0000, ring RBF2: 0.8631, rand linear: 0.0368, rand RBF1: 0, rand RBF2:0.0875). The convergence rate of the OKKC is also illustrated in the Figure 5.3.

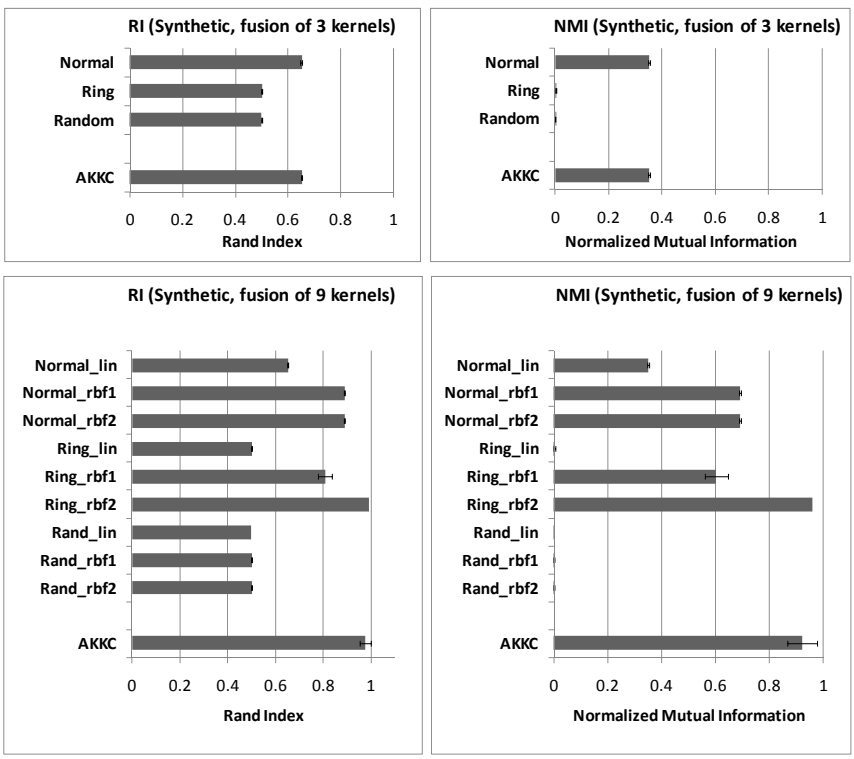


Figure 5.2: Performance of OKKC compared to clustering by individual data source. Top figures: comparison of 3 linear kernel matrices. Bottom figures: comparison of 9 kernel matrices. Algorithmic parameters: $\lambda = 0.001$, tol=0.0001. The error bars are standard deviations of performance evaluated in 20 random repetitions.

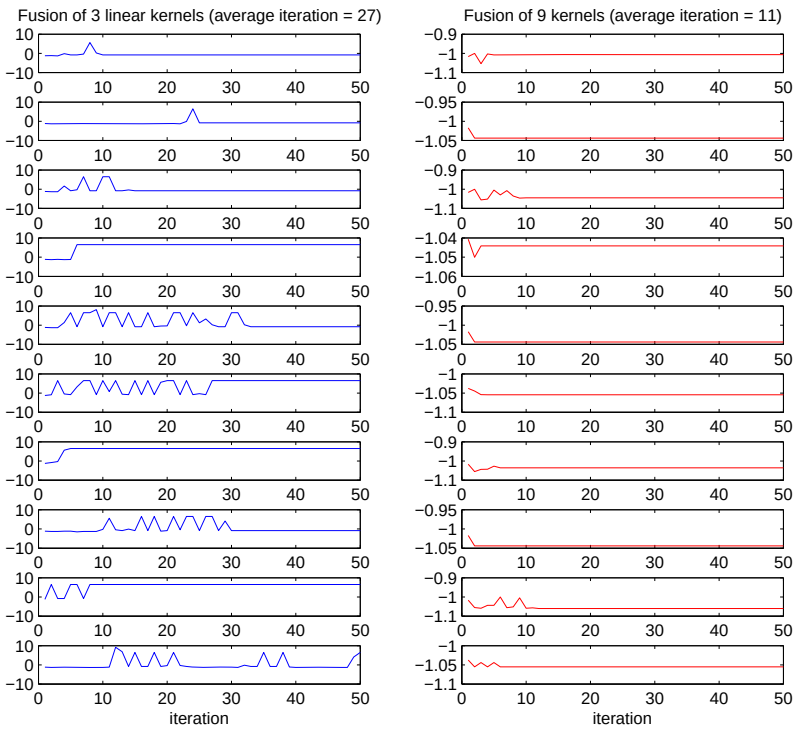


Figure 5.3: Convergence of OKKC when combining 3 synthetic data sources. Left: 10 different repetitions of OKKC when combining 3 linear kernel matrices. Right: 10 repetitions of OKKC when combining 9 kernel matrices. When combining 3 kernel matrices, OKKC takes more iterations to converge (averagely 27 iterations). When combining 9 kernel matrices, it takes averagely 11 iterations to converge. Algorithmic parameters: $\lambda = 0.001$, tol=0.0001.

5.6.2 UCI machine learning data sets

We adopt five data sets from UCI machine learning repository: Iris, Wine, Yeast, Satimage and Pen digit recognition. The original Satimage and Pen digit data contain large amount of data points, so we sample 80 data points from each class and construct the data sets for experimental comparison. For each data set, we generate ten RBF kernel matrices using different kernel widths. We denote the average covariance of data sample as σ, the 10 different kernel widths are $\{\frac{1}{4}\sigma, \frac{1}{2}\sigma, \sigma, ..., 7\sigma, 8\sigma\}$. We apply three different algorithms (OKKC,Nonlinear Adaptive Metric Learning (NAML) and the average combination of kernel matrices) to combine the ten RBF kernel matrices for clustering. The summary of data sets is presented in Table 5.1. The experimental results are presented in Table 5.2.

Table 5.1: Summary of the benchmark data sets

Data set	Dimension	Instance	Class
iris	4	150	3
wine	13	178	3
yeast	17	384	5
satimage	36	480	6
pen digit	16	800	10

As shown, on all the data sets, the performance obtained by OKKC is comparable to the results of the best individual kernel matrices. On Wine and Pen digit data, OKKC performs significantly better than NAML. Moreover, on most of the data sets, OKKC costs less computational time than NAML because of its bi-level optimization procedure has less computational burden than the three-level procedure of NAML. In particular, the QCQP formulation of MKL becomes challenging when the number of data points, clusters and kernels are all large. For Pen digit data set, the problem causes memory overflow on the laptop computer and we have to solve it on a Unix sever with 7G memory. In contrast, the SIP formulation significantly reduces the computational burden of MKL so the OKKC is able to solve the problem on a Laptop computer. We apply paired t-test as significant test to compare the performance of 20 repetitions and the results are illustrated in Figure 5.3.

Table 5.2: Overall results of clustering using the combination of ten RBF kernel matrices for each data set. For each data set, we present the best performance of clustering obtained on a single RBF kernel, the performance obtained on average combination of ten RBF kernel matrices, OKKC and NAML. The clustering performance is evaluated by ARI and NMI. The regularization term λ in OKKC is estimated jointly as the coefficient of an identity matrix. We try ten different values from 2^{-5} to 2^4 as the regularization term λ in NAML and the performance is not very sensitive to λ. In this table, the highest performance of ten different regularization terms is presented. For each data set, the clustering is repeated 20 times and the average values and the standard deviations (in parentheses) are shown. The average numbers and deviations of iterations in OKKC and NAML are also calculated from the 20 repetitions. The computational time (seconds) is benchmarked using Matlab v7.6.0 on Windows XP SP2 installed on a Laptop computer with Intel Core 2 Duo 2.26GHz CPU and 2G memory. The optimization of OKKC is solved by Matlab built-in functions. The optimization of NAML is solved by quadratic programming using MOSEK [8]. For Pen digit data, the QCQP problem has memory issue and cannot be solved on the Laptop computer.

	best individual		average combine			OKKC				NAML			
	ARI	NMI	ARI	NMI	time	ARI	NMI	itr	time	ARI	NMI	itr	time
Iris	0.7177 (0.0734)	0.7583 (0.0209)	0.6903 (0.1072)	0.7553 (0.0382)	0.22 (0.13)	0.7351 (0.0671)	0.7710 (0.0457)	7.8 (3.7)	5.32 (2.46)	0.7459 (0.0210)	0.7729 (0.0147)	9.2 (2.5)	15.45 (6.58)
Wine	0.3424 (0.0822)	0.3714 (0.0651)	0.3188 (0.1264)	0.3343 (0.1078)	0.25 (0.03)	0.3782 (0.0547)	0.3955 (0.0527)	10 (4.0)	18.41 (11.35)	0.2861 (0.1357)	0.3053 (0.1206)	6.7 (1.4)	16.92 (3.87)
Yeast	0.4297 (0.0525)	0.5027 (0.0218)	0.4193 (0.0529)	0.4994 (0.0271)	2.47 (0.05)	0.4126 (0.0437)	0.4839 (0.0206)	7 (1.7)	81.85 (14.58)	0.4256 (0.0456)	0.4998 (0.0188)	10 (2)	158.20 (30.38)
Satimage	0.4765 (0.0515)	0.5922 (0.0383)	0.4847 (0.0575)	0.5994 (0.0324)	4.54 (0.07)	0.4996 (0.0571)	0.6004 (0.0415)	10.2 (3.6)	213.40 (98.70)	0.4708 (0.0488)	0.5857 (0.0314)	8 (0.7)	302 (55.65)
Pen digit	0.5818 (0.0381)	0.7169 (0.0174)	0.5880 (0.0531)	0.7201 (0.0295)	15.95 (0.08)	0.5904 (0.0459)	0.7461 (0.0267)	8 (4.38)	396.48 (237.51)	0.5723 (0.0492)	0.7165 (0.0295)	8	overflow

Table 5.3: Significance test of clustering performance. The p values shown in the table are obtained from the paired t-test comparing the performance in 20 repetitions. When the null hypothesis is rejected, "+" means the performance of OKKC is higher than the comparing approaches. "-" means its performance is lower.

data	OKKC vs. single		OKKC vs. NAML		OKKC vs. average	
	ARI	NMI	ARI	NMI	ARI	NMI
iris	0.2213	0.8828	0.7131	0.5754	0.2282	0.9825
wine	0.2616	0.1029	0.0085 (+)	0.0048 (+)	0.0507	0.0262 (+)
yeast	0.1648	0.0325 (-)	0.1085	0.0342 (-)	0.2913	0.0186 (-)
satimage	0.1780	0.4845	0.6075	0.8284	0.5555	0.9635
pen	0.0154 (+)	0.2534	3.88e-11 (+)	3.71e-04 (+)	0.4277	0.0035 (+)

5.7 Summary

In this chapter we presented the OKKC algorithm for kernel fusion based clustering. The proposed algorithm is constructed on an alternative minimization framework and the two components are optimizing towards the same Rayleigh quotient objective. As shown in a synthetic example, the main advantage of the weighted combination is the usefulness to distinguish informative sources from uninformative sources. The kernel fusion component in OKKC is based on the Kernel Fisher Discriminant Analysis (KFDA), which has been shown as equivalent to the LS-SVM MKL in Chapter 3 and Chapter 4. The efficiency of LS-SVM MKL enables us to apply OKKC on large scale problems. We also discussed some further advances of OKKC, such as the extension towards fuzzy clustering, the extension to non-sparse kernel fusion, etc. We also theoretically and empirically compared our proposed algorithm with NAML and it shows that the bi-level optimization procedure of OKKC is much efficient than NAML. In some specific problems, the OKKC performs significantly better than the NAML.

Chapter 6

Optimized data fusion for K-means Laplacian Clustering

6.1 Introduction

In the preceding chapter, we have discussed an algorithm to combine K-means clustering with kernel fusion. K-means is proposed to cluster attribute-based data, which the goal is to partition data into K numbers of clusters with the minimal distortion measure. Another well known method, spectral clustering [91] [136] [156], is also widely adopted in many applications. Unlike K-means, spectral clustering is specifically developed for graphs, where the data samples are represented as vertices connected by non-negative weighted undirected edges. The problem of clustering on graphs belongs to another paradigm than the algorithms based on the distortion measure. The goal of graph clustering is to find partitions on graph such that the edges between different groups have a very low weight [181]. To model this, different objective functions are adopted and the typical criteria include the RatioCut [70], the normalized cut [156], and many others. To solve these objectives, the discrete constraint of the clustering indicators is usually relaxed to real values thus the approximated solution of spectral clustering can be obtained from the eigenspectrum of the graph Laplacian matrix [7]. Many investigations [51] have shown the connection between K-means and spectral clustering. Moreover, in practical applications, the weighted similarity matrix is often used interchangeably as the kernel matrix in K-means or, in spectral clustering, as the affinity matrix to construct the Laplacian.

Recently, a new algorithm, KL, is proposed to combine a kernel and a Laplacian simultaneously in clustering analysis [183]. This method combines the objectives

of K-means and spectral clustering in a quotient trace maximization form and solves the problem via eigen-decomposition. This integration, is shown empirically outperforming its K-means and spectral components on real data sets. Though the idea seems straightforward, the aggregated algorithm may settle many practical problems, especially those pertaining to combine attribute-based data with interaction-based networks. For example, in web analysis and scientometrics, the combination of text mining and bibliometrics has become a standard approach in clustering science or technology fields towards the detection of emerging fields or hot topics [84]. In bioinformatics, the protein-protein interaction networks and the gene expression data sets are two of the most important sources used to reveal the relevance of genes and proteins with diseases. Conventional methods usually transform all of them as kernels or graphs and consequently applied KM or spectral clustering on the combined matrices. However, the potential disadvantage is that the attribute kernels are often very dense whereas the interaction graphs are usually sparse; integrating these two representations directly may overwhelm the sparse structure of the interaction data. This difficulty is tackled in KL clustering as integrating the objectives of KM and Spectral Clustering (SC). However, the performance of KL is still strongly dependent on the quality of the kernel and the Laplacian, which results in the model selection problem to determine the optimal parameters of the kernel and the Laplacian. The strategy presented in this chapter applies kernel fusion technique to incorporate multiple kernels and Laplacians in KL clustering, which is a novel problem and no existing method has been proposed in the literature before. We propose an algorithm, OKLC to learn *the optimal combination of kernel* and *the optimal combination of Laplacian*.

6.2 Clustering by multiple kernels and Laplacians

6.2.1 Background of KL clustering

Let us denote X as the attribute data set and W as the graph affinity matrix, , we assume both of them are representations of the same sets of samples. The objective of the KL clustering is briefly defined as

$$J_{KL} = \kappa J_{Ncut} + (1-\kappa) J_{K-means}, \tag{6.1}$$

where κ is the coefficient adjusting the importance of K-means and spectral clustering in the overall objective. Denote A as the weighted cluster indicator matrix $A = [\vec{a}_1, \vec{a}_2, ..., \vec{a}_k]$ defined as (5.9), D as the diagonal matrix whose (i,i) entry is the sum of the entries of row i in the affinity matrix W. The *normalized Laplacian matrix* [181] is given by $\tilde{L} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. The objective of normalized cut based spectral clustering is formulated as

$$\min_{A} \operatorname{trace}\left(A^T \tilde{L} A\right). \tag{6.2}$$

As discussed, the objective of the K-means is given by

$$\max_{A} \text{trace}(A^T X^T X A). \tag{6.3}$$

Analogous to the discussions presented in the previous chapter, the objective in (6.3) can be extended to the Hilbert space as

$$\max_{A} \text{trace}\left(A^T X^{\Phi T} X^{\Phi} A\right). \tag{6.4}$$

where X^{Φ} is defined as (5.10). Then the objective of KL becomes

$$\begin{aligned} \min_{A} \quad & \text{trace}\left(A^T \tilde{L} A\right) + (1-\kappa) \max_{A} \text{trace}\left(A^T G A\right) \\ \text{s.t.} \quad & A^T A = I_k, \end{aligned} \tag{6.5}$$

where G is the centered kernel matrix. To solve the optimization problem without tuning κ, Wang *et al.* formulate it as a trace quotient of the two components [183]. The trace quotient is then further relaxed as a maximization of quotient trace, given by

$$\begin{aligned} \max_{A} \quad & \text{trace} \ \frac{A^T G A}{A^T \tilde{L} A} \\ \text{s.t.} \quad & A^T A = I_k. \end{aligned} \tag{6.6}$$

As mentioned in Chapter 2, the problem in (6.6) is a generalized Rayleigh quotient and the optimal solution A^* is obtained in the generalized eigenvalue problem. To maximize this objective, A^* is approximated as the largest k eigenvectors of $\tilde{L}^+ G$, where $\tilde{L}^+$ is the pseudo inverse of $\tilde{L}$ [183].

6.2.2 Combining multiple kernels and multiple Laplacians

Let us denote a set of graphs as H_μ, $\mu \in 1, ..., r$, all having n vertices, and a set of Laplacians $\tilde{L_\mu}$ constructed from H_μ. Let us also denote a set of centered kernel matrices as G_ν, $\nu \in 1, ..., s$ with n samples. To combine these kernels and Laplacians for clustering, we propose a strategy to learn their optimal convex linear

combinations[1] for KL as follows:

$$\boxed{\text{Q6.1:}} \qquad \max_{A,\alpha_\mu,\beta_\nu} \mathcal{J} = \text{trace}\, \frac{A^T \mathbf{\Omega} A}{A^T \tilde{\mathrm{L}} A} \tag{6.7}$$

$$\text{s.t. } \tilde{\mathrm{L}} = \left\{ \sum_{\mu=1}^{r} \alpha_\mu \tilde{L}_\mu \Big| \sum_{\mu=1}^{r} \alpha_\mu = 1, \alpha_\mu \geq 0 \right\},$$

$$\mathbf{\Omega} = \left\{ \sum_{\nu=1}^{s} \beta_\nu G_\nu, \Big| \sum_{\nu=1}^{s} \beta_\nu = 1, \beta_\nu \geq 0 \right\},$$

$$A^T A = I_k.$$

In Q6.1, to solve the optimal $A^*, \alpha_\mu^*, \beta_\nu^*$ simultaneously is difficult, alternatively, it can be solved via alternating minimization [42]. In the first phase we minimize $\mathcal{J}$ with respect to β_ν, keeping A, α_μ fixed. In the second phase we minimize $\mathcal{J}$ with respect to α_μ, keeping A, β_ν fixed. In the third phase, we minimize $\mathcal{J}$ with respect to A, keeping α_μ, β_ν fixed. This three-stage optimization is then repeated until convergence. When A and α_μ are fixed, the objective of learning the optimal combination of kernel matrices is given by

$$\max_{\beta_\nu} \text{ trace} \left(U^{-\frac{1}{2}} A^T \mathbf{\Omega} A U^{-\frac{1}{2}} \right) \tag{6.8}$$

$$\text{s.t. } \mathbf{\Omega} = \left\{ \sum_{\nu=1}^{s} \beta_\nu G_\nu \Big| \sum_{\nu=1}^{s} \beta_\nu = 1, \beta_\nu \geq 0 \right\},$$

$$A^T A = I_k,$$

where U is equivalent to $A^T \tilde{\mathrm{L}} A$, and $\tilde{\mathrm{L}}$ is the combined Laplacians with fixed weights α_μ. $U^{-\frac{1}{2}}$ is a symmetric matrix, denote $S = AU^{-\frac{1}{2}}$, (6.8) becomes

$$\max_{\beta_\nu} \text{ trace} \left(S^T \mathbf{\Omega} S \right) \tag{6.9}$$

$$\text{s.t. } \mathbf{\Omega} = \left\{ \sum_{\nu=1}^{s} \beta_\nu G_\nu \Big| \sum_{\nu=1}^{s} \beta_\nu = 1, \beta_\nu \geq 0 \right\},$$

$$A^T A = I_k,$$

which is equivalent to the kernel fusion step of the OKKC algorithm presented in the previous chapter. S is a fixed weighted cluster indicator matrix, which

[1]For normalized Laplacians, the smallest eigenvalue of the combined Laplacians $\tilde{\mathrm{L}}$ is a small positive number thus $\tilde{\mathrm{L}}$ is invertible. Though there is no eigenvalue pair $(0,\vec{1})$ in $\tilde{\mathrm{L}}$, the objective for spectral clustering still holds and the partitions are obtained by applying K-means on the K smallest eigenvectors of $\tilde{\mathrm{L}}$

is equivalent to the coding of class labels in the MKL learning. Though S is not necessarily orthogonal, the objective still holds because the discrete cluster assignment task has already been relaxed as real values and S represents the "soft" cluster assignment of data samples. Based on the previous discussions, (6.9) can be solved in its dual form, for example, as a QCQP problem given by

$$\begin{aligned} \min_{\vec{\eta}_j, t} \quad & \sum_{j=1}^{k} \frac{1}{2} \vec{\eta}_j^T \vec{\eta}_j + \frac{1}{2\lambda} t - \sum_{j=1}^{k} \vec{\eta}_j^T \vec{s}_j \\ \text{s.t.} \quad & t \geq \sum_{j=1}^{k} \vec{\eta}_j^T G_\nu \vec{\eta}_j, \quad \nu = 1, \ldots, s, \end{aligned} \tag{6.10}$$

where $\vec{s}_j$ is the j-th column of S, $\vec{\eta}$ are the dual variables. Since (6.10) is unconstrained, it can be solved more efficiently as iteration of linear systems as SIP-LS-SVM-MKL proposed in Algorithm 4.3.2. When A and β_ν are fixed, we set $V = A^T \mathbf{\Omega} A$, the overall objective (6.7) becomes

$$\begin{aligned} \max_{\alpha_\mu} \quad & \text{trace} \left(\left(A^T \sum_{\mu=1}^{r} \alpha_\mu \tilde{L_\mu} A \right)^{-1} V \right) \\ \text{s.t.} \quad & \sum_{\mu=1}^{r} \alpha_\mu = 1, \quad \alpha_\mu \geq 0, \quad A^T A = I_k. \end{aligned} \tag{6.11}$$

This optimization problem in (6.11) can be solved, for example, by gradient descent method with the derivative as

$$\frac{\partial}{\partial \alpha_\mu} = \text{trace} \left(-(A^T \sum_{\mu=1}^{r} \alpha_\mu \tilde{L_\mu} A)^{-1} A^T \tilde{L_\mu} A (A^T \sum_{\mu=1}^{r} \alpha_\mu \tilde{L_\mu} A)^{-1} V \right). \tag{6.12}$$

The problem in (6.11) is connected to Tsuda *et al.*'s approach on multiple network learning [174] and the solution is also similar. To avoid the sparse solution on coefficients, it is necessary to regularize the objective function to avoid only selecting a single kernel or Laplacian, or, to adopt the L_2 norm kernel fusion paradigm addressed in Chapter 3. In this specific algorithm, we set a minimal boundary θ_{min} as the constraint of weights so that $\alpha_\mu \geq \frac{\theta_{min}}{r}$ and $\beta_\nu \geq \frac{\theta_{min}}{s}$, where r and s are the number of Laplacians and kernels. The OKLC algorithm also has a free parameter λ which controls the regularization term of covariance matrix. In this version, λ is set as a small positive constant, whereas it can be estimated jointly in the kernel fusion step as (5.21). When α_μ and β_ν are fixed, the partition A is then solved by the KL algorithm taking $\tilde{\mathbf{L}}$ and $\mathbf{\Omega}$ as input. Putting all these steps together, we propose a new algorithm as the extension of

KL clustering algorithm to integrate multiple Laplacians and kernels, termed as OKLC. The pseudo code of the OKLC algorithm is presented in as follows:

Algorithm 6.2.1: OKLC$(\tilde{L_1}, \tilde{L_2}, ..., \tilde{L_r}, G_1, G_2, ..., G_s, k)$

$A^{(0)}, U^{(0)} \leftarrow \mathrm{KL}(L_0, G_0, \lambda)$
comment: obtain an initial partition from a random Laplacian and kernel
$S^{(0)} = A^{(0)} U^{(0)-\frac{1}{2}}$
$\tau = 0$
while $(\Delta A > \varepsilon)$
do
$step1 : F^{(\tau)} \leftarrow S^{(\tau)}$
$step2 : \mathbf{\Omega}^{(\tau)} \leftarrow$ SIP LS-SVM MKL$(G_1, G_2, ..., G_s, F^{(\tau)})$
$step3 : V^{(\tau)} = A^{(\tau)T} \mathbf{\Omega}^{(\tau)} A^{(\tau)}$
$step4 : Ł^{(\tau)} \leftarrow$ Laplacian Fusion$(\tilde{L_1}, \tilde{L_2}, ..., \tilde{L_r}, A^{(\tau)}, V^{(\tau)})$
$step5 : A^{(\tau+1)} \leftarrow \mathrm{KL}(Ł^{(\tau)}, \mathbf{\Omega}^{(\tau)}, k)$
$step6 : \Delta A = ||A^{(\tau+1)} - A^{(\tau)}||^2 / ||A^{(\tau+1)}||^2$
$step7 : \tau = \tau + 1$
return $(A^{(\tau+1)})$

The convergence of Algorithm 6.2.1 can be determined by comparing the update of A, if the update ration of A is smaller than the threshold ε, the algorithm stops iteration. In our implementation, the ε is set as 0.05.

6.3 Experiment

Table 6.1: Summary of the benchmark data sets

Data set	Dimension	Instance	Class
iris	4	150	3
promoter	57	106	2
wine	13	178	3
breast	9	699	2
yeast	17	384	5
satimage	36	480	6
pen digit	16	800	10

To empirically evaluate the performance of the proposed algorithm, we adopt six UCI [22] data sets in the experiment: iris, promoter, satimage, wine, yeast, and pen digit. For the satimage and the pen digit data, we randomly sample 80 instances from each class. Ten RBF kernel matrices and ten Laplacians are constructed on each data set with different kernel widths σ_j. Analogous to previous approaches, the σ_1 of the 1st RBF kernel matrix is set as the average distance to the nearest neighbor of all samples. Then for the 2nd kernel matrix and 3rd kernel matrix, twice and triple of this value is set and so on. These kernel matrices are also used as weighted similarity matrices to construct the normalized Laplacians. For the sequence based promoter data, we apply 2-mer, 3-mer, and 4-mer string kernels in combination with the normal sequence, 1 gap, and 1 wildcard configurations (9 combinations) [108]. The tenth kernel matrix for promoter data is constructed by the BLAST scores. For each data set, the proposed OKLC algorithm is applied to integrate all the ten kernel matrices and the ten Laplacians. The KL algorithm is applied to integrate each pair of the kernel and the Laplacian derived by the same σ, then the best result is presented for comparison. We also apply the kernel K-means (KM) on each individual kernel and the normalized cut spectral clustering (SC) on each individual Laplacian and select the best results for comparison. The performance is evaluated by the Rand Index (RI) and the Normalized Mutual Information (NMI). We also try 3 different minimal boundaries $\theta_{min} = 0, 0.5, 1$ to regularize the coefficients assigned on the kernels and the Laplacians. When $\theta_{min} = 1$, the kernels and Laplacians are combined in the average manner. In our experiment, we set the tolerance value of convergence to 10^{-4} and the OKLC usually converges rapidly within four or five iterations. The experimental results are presented in Table 6.2.

Table 6.2: Comparison of OKLC with the best performance of KL, KM, and SC on benchmark data sets. Mean values of 20 random repetitions are presented for comparison. The numbers in the parentheses represent the value of regularization parameter θ_{min}. The values in bold are the best performance obtained on each data set. The values in italic are the second best performance. In this experiment, for KL and OKLC algorithms, we set the regularization parameter explicitly as 10^{-4}. For each data set, the results shown in boldface are not significantly different from the best ones, judged by Wilcoxon rank-sum test, with a significance level of 0.05. The results shown in italic are the second best ones.

	data set	OKLC(0)	OKLC(0.5)	OKLC(1)	KL(best)	SC(best)	KM(best)
RI	iris	**0.8859**	**0.8923**	**0.8895**	**0.8816**	**0.8859**	**0.8825**
	promoter	*0.8567*	**0.9206**	0.6464	0.7547	0.6792	0.6570
	wine	*0.6399*	**0.7176**	*0.6462*	**0.7076**	**0.7144**	**0.7132**
	yeast	0.7839	*0.7917*	0.7759	**0.7951**	**0.8071**	**0.8046**
	satimage	*0.8263*	**0.8530**	0.8140	**0.8413**	**0.8499**	**0.8489**
	pen	*0.9203*	**0.9268**	0.9118	**0.9308**	**0.9246**	**0.9349**
NMI	iris	**0.7980**	**0.8057**	0.7403	**0.7930**	**0.7980**	*0.7733*
	promoter	*0.6816*	**0.7927**	0.2376	0.4148	0.2912	0.2455
	wine	*0.4081*	**0.4234**	0.2239	*0.3787*	**0.4206**	*0.3875*
	yeast	*0.4624*	**0.4957**	0.4445	**0.4973**	**0.5041**	**0.5102**
	satimage	*0.5496*	**0.6146**	0.5023	**0.5984**	**0.6074**	**0.6066**
	pen	**0.7367**	**0.7459**	0.6582	*0.7298*	0.6844	**0.7665**

As shown, the performance of the OKLC is comparable to the best KL, KM and SC results. Furthermore, the weighted integration of kernels and Laplacians outperforms the naive average combination, which shows the advantage and importance of tuning the weights in multiple source integration. To further demonstrate the advantage of the proposed algorithm, we conduct a sensitivity study to evaluate the quality of kernels and Laplacians. In Figure 6.1, we present the evaluation of clustering results of all the algorithms obtained by all the kernels and Laplacians. It is shown that the performance of KM, SC and KL relies much on the quality of the individual kernel and Laplacian. Unfortunately, in real life researches to identify the best candidate kernel or Laplacian is non-trivial. Furthermore, it is also hard to evaluate their quality on unlabeled data samples. OKLC provides a data fusion way to tackle this problem, which is more likely to perform as well as the best individual kernel and Laplacian.

The free parameter λ is used to improve the reliability of the estimation in the covariance matrix. In OKLC algorithm, λ is involved in two places: the kernel fusion procedure and the KL clustering procedure. In kernel fusion, the effect of λ is analyzed by Chen *et al.* and they find that λ is related to the sparseness of the optimization result [32]. In this chapter, we study the effect of λ by trying ten different values ($1,10^{-1},...,10^{-9}$) and the performance of OKLC is presented in Figure 6.1. As observed, in general, the performance of the OKLC is not very sensitive to the selection of λ. One exception is on the promoter data where the performance varies strongly when λ is approaching 1.

The question arises of what is the advantage of using a complicated algorithm, when it seems stable but there is no significant improvement over the single data set. This is probably because in this experiment the kernels and Laplacians are derived from the same similarity matrix, so the information is highly redundant. Therefore it is unlikely to gain significant improvement in KL or OKLC integration. In real world problems, to combine heterogeneous data representations obtained from different measurements and conditions would be more meaningful, and we will show a real application in Chapter 11 using OKLC to retrieve complementary information from multi-source and multi-representation data.

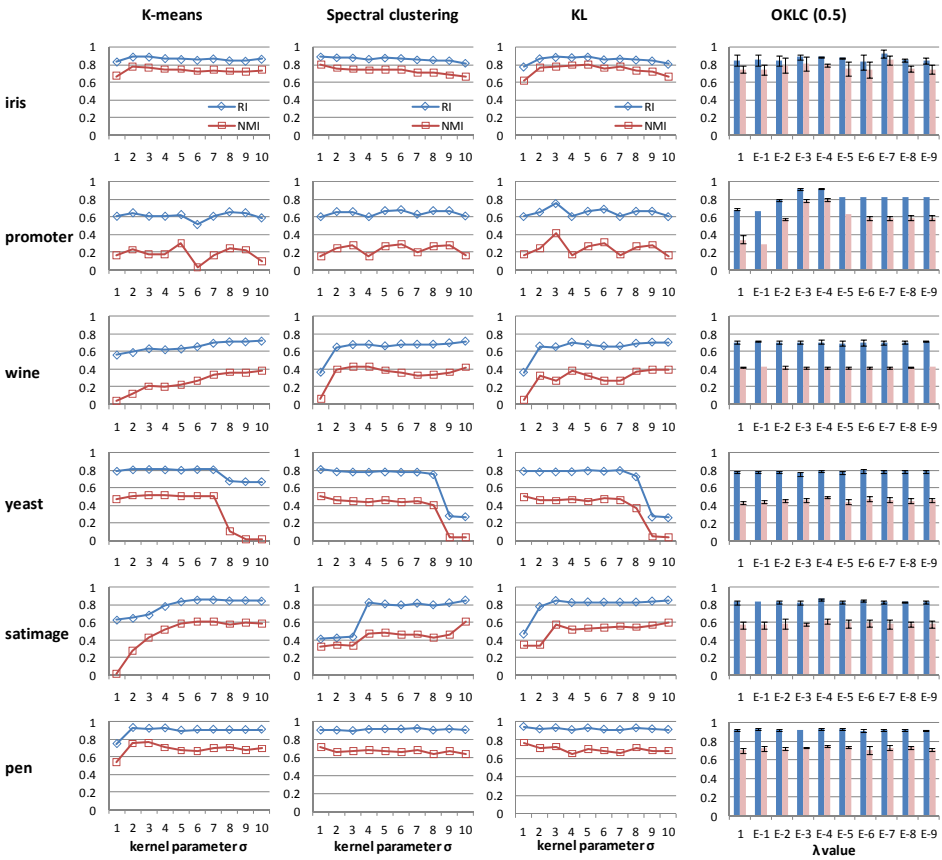


Figure 6.1: Comparison of the performance obtained on all the kernels and Laplacians. The mean values obtained from 20 random repetitions are shown in the figure. The regularization parameter of covariance matrix in KL is set as $\lambda = 10^{-4}$; the regularization parameter of kernel fusion for OKLC is set as $\theta_{min} = 0.5$.

Chapter 7

Weighted Multiple Kernel Canonical Correlation

7.1 Introduction

In the preceding two chapters we have presented two clustering algorithms using kernel fusion to combine multi-source and multi-representation of data. In this chapter we will investigate a different unsupervised learning problem, CCA, and its extension in kernel fusion techniques. The goal of CCA (taking two data sets for example) is to identify the canonical variables that minimize or maximize the linear correlations between the transformed variables [77]. The conventional CCA is employed on two data sets in the observation space (original space). The extension of CCA on multiple data sets is also proposed by Kettenring and it leads to different criteria of selecting the canonical variables, which are summarized as 5 different models: sum of correlation model, sum of squared correlation model, maximum variance model, minimal variance model and generalized variance model [93]. Akaho generalizes CCA by kernel methods to find the canonical variables of data sets in the Hilbert space [6]. Bach and Jordan further propose a kernel version of CCA on multiple data sets [13]. Multiple Kernel Canonical Correlation Analysis (MKCCA) is also useful as a independence measure to find uncorrelated variables in the Hilbert space [13]. In this chapter, we will show that MKCCA is also useful to extract common information through maximization of the pairwise correlations among multiple data sources. A weighted extension of MKCCA can be easily derived through modifications of the objective of MKCCA [194]. The weighted MKCCA method can also be extended to out-of-sample points, which becomes important for model selection. Another important issue for MKCCA is that the problem scales up exponentially with the number of incorporated data

sets and the number of samples. To make this method applicable on machines with standard CPU and memory, low rank approximation techniques based on ICD and Singular Value Decomposition (SVD) are introduced. Moreover, for the weighted extension of MKCCA, a incremental SVD algorithm is proposed to avoid recomputing eigenvalue decomposition each time when the weights of MKCCA are updated.

7.2 Weighted Multiple Kernel Canonical Correlation

7.2.1 Linear CCA on Multiple Data Sets

The problem of CCA consists in finding linear relations between two sets of variables [77]. For the problem of two variables $\vec{x}_1$ and $\vec{x}_2$ with zero means, the objective is to identify vectors $\vec{w}_1$ and $\vec{w}_2$ such that the correlation between the projected variables $\vec{w}_1^T\vec{x}_1$ and $\vec{w}_2^T\vec{x}_2$ is maximized, given by

$$\max_{\vec{w}_1,\vec{w}_2} \rho = \frac{\vec{w}_1^T C_{x_1x_2}\vec{w}_2}{\sqrt{\vec{w}_1^T C_{x_1x_1}\vec{w}_1}\sqrt{\vec{w}_2^T C_{x_2x_2}\vec{w}_2}}, \tag{7.1}$$

where $C_{x_1x_1} = \mathcal{E}[x_1x_1^T], C_{x_2x_2} = \mathcal{E}[x_2x_2^T], C_{x_1x_2} = \mathcal{E}[x_1x_2^T]$. Extending this objective function to multiple sets of variables $\vec{x}_1, \ldots, \vec{x}_m$, one obtains the form of multiple CCA, given by

$$\max_{\vec{w}_1,\ldots,\vec{w}_m} \rho = \frac{\mathcal{O}[\vec{x}_1, \ldots, \vec{x}_m]}{\prod_{i=1}^m \sqrt{\vec{w}_i^T C_{x_ix_i}\vec{w}_i}}, \tag{7.2}$$

where $\mathcal{O}[\vec{x}_1, \ldots, \vec{x}_m]$ represents the correlations among multiple sets. To keep the problem analogous as the two-set one, we use the sum of correlation criterion and rewrite (7.2) as

$$\max_{\vec{w}_i, 1\le u<v\le m} \rho = \frac{\sum_{u,v} \vec{w}_u^T C_{x_ux_v}\vec{w}_v}{\prod_{i=1}^m \sqrt{\vec{w}_i^T C_{x_ix_i}\vec{w}_i}}. \tag{7.3}$$

As discussed in Chapter 2, the solution is found in the generalized eigenvalue problem, given by

$$\begin{bmatrix} 0 & C_{x_1x_2} & \ldots & C_{x_1x_m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{x_mx_1} & C_{x_{m-1}x_1} & \ldots & 0 \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_m \end{bmatrix} = \rho \begin{bmatrix} C_{x_1x_1} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & C_{x_mx_m} \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_m \end{bmatrix}, \tag{7.4}$$

where ρ is the correlation coefficient.

7.2.2 Multiple Kernel CCA

Kernel Canonical Correlation Analysis (KCCA) is a nonlinear extension of CCA using kernel methods. The data is first mapped into a high dimensional Hilbert space induced by a kernel and then the linear CCA is applied. In this way, a linear correlation discovered in the Hilbert space corresponds to a nonlinear correlation concealed in the observation space. Let us denote $\{\vec{x}_1^k, \ldots, \vec{x}_m^k\}_{k=1}^N$ as N observations, respectively, obtained on m data sets $\vec{x}_1, \ldots, \vec{x}_m$ and $\phi_1(\cdot), \ldots, \phi_m(\cdot)$ as the feature maps from input spaces to the high dimensional Hilbert spaces. The centered kernel matrices of the m data sets becomes

$$\begin{aligned} G_1 &= [\phi_1(\vec{x}_1^{(1)})^T - \hat{\mu}_{\phi_1}; \ldots; \phi_1(\vec{x}_1^{(N)})^T - \hat{\mu}_{\phi_1}], \\ &\ldots \\ G_m &= [\phi_m(\vec{x}_m^{(1)})^T - \hat{\mu}_{\phi_m}; \ldots; \phi_m(\vec{x}_m^{(N)})^T - \hat{\mu}_{\phi_m}]. \end{aligned} \tag{7.5}$$

The projection vectors $\vec{w}_1, \ldots, \vec{w}_m$ lie in the span of the mapped data

$$\begin{aligned} \vec{p}_1 &= \Phi_1 \vec{w}_1, \\ &\ldots \\ \vec{p}_m &= \Phi_m \vec{w}_m. \end{aligned} \tag{7.6}$$

The resulting problem of KCCA can be deduced as the analogue of the linear CCA problem on the projected data sets $\vec{p}_1, \ldots, \vec{p}_m$ in Hilbert space:

$$\max_{\vec{w}_i, 1 \le u < v \le m} \rho = \frac{\sum_{u,v} \vec{w}_u^T C_{\Phi_u \Phi_v} \vec{w}_v}{\prod_{i=1}^m \sqrt{\vec{w}_i^T C_{\Phi_i \Phi_i} \vec{w}_i}}, \tag{7.7}$$

which leads to the generalized eigenvalue problem, given by

$$\begin{bmatrix} 0 & G_1 G_2 & \ldots & G_1 G_m \\ \vdots & \vdots & \ddots & \vdots \\ G_m G_1 & G_{m-1} G_1 & \ldots & 0 \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_m \end{bmatrix} = \rho \begin{bmatrix} G_1 G_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & G_m G_m \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_m \end{bmatrix}, \tag{7.8}$$

where G_i denotes the centered kernel matrix of the i-th data set where Mercer's condition is applied as:

$$G_{(ij)} = \phi(x_i)^T \phi(x_j) = G(x_i, x_j). \tag{7.9}$$

However, the problem in (7.8) is trivial and the non-zero solutions of generalized eigenvalue problem are $\rho = \pm 1$. To obtain meaningful estimations of canonical

correlation, it needs to be regularized [13, 71, 69]. The regularization, e.g., is done by the method proposed by Hardoon *et al.* [71] which results in the following regularized general eigenvalue problem:

$$\begin{bmatrix} 0 & \dots & G_1G_m \\ \vdots & \ddots & \vdots \\ G_mG_1 & \dots & 0 \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_m \end{bmatrix} = \rho \begin{bmatrix} (G_1+\kappa I)^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & (G_m+\kappa I)^2 \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_m \end{bmatrix} \tag{7.10}$$

where κ is a small positive regularization constant.

7.2.3 WMKCCA

Based on the objective function in (7.7), the weighted extension of MKCCA can be formulated by employing additional weights $\xi_{u,v}$ on the pairwise correlations as

$$\max_{\vec{w}_i, 1\le u<v\le m} \rho = \frac{\sum_{u,v} \xi_{u,v} \vec{w}_u^T C_{\Phi_u\Phi_v} \vec{w}_v}{\prod_{i=1}^{m} \sqrt{\vec{w}_i^T C_{\Phi_i\Phi_i} \vec{w}_i}} \tag{7.11}$$

where $\xi_{u,v}$ is the scalar weight of the correlation between $\vec{x}_u$ and $\vec{x}_v$. Let us denote the generalized eigenvalue problem in (7.10) as the form of $\Omega\vec{\alpha} = \lambda\Psi\vec{\alpha}$, the weights of KCCA is decomposed as an additional positive definite matrix $\mathcal{W}$ multiplying at the left and right side of the matrix Ω, given by

$$\mathcal{W}\Omega\mathcal{W}\vec{\alpha} = \lambda\Psi\vec{\alpha} \tag{7.12}$$

where

$$\mathcal{W} = \begin{bmatrix} \theta_1 I & 0 & \dots & 0 \\ 0 & \theta_2 I & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \theta_m I \end{bmatrix},$$

$$\Omega = \begin{bmatrix} 0 & G_1 G_2 & \dots & G_1 G_m \\ G_2 G_1 & 0 & \dots & G_2 G_m \\ \vdots & \vdots & \ddots & \vdots \\ G_m G_1 & G_{m-1} G_1 & \dots & 0 \end{bmatrix},$$

$$\Psi = \begin{bmatrix} (G_1 + \kappa I)^2 & 0 & \dots & 0 \\ 0 & (G_2 + \kappa I)^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (G_m + \kappa I)^2 \end{bmatrix},$$

$$\sum_{k=1}^{m} \theta_k = k, \quad \xi_{u,v} = \psi \theta_u \theta_v, \quad \psi = \frac{1}{\sum_{1 \le i < j \le m} \theta_i \theta_j}.$$

In the formulation above, the weights of pairwise correlation ξ in the objective function (7.11) are decomposed as the weights θ assigned on each data set. The sum of θ is constrained to keep the mean value as 1. ψ is a normalization parameter to make the sum of ξ equal to 1. This normalization constant ψ only affects the solution of eigenvalue but does not affect the eigenvector solution.

7.3 Computational Issue

7.3.1 Standard Eigenvalue Problem for WMKCCA

To solve the computational burden, we follow the transformation presented in [13] to rewrite the generalized eigenvalue problem in (7.12) as

$$[\mathcal{W}\Omega\mathcal{W} + \Psi]\vec{\alpha} = (\lambda + 1)\Psi\vec{\alpha}. \tag{7.13}$$

The problem of finding the maximal generalized eigenvalues in (7.12) is equivalent to finding the minimal generalized eigenvalues in (7.13) because if the generalized eigenvalues in (7.12) are given by

$$\lambda_1, -\lambda_1, \dots, \lambda_p, \lambda_p, 0, \dots, 0,$$

then the corresponding generalized eigenvalues in (7.13) are

$$1+\lambda_1, 1-\lambda_1, \ldots, 1+\lambda_p, 1-\lambda_p, 1, \ldots, 1.$$

Since Ψ is already regularized thus it can be decomposed as $\Psi = \mathcal{C}^T\mathcal{C}$, defining $\vec{\beta} = \mathcal{C}\vec{\alpha}$, and $K_\kappa = \mathcal{W}\Omega\mathcal{W} + \Psi$, the problem in (7.13) can be transformed as follows:

$$\mathcal{K}_\lambda\vec{\alpha} = \lambda^\sharp\Psi\vec{\alpha}$$

$$\mathcal{K}_\lambda\vec{\alpha} = \lambda^\sharp\mathcal{C}^T\mathcal{C}\vec{\alpha}$$

$$\mathcal{C}^{-T}\mathcal{K}_\kappa\mathcal{C}^{-1}\vec{\beta} = \lambda^\sharp\vec{\beta} \tag{7.14}$$

Notice that Ψ is a positive definite matrix in a block diagonal form, we have

$$\mathcal{C}^T = \mathcal{C} = \Psi^{1/2} = \begin{bmatrix} G_1+\kappa I & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & G_m+\kappa I \end{bmatrix}. \tag{7.15}$$

Replace (7.15) in (7.14), the problem is written in the form of a standard eigenvalue problem, given by

$$\begin{bmatrix} I & \ldots & \theta_1\theta_m I r_\kappa(G_1)r_\kappa(G_m) \\ \vdots & \ddots & \vdots \\ \theta_1\theta_m I r_\kappa(G_m)r_\kappa(G_1) & \ldots & I \end{bmatrix}\vec{\beta} = \lambda^\sharp\vec{\beta}, \tag{7.16}$$

where

$$r_\kappa(G_i) = G_i(G_i+\kappa I)^{-1}.$$

If eigenvalues $\lambda^\sharp$ and eigenvectors $\vec{\beta}$ are solved from (7.16), the eigenvalues and eigenvectors of problem (7.12) is $\lambda^\sharp$ and $\mathcal{C}^{-1}\vec{\beta}$. More formally, the eigenvectors $\vec{\alpha}_i$ of problem (7.10) are equivalent to

$$\vec{\alpha}_i = (G_i+\kappa I)^{-1}\vec{\beta}_i. \tag{7.17}$$

7.3.2 Incomplete Cholesky Decomposition

As introduced in [13], the full rank $N \times N$ centered kernel matrix G_i can be factorized as $G_i \approx \mathscr{C}\mathscr{C}^T$ by ICD, where $\mathscr{C}_i$ is in low rank $N \times M_i$ ($M_i \leq N$) matrix. Applying SVD on $\mathscr{C}_i$ one obtains $N \times M_i$ matrix U_i with orthogonal columns and $M_i \times M_i$ diagonal matrix Λ_i, given by

$$G_i \approx \mathscr{C}_i\mathscr{C}_i^T = U_i\Lambda_i V_i^T(U_i\Lambda_i V_i^T)^T = U_i\Lambda_i^2 U_i^T. \tag{7.18}$$

Denote E_i as the orthogonal complement of U_i, (U_iE_i) is then a full rank $N \times N$ matrix, G_i is given by

$$G_i \approx U_i \Lambda_i^2 U_i^T = [U_i E_i] \begin{bmatrix} \hat{\Lambda}_i & 0 \\ 0 & 0 \end{bmatrix} [U_i E_i]^T. \tag{7.19}$$

For regularized matrices in (7.10), one obtains:

$$r_\kappa(G_i) \approx [U_i E_i] \begin{bmatrix} R_i & 0 \\ 0 & 0 \end{bmatrix} [U_i E_i]^T = U_i R_i U_i^T, \tag{7.20}$$

where R_i is the diagonal matrix obtained from the diagonal matrix $\hat{\Lambda}_i$ by transformation $R_i^j = (\hat{\Lambda}_i^j)/(\hat{\Lambda}_i^j + \kappa)$ to its elements. Replacing (7.16) with (7.20), decomposing (7.16) as

$$U R_\kappa U^T \beta = \lambda^\sharp \beta, \tag{7.21}$$

where

$$U = \begin{bmatrix} U_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & U_m \end{bmatrix},$$

$$R_\kappa = \begin{bmatrix} I & \dots & \zeta_1\zeta_m I R_1 U_1^T U_m R_m \\ \vdots & \ddots & \vdots \\ \zeta_1\zeta_m I R_m U_m^T U_1 R_1 & \dots & I \end{bmatrix}. \tag{7.22}$$

Since R_k is deduced from a similar matrix transformation, the eigenvalues in (7.21) are equivalent to (7.16), moreover, the eigenvectors of the low rank approximation is related to the full rank problem by the following transformation:

$$\begin{aligned} U R_k U^T \vec{\beta} &= \lambda^\sharp \vec{\beta} \\ R_k U^T \vec{\beta} &= \lambda^\sharp U^T \vec{\beta} \\ R_k \vec{\gamma} &= \lambda^\sharp \vec{\gamma} \end{aligned} \tag{7.23}$$

Once we obtained the eigenvector $\vec{\gamma}_i$ in low rank approximation problem (7.23), we can recover the full rank solution in (7.16) by $\vec{\beta}_i = U_i \vec{\gamma}_i$. Furthermore, the generalized eigenvector $\vec{\alpha}_i$ of the original problem is computed as (7.17), so we have:

$$\vec{\alpha}_i \approx (G_i + \kappa I)^{-1} U_i \vec{\gamma}_i \tag{7.24}$$

We have several parameters involved in WMKCCA computation: κ as the regularization parameter, η as the precision parameter for ICD, τ as the cut

value of eigenvalues determining the size of U_i and λ_i in SVD of G_i. Among these parameters, η and τ are easy to select by numerical experiments because they monotonically control the precision of ICD and SVD. The regularization parameter κ is often determined empirically.

7.3.3 Incremental EVD solution for WMKCCA

Based on the weighted problem in (7.12), the update of weights in WMKCCA is equivalent to multiplying an update matrix $\mathcal{V}$ at the left and right sides of the WMKCCA formulation, given by

$$\mathcal{V}\mathcal{W}\Omega\mathcal{W}\mathcal{V}\vec{\alpha} = \lambda\Psi\vec{\alpha} \tag{7.25}$$

where

$$\mathcal{V} = \begin{bmatrix} v_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & v_m \end{bmatrix}, \tag{7.26}$$

and $v_1, \dots, v_m$ are the update ratios of weights corresponding to $\theta_1, \dots, \theta_m$. Following the analog steps from (7.13) to (7.16), the standard EVD problem with updated weights is in the form of:

$$\mathcal{A}\vec{\gamma} = \lambda^\sharp\vec{\gamma}, \tag{7.27}$$

where

$$\mathcal{A} = \begin{bmatrix} I & \dots & v_1 v_m \theta_1 \theta_m I R_1 U_1^T U_m R_m \\ \vdots & \ddots & \vdots \\ v_1 v_m \theta_1 \theta_m I R_m U_m^T U_1 R_1 & \dots & I \end{bmatrix}. \tag{7.28}$$

For simplicity let us denote the matrix in (7.28) before weight updating as $\mathcal{A}_{old}$, and the one after updating as $\mathcal{A}_{new}$. We denote $E = \mathcal{A}_{new} - \mathcal{A}_{old}$. For the weights updated problem, we are interested in solving $\mathcal{A}_{new}\vec{\gamma} = \lambda^\sharp\vec{\gamma}$ by reusing the solution obtained on $\mathcal{A}_{old}$. Thus, during weight update we could avoid redoing the EVD each time from scratch. Suppose we already have solution as

$$\vec{\gamma}_k\Lambda_k\vec{\gamma}_k^T = \mathcal{A}_{old}. \tag{7.29}$$

Then the updated problem is equivalent to adding E on both sides of equation, given by

$$\begin{aligned} \vec{\gamma}_k\Lambda_k\vec{\gamma}_k^T + E &= \mathcal{A}_{old} + E \\ \vec{\gamma}_k(\Lambda_k + \vec{\gamma}_k^T E\vec{\gamma}_k)\vec{\gamma}_k^T &= \mathcal{A}_{new} \\ \vec{\gamma}_k T\vec{\gamma}_k^T &= \mathcal{A}_{new} \end{aligned} \tag{7.30}$$

In (7.28) the weight updating only affects the off-diagonal elements of the matrix. Moreover, the constraints of weights matrix in (7.12) bounds the update weights $v_1, ..., v_m$ within a certain scope. Especially, for small scale updates, these values are close to 1. Thus the matrix E is given by

$$E = \begin{bmatrix} 0 & e_{1,2} & \dots & e_{1,m} \\ \vdots & \ddots & \vdots & \\ e_{m,1} & e_{m,2} & \dots & 0 \end{bmatrix}, \tag{7.31}$$

where

$$e_{i,j} = (v_i v_j - 1)\theta_i \theta_j I R_i U_i^T U_j R_j. \tag{7.32}$$

As shown, E only has non-zero values at off-diagonal positions and most of the elements are close to 0. Because $\vec{\gamma}_k$ is a unitary matrix, and $\vec{\gamma}_k^T E \vec{\gamma}_k$ is also a sparse matrix with most of the off diagonal elements are close to 0, therefore, the matrix T in (7.30) is a nearly diagonal matrix which can be solved efficiently by iterative EVD algorithms. Through this transformation, we can have a "warm start" solution of weights by storing the previous EVD solution and computing the EVD solution of the new problem incrementally.

7.4 Learning by WMKCCA

WMKCCA extracts common information among multiple heterogeneous data sets. Given a group of objects represented in different sources, the inter-relationships among these objects may contain some intrinsic patterns. Through WMKCCA, the correlations are calculated in the Hilbert space thus complex intrinsic patterns among multiple sources may be revealed. When multiple sources are combined, WMKCCA has the flexibility to leverage the importance of correlations among several data sources. The projections obtained in canonical spaces are also useful for machine learning applications. To illustrate this, we propose an learning framework based on WMKCCA. As illustrated in Figure 7.1, the proposed framework integrates WMKCCA with supervised learning where the validation data and test data are projected to the embedding of the training data by out-of-sample projection [17]. In this thesis, the model for WMKCCA is selected by cross-validating the machine learning performance on the validation set so that the parameters of kernel function and the weights assigned on correlations are optimized.

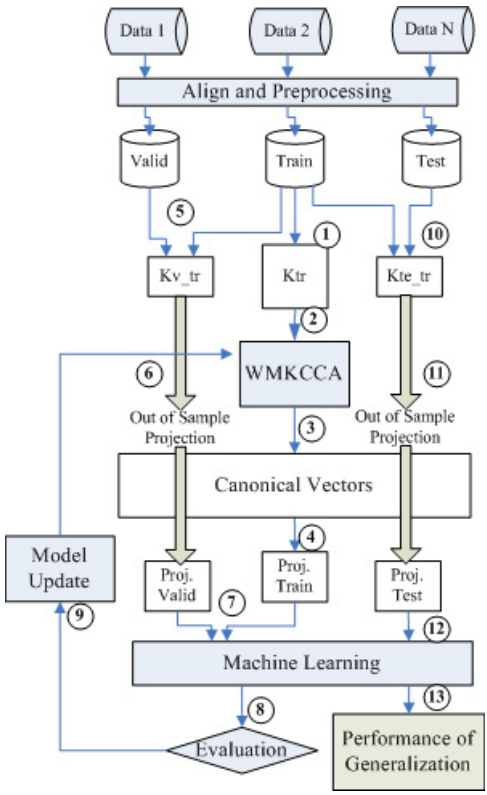


Figure 7.1: A framework of learning by WMKCCA on heterogeneous data sources.

7.5 Experiment

7.5.1 Classification in canonical spaces

We adopt the two UCI pattern recognition data sets, pen-based recognition of handwritten digits and optical recognition of handwritten digits, as mentioned in Chapter 3. As abbreviation, we denote them as PenData and OptData respectively. Both data sets have 10 digit labels from 0 to 9. PenData has 16 input attributes measured from 0 to 100 and OptData has 64 attributes measured from 0 to 16. We extract 3750 samples (375 samples for each digit) from the training part of both data sets, 80% of them are used for training and 20% are used for validation. We use their original test data as test set (3498 in PenData, 1797 in OptData). We apply RBF functions on both data sets and the kernel widths are selected as the mean of the sample covariance (for PenData $\sigma = 97$, for OptData $\sigma = 13$). We create a third data set by transforming the label information of

data as a kernel matrix. Firstly, the vector of class labels is coded as $N \times 10$ matrix L where the i-th column represents the label of the $i-1$-th digit. Then, the label matrix is transformed into a kernel matrix by linear function $L * L^T$. Therefore, in our training step we use three 3000×3000 kernel matrices, denoted as $K_{pen}, K_{opt}, K_{label}$ respectively.

We apply a centroid classification approach on the projected data in the canonical spaces which treats each set of digits as a cluster and calculates the centroid as the mean prototype in canonical space. When a new sample is presented for classification, the Euclidean distances from the new point to all the cluster centroids are calculated and the label is assigned to the new data as the cluster with the shortest distance. To obtain the validation performance and test performance, the validation data and test data are firstly projected to the canonical spaces of the training data with out-of-sample projection, then classified by the centroid method. The accuracy of the classification is evaluated by calculating the percentage of the correctly classified data in all digit labels.

We benchmark the performance in the validation sets with different weights, using the incremental EVD method presented in Section 7.3.3. The weights on $K_{pen}, K_{opt}, K_{label}$ are denoted as $\theta_{pen}, \theta_{opt}, \theta_{label}$ and they are optimized in a grid search from 0.1 to 2.9 with the step 0.1. We also benchmark the dimensionality of the canonical space by evaluating the performance of classification. For each validation, the data is projected to the subspace spanned by 10, 100 and 500 canonical vectors respectively. The weights and the size of the canonical space are selected by the average classification accuracies on the two validation data sets. Then the test data is projected to the WMKCCA model parameterized by the optimal weights and the optimal canonical space. The performance of the test set is evaluated by comparing the classification labels and the true labels. Furthermore, we compare the performance obtained by WMKCCA learning with the results reported by other methods in the literature. The results obtained by KCCA, MKCCA are also compared. As shown in Table 7.1, the performance obtained by the simple classification method applied in the canonical space is comparable to the best results reported in the literature.

Table 7.1: Classification accuracy obtained on the test data. By 10-fold cross-validation, the weights in WMKCCA are set as $\theta_{pen} = 1.3, \theta_{opt} = 1.3, \theta_{label} = 0.4$ and the dimensionality of the canonical space is set to 500. As shown, the results of WMKCCA, MKCCA and KCCA are comparable to the performance obtained by some complicated classification methods. However, the performance of WMKCCA, MKCCA and KCCA is not significantly different from each other. The performance of other algorithms is mean values reported in the literature. According to the comparison of the mean values, the proposed CCA approaches might perform better than the linear SVM. As mentioned, the kernel matrices in CCA approaches are all constructed by linear function, so the improvements should ascribe to the effects of canonical projections obtained from data fusion.

METHODS	PenData	OptData	Notes
WMKCCA	**0.9794**	**0.9716**	See text
MKCCA	**0.9766**	**0.9688**	Equal weights
KCCA	**0.9783**	**0.9711**	3750 training
Linear SVM	0.9494	0.9602	
RBF DDA SVM [138]	0.9708	0.9722	
SVM Ensemble [94]	N/A	0.9783	45 SVMs
MLP [138]	0.9703	0.9517	
kNN [138]	0.9771	0.9649	k=3
Bayes+PCA [75]	0.9763	0.9694	

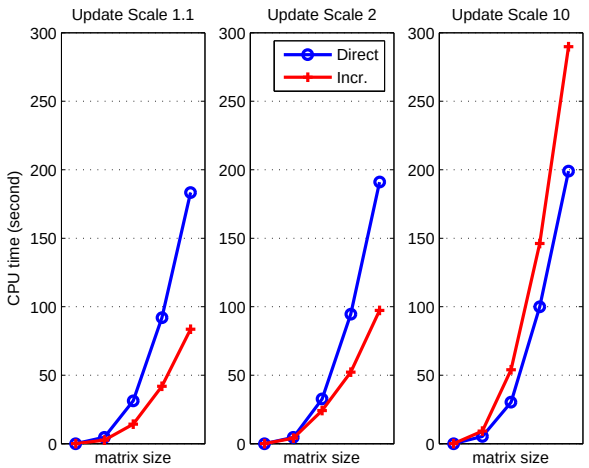


Figure 7.2: Comparison of CPU time used by direct EVD method and incremental EVD method in WMKCCA computation.

7.5.2 Efficiency of incremental EVD solution

We also compare the direct EVD approach and the incremental EVD algorithm in the WMKCCA experiments with different matrix sizes and update scales. The benchmark does not take into account the ICD and SVD processes but only compares the CPU time of solving the eigenvalue problem in (7.23) between the direct method and the incremental method. For the incremental method, the CPU time of calculating E, T, the EVD of T and the multiplication process of calculating the canonical vectors are all taken into account. We modify the scale of the problem by increasing the size of training set as 100, 500, 1000, 1500 and 2000 data samples. The linear kernels are constructed on two data sets and fed to the WMKCCA algorithm parameterized as ($\eta = 0.9$, $\kappa = 0.1$, $\tau = 0.9$). After ICD and SVD, the matrix $\mathcal{A}$ in (7.27) has the size of 160, 701, 1323, 1911 and 2483 respectively. We also adjust the scale of the weight update parameter v, which is denoted as the ratio between the new weight and the old weight in $\mathcal{V}$ as (7.26). As discussed, when v is close to 1, the off-diagonal elements in matrix E become sparse. When v is much larger than 1, E is not necessarily a sparse off-diagonal matrix, thus the assumption of incremental EVD does not hold. To demonstrate this, we compared three v values: 1.1, 2, and 10 which respectively represents a weak update, a moderate update and a strong update of weights in WMKCCA model. For the problem of 3 data sets, when the update scale is set to 2 and 10, the mean value of θ does not necessarily equal to 1 hence the constraint in (7.12) does not hold. The experiment is conducted on a desktop

PC with Intel Core 2 1.86GHz CPU and 2G memory. The software package for simulation is MATLAB 2006a. The EVD algorithm is implemented by the *eig* function in MATLAB, which is based on QR method. As shown in Figure 7.2, the incremental EVD algorithm significantly reduces the CPU time when the weight update is small and moderate. However, when the update is too large, the matrix T is not diagonal thus the effect of incremental algorithm is overwhelmed by the additional cost paid on matrix multiplication.

7.6 Summary

In this chapter we proposed a new weighted formulation of kernel CCA on multiple sets. Using low rank approximation and incremental EVD algorithm, the WMKCCA is applicable in machine learning problems as a flexible model to extract common information among multiple data sources. We tried some preliminary experiments to demonstrate the effect. The CCA based data fusion is in a different paradigm than the MKL method introduced in previous chapters. The multiple data sources are not merged, whereas the merit of data fusion relies in the subspace spanned by the canonical vectors. In machine learning, the projections of data in these canonical spaces may be useful and more significant to detect the common underlying patterns reside in multiple sources.

Chapter 8

Multi-view text mining for disease gene prioritization

So oft in theologic wars,
The disputants, I ween,
Rail on in utter ignorance
Of what each other mean,
And prate about an Elephant
Not one of them has seen!

– "The Blind Men and the Elephant",
John Godfrey Saxe (1816-1887) –

8.1 Introduction

Text mining helps biologists to collect disease-gene associations automatically from large volumes of biological literature. During the past ten years, there was a surge of interests in automatic exploration of the biomedical literature, ranging from the modest approach of annotating and extracting keywords from text to more ambitious attempts such as Natural Language Processing, text-mining based network construction and inference. In particular, these efforts effectively help biologists to identify the most likely disease candidates for further experimental validation. The most important resource for text mining applications now is the Medical Literature Analysis and Retrieval System Online (MEDLINE) database developed by the National Center for Biotechnology Information (NCBI) at the

National Library of Medicine (NLM). MEDLINE covers all aspects of biology, chemistry, and medicine, there is almost no limit to the types of information that may be recovered through careful and exhaustive mining [154]. Therefore, a successful text mining approach relies much on an appropriate model. To create a text mining model, the selection of CV and the representation schemes of terms occupy a central role and the efficiency of biomedical knowledge discovery varies greatly between different text mining models. To address these challenges, we propose a *multi-view text mining* approach to retrieve information from different biomedical domain levels and combine them to identify the disease relevant genes through prioritization. The *view* represents a text mining result retrieved by a specific CV, so the concept of *multi-view text mining* is featured as applying multiple controlled vocabularies to retrieve the gene-centric perspectives from free text publications. Since all the information is retrieved from the same MEDLINE database but only varied by the CV, the term *view* also indicates that the data consists of multiple domain-based perspectives of the same corpus. We expect that the correlated and complementary information contained in the *multi-view* textual data can facilitate the understanding about the roles of genes in genetic diseases.

8.2 Background: Computational gene prioritization

Genome-wide experimental methods to identify disease causing genes, such as linkage analysis and association studies, are often overwhelmed by large sets of candidate genes produced by the high throughput techniques [146]. In contrast, the low-throughput validation of candidate disease genes is time-consuming and expensive. Computational prioritization methods can rank candidate disease genes from these gene sets according their likeliness of being involved in a certain disease. Moreover, a systematic gene prioritization approach that integrates multiple genomic data sets provides a comprehensive *in silico* analysis on the basis of multiple knowledge sources. Several computational gene prioritization applications have been previously described. GeneSeeker [176] provides a web interface that filters candidate disease genes on the basis of cytogenetic location, phenotypes, and expression patterns. DGP (Disease Gene Prediction) [116] assigns probabilities to genes based on sequence properties that indicate their likelihood to the patterns of pathogenic mutations of certain monogenetic hereditary disease. PROSPECTR [1] also classifies disease genes by sequence information but uses a decision tree model. SUSPECTS [2] integrates the results of PROSPECTR with annotation data from Gene Ontology (GO), InterPro, and expression libraries to rank genes according to the likelihood that they are involved in a particular disorder. G2D (Candidate Genes to Inherited Diseases) [142] scores all concepts in GO according to their relevance to each disease via text mining. Then, candidate genes are scored through a BLASTX search on reference sequence. POCUS [175] exploits the tendency for genes to be involved in the same disease by identifiable

similarities, such as shared GO annotation, shared InterPro domains or a similar expression profile. eVOC annotation [170] is a text mining approach that performs candidate gene selection using the eVOC ontology as a controlled vocabulary. It first associates eVOC terms and disease names according to co-occurrence in MEDLINE abstracts, and then ranks the identified terms and selects the genes annotated with the top-ranking terms. In the work of Franke *et al.* [58], a functional human genetic network was developed that integrates information from KEGG, BIND, Reactome, human protein reference database, Gene Ontology, predicted-protein interaction, human yeast two-hybrid interactions, and microrray coexpressions. Gene prioritization is performed by assessing whether genes are close together within the connected gene network. Endeavour [3] takes a machine learning approach by building a model on a training set, then that model is used to rank the test set of candidate genes according to the similarity to the model. The similarity is computed as the correlation for vector space data and BLAST score for sequence data. Endeavour incorporates multiple genomic data sources (microarray, InterPro, BIND, sequence, GO annotation, Motif, Kegg, EST, and text mining) and builds a model on each source of individual prioritization results. Finally, these results are combined through order statistics to a final score that offers an insight on how related a candidate gene to the training genes on the basis of information from multiple knowledge sources. More recently, CAESAR [63] has been developed as a text mining based gene prioritization tool for complex traits. CAESAR ranks genes by comparing the standard correlation of term-frequency vectors (TF profiles) of annotated terms in different ontological descriptions and integrates multiple ranking results by arithmetical (min, max, and $average$) and parametric integrations.

8.3 Single view gene prioritization: a fragile model with respect to the uncertainty

In our previous work [196], we found the *configuration* of text mining model has a strong impact on the quality of prioritization model. The term *configuration* denotes the triplet choice of domain vocabulary, representation scheme, and ranking algorithm in text mining based gene prioritization. According to the result of full benchmark experiments shown in Table 8.1, the improperly selected *configuration* could lead to a large error (no-voc, single max, TFIDF, 0.3757) on prioritization, which is more than 7 times larger than the error of a carefully selected *configuration* (eVOC, 1-SVM, IDF, 0.0477). If the prioritization result is used as the reference list in biological validation, the difference of efficiency gained from a good *configuration* and a bad *configuration* will be remarkable. However, how to determine the good *configuration* for prioritization is a non-trivial issue.

Table 8.1: Errors of Leave-one-out validation of prioritization on single text mining profiles. The values in this table are obtained by leave-one-out validation of 618 disease relevant genes in 9,999 random candidate genes. The validations are repeated 10 times for all *configurations* and the deviations are all smaller than 0.0001. For each column, which means comparing the performance of algorithms using the same vocabulary and representation scheme, the smallest error is shown in **Bold**, the largest error is shown in *Italic*. The introduction of experimental setup is available in [196]

		GO		EVOC		MESH		LDDB		OMIM		ALLVOC	
No.	Algorithm	*idf*	*tfidf*	*idf*	*tfidf*	*idf*	*tfidf*	*idf*	*tfidf*	*idf*	*tfidf*	*idf*	*tfidf*
1	1-SVM	**0.0758**	**0.0916**	**0.0477**	**0.1316**	**0.0497**	0.1107	**0.0877**	0.1713	**0.0714**	0.1288	0.0936	0.1745
2	Correlation	0.0781	0.1168	0.0499	0.137	0.0553	**0.1103**	0.089	0.1928	0.0798	0.1364	**0.0905**	0.1736
3	1-NN	0.1203	0.1201	0.1004	0.1676	0.1274	0.1150	0.1332	*0.2832*	0.1034	0.1296	0.1249	0.1816
4	2-NN γ	0.1038	0.1138	0.082	0.1513	0.1073	0.1075	0.1173	0.2570	0.0931	**0.1243**	0.1155	0.1760
5	2-NN δ	0.1297	0.1203	0.0994	0.1596	0.1421	0.1122	0.1359	0.2572	0.1156	0.1278	0.1288	0.1767
6	2-NN κ	0.0971	0.1152	0.0733	0.1471	0.0960	0.1088	0.1111	0.2511	0.0933	0.1333	0.1167	0.1792
7	3-NN γ	0.0973	0.1113	0.0744	0.1473	0.0974	0.1047	0.1104	0.2498	0.1103	0.2360	0.1110	0.1731
8	3-NN δ	0.1313	0.1162	0.0961	0.1495	0.1394	0.1122	0.1365	0.2458	0.1174	0.1254	0.1146	0.1824
9	3-NN κ	0.0970	0.1162	0.0670	0.1460	0.0918	0.1090	0.1044	0.2446	0.0959	0.1341	0.1157	0.1801
10	2-mean min	0.0888	0.1180	0.0641	0.1529	0.0833	0.1094	0.0982	0.2303	0.0899	0.1332	0.1088	0.1776
11	2-mean avg	0.0822	0.1241	0.0564	0.1611	0.0638	0.1148	0.0885	0.1956	0.0828	0.1483	0.0956	0.1820
12	2-mean max	0.1098	0.1695	0.0812	0.1968	0.0906	0.1619	0.1135	0.1991	0.1134	0.1977	0.1266	0.2387
13	3-mean min	0.0974	0.1202	0.0778	0.1559	0.0987	0.1132	0.1059	0.2572	0.0955	0.1311	0.1184	0.1827
14	3-mean avg	0.0830	0.1243	0.0558	0.1508	0.0645	0.1156	0.0899	0.1975	0.0850	0.1433	0.0983	0.1833
15	3-mean max	0.1293	0.2019	0.0882	0.2237	0.1065	0.2017	0.1208	0.1849	0.1323	0.2043	0.1556	0.2836
16	SL min	0.0858	0.1184	0.0601	0.1560	0.0694	0.1110	0.0922	0.2223	0.0900	0.1381	0.0913	**0.1749**
17	SL avg	0.1073	0.1659	0.0717	0.2069	0.0866	0.1625	0.0923	0.2031	0.1018	0.1742	0.1079	0.2281
18	SL max	*0.2888*	*0.3095*	*0.2133*	*0.3212*	*0.2529*	*0.3027*	*0.1823*	**0.1657**	*0.2276*	*0.2756*	0.2789	*0.3757*
19	CL min	0.0957	0.1199	0.082	0.156	0.0888	0.1091	0.098	0.232	0.0956	0.1337	0.1028	0.1766
20	CL avg	0.1018	0.1443	0.0776	0.1788	0.0871	0.1413	0.0946	0.2000	0.0994	0.155	0.1062	0.1995
21	CL max	0.2219	0.2262	0.1481	0.2549	0.1766	0.2405	0.1665	0.1906	0.1774	0.2183	0.2239	0.2992
22	AL min	0.0884	0.119	0.0679	0.1612	0.0749	0.1075	0.0895	0.2189	0.0878	0.1381	0.0961	0.1707
23	AL avg	0.1097	0.1666	0.0783	0.2068	0.088	0.1594	0.0886	0.195	0.1035	0.1747	0.1107	0.2267
24	AL max	0.2827	0.3039	0.2119	0.3173	0.2341	0.2896	0.185	0.1557	0.2464	0.2738	*0.294*	0.3736
25	WL min	0.0839	0.1176	0.0614	0.151	0.0765	0.1129	0.0965	0.2334	0.0889	0.1336	0.1088	0.1774
26	WL avg	0.0795	0.1285	0.0567	0.1571	0.0577	0.1241	0.0896	0.2013	0.0817	0.1483	0.0927	0.1878
27	WL max	0.1017	0.1823	0.0934	0.2149	0.0813	0.1754	0.1214	0.2074	0.1164	0.2033	0.1239	0.2564

8.4 Data fusion for gene prioritization: distribution free method

In Endeavour [3] [172], the combination of multi-source prioritization is based on a generic model named order statistics. A Q statistics is calculated from all rank ratios using the joint cumulative distribution of an N-dimensional order statistic as previously done by Stuart *et al.* [163]

$$Q(r_1, r_2, ..., r_N) = N! \int_0^{r_1} \int_{s_1}^{r_2} ... \int_{r_{N-1}}^{r_N} ds_N ds_{N-1} ... ds_1 \tag{8.1}$$

To reduce the complexity, Aerts *et al.* implement a much faster alternative formula, given by

$$V_k = \sum_{i=1}^{k} (-1)^{i-1} \frac{V_{k-i}}{i!} r_{N-k+1}^i \tag{8.2}$$

with $Q(r_1, r_2, ..., r_N) = N!V_N$, $V_0 = 1$, and r_i is the rank ratio for data source i. The Q statistics for randomly and uniformly drawn rank ratios is found approximately distributed as a beta distribution when $N \leq 5$, and a gamma distribution for $N > 5$. According to the cumulative distribution, we obtain P-value for every Q statistic from the order statistics [3]. In this way, the original N rankings are combined to a ranking of p-values, as shown in Figure 8.1. The main problem for the generic model based integration is that the performance highly relies on the distribution estimation of the gene prioritization scores. In text mining, the genes are often expressed as sparse vectors spanned in very high dimensional space, therefore, to estimate the distribution of the scores ("distances") among these genes is not always reliable, e.g., "the Pearson variation of the corresponding distance distribution degrades to 0 with increasing dimensionality"[18]. To tackle this problem, one may refine the estimation by designing new distance functions taking the high dimensionality affect into account, such as the work presented by Hsu and Chen [79]. Alternatively, one may adopt the algorithms based on statistical learning theory [178] where the error bound is distribution-free. De Bie *et al.* [46] propose a kernel novelty detection method on the basis of statistical learning theory and prove the error bound of false negative genes in prioritization. Given that bound, the number of false positives is controlled by the total number of positives [46]. To control the number of the false positive, the kernel matrix is centered for the whole genome. In practical, it is centered for the union set of the training genes and the test genes. The algorithm of kernel novelty detection is similar to the 1-SVM MKL introduced in (3.5) but

involves more statistical learning considerations, given by [46]

$$\begin{aligned} \min_{\vec{\alpha}} \quad & t \\ \text{s.t.} \quad & t \geq \vec{\alpha}^T \frac{K_j}{\upsilon_j} \vec{\alpha}, \ j = 1, ..., p \\ & 0 \leq \alpha_i \leq \frac{1}{\nu N}, \ i = 1, ..., N \\ & \sum_{i=1}^{N} \alpha_i = 1, \end{aligned} \tag{8.3}$$

and the prioritization function is given by

$$f(x) = \frac{1}{\sqrt{\vec{\alpha}^T \Omega \vec{\alpha}}} \sum_{i=1}^{N} \alpha_i \left[\sum_{j=1}^{p} \theta_j \frac{K_j(\vec{x}, \vec{x}_i)}{\upsilon_j} \right], \tag{8.4}$$

where

$$\Omega = \left\{ \sum_{j=1}^{p} \theta_j \frac{K_j}{\upsilon_j} \middle| \sum_{j=1}^{p} \theta_j = 1, \quad \forall j, \ \theta_j \geq 0 \right\}. \tag{8.5}$$

In (8.3), (8.4), and (8.5), N is the number of training genes, p is the number of kernels, ν is the regularization parameter, υ_j is a positive constant controlling the complexity measured from each kernel K_j [106]. Empirically, υ_j is often equated as the trace of the $j-$th centred genome-wide kernel matrix divided by the total number of genes in the genome (practically the sum of training plus test genes), thus, the trace of the combined kernel matrix Ω of the whole genome is equivalent to the number of genes in the whole genome and the norm $K(\vec{x}, \vec{x})$ of a gene is equal to 1 on average [46]. The conceptual framework of the 1-SVM MKL approach for gene prioritization is illustrated in Figure 8.2. This approach is empirically compared with the generic model adopted in Endeavour. When using the same genomic data sources, the distribution-free kernel method outperforms the generic model.

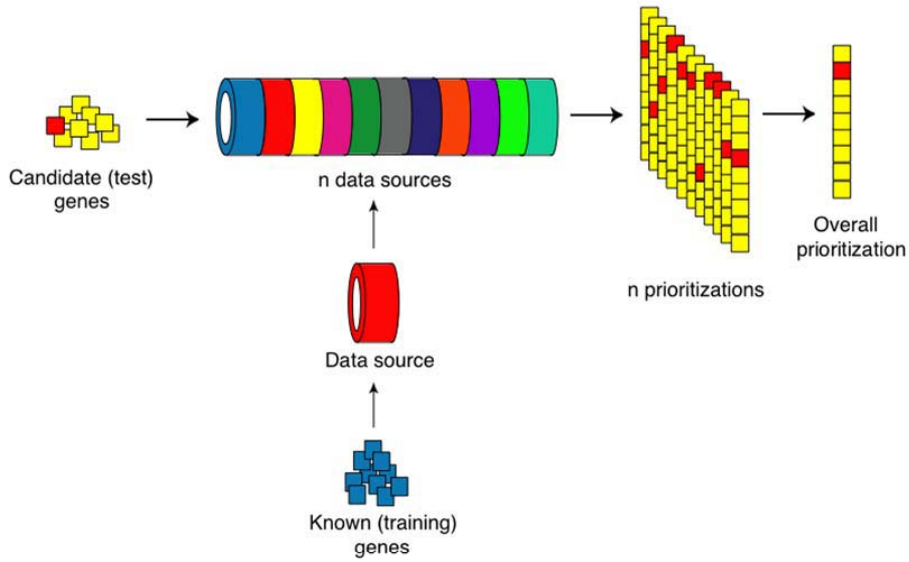


Figure 8.1: The conceptual framework of gene prioritization by data fusion [3] [172]. The unified prioritization is obtained by the generic integration of individual prioritization results based on order statistics.

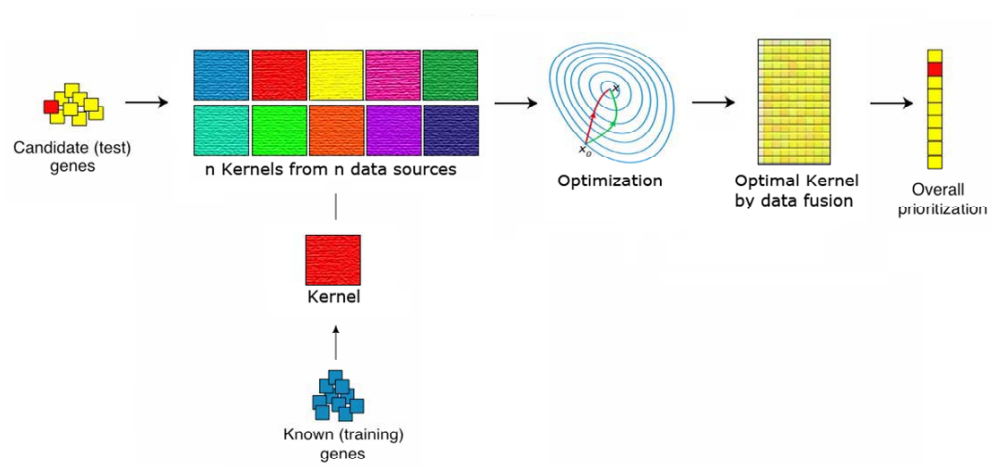


Figure 8.2: The conceptual framework of kernel gene prioritization by data fusion proposed in [46]. The unified prioritization is obtained by the scoring function based on the optimal combination of kernel matrices in (8.4)

8.5 Multi-view text mining gene prioritization

8.5.1 Construction of controlled vocabularies from multiple bio-ontologies

We select vocabularies from nine bio-ontologies for text mining, among which five of them (GO, MeSH, eVOC, OMIM and LDDB) have proven their merit in our earlier work of text based gene prioritization [196] and text based cytogenetic bands mapping [177]. Besides these five, we select four additional ontologies (KO, MPO, SNOMED CT, and UniprotKB) because they are also frequently adopted in the identification of genetic diseases and signaling pathways, for instance, in the works of Gaulton *et al.* [63], Bodenreider [23], Mao *et al.* [118], Smith *et al.* [159], and Melton *et al.* [121]. The nine bio-ontlogies are briefly introduced as follows.

The Gene Ontology
GO [38] provides consistent descriptions of gene and gene-product attributes in the form of three structured controlled vocabularies that each provide a specific angle of view (biological processes, cellular components and molecular functions). GO is built and maintained with the explicit goal of applications in text mining and semantic matching in mind [177]. Hence, it is an ideal source as domain-specific views in our approach. We extract all the terms in GO Slims (due to the version released in December, 2008) as the CV of GO.

Medical Subject Headings
MeSH is a controlled vocabulary produced by NLM for indexing, cataloging, and searching biomedical and health-related information and documents. The descriptors or subject headings of MeSH are arranged in a hierarchy. MeSH covers a broad ranges of topics and its current version consists of 16 top level categories. Though most of the articles in MEDLINE are already annotated with MeSH terms, our text mining process does not rely on these annotations but indexes the MEDLINE repository automatically with the MeSH descriptors (version 2008).

Online Mendelian Inheritance in Man's Morbid Map
OMIM [120] is a database that catalogues all the known diseases with genetic components. It contains available links between diseases and relevant genes in the human genome and provides references for further research and tools for genomic analysis of a catalogued gene. OMIM is composed of two mappings: the OMIM Gene Map, which presents the cytogenetic locations of genes that are described in OMIM; the OMIM Morbid Map, which is an alphabetical list of diseases described in OMIM and their corresponding cytogenetic locations. Our approach retrieves the disease descriptions from the OMIM Morbid Map (version due to December, 2008) as the CV.

London Dysmorphology Database
LDDB is a database containing information over 3000 dysmorphic and neurogenetic syndromes, which is initially developed to help experienced dysmorphologists to arrive at the correct diagnosis in difficult cases with multiple congenital anomalies [187]. Information in the database is constantly updated and over 1000 journals are regularly reviewed to ascertain appropriate reports. The London Neurology Database (LNDB) is a database of genetic neurological disorders based on the same data structure and software as the LDDB [11]. We extract the dysmorphology taxonomies from LNDB (version 1.0.11) and select the vocabulary terms.

eVOC
eVOC [92] is a set of vocabularies that unifies gene expression data by facilitating a link between the genome sequence and expression phenotype information. It was originally categorized as four orthogonal controlled vocabularies (anatomical system, cell type, pathology, and developmental stage) and now extended into 14 orthogonal subsets subsuming the domain of human gene expression data. Our approach selects the vocabulary from the eVOC version 2.9.

KEGG Orthology
KO is a part of the KEGG suite [90] of resources. KEGG is known as a large pathway database and KO is developed to integrate pathway and genomic information in KEGG. KO is structured as a directed acyclic graph (DAG) hierarchy of four flat levels [118]. The top level consists of the following five categories: metabolism, genetic information processing, environmental information processing, cellular processes and human diseases. The second level divides the five functional categories into finer sub-categories. The third level corresponds directly to the KEGG pathways, and the fourth level consists of the leaf nodes, which are the functional terms. In literature, KO has been used as an alternative controlled vocabulary of GO for automated annotation and pathway identification [118]. The KO based controlled vocabulary in our approach is selected on the version due to December 2008.

Mammalian Phenotype Ontology
MPO [159] contains annotations of mammalian phenotypes in the context of mutations, quantitative trait loci and strains which was initially used in Mouse Genome Database and Rat Genome Database to represent phenotypic data. Because mouse is the premier model organism for the study of human biology and disease, in the CAESAR [63] system, MPO has also been used as a controlled vocabulary for text mining based gene prioritization of human diseases. The MPO

based controlled vocabulary in our approach is selected on the version due to December 2008.

Systematized Nomenclature of Medicine–Clinical Terms
SNOMED is a huge and comprehensive clinical terminology, originally created by the College of American Pathologists and, now owned, maintained, and distributed by the International Health Terminology Standards Development Organization (IHTSDO). SNOMED is a very "fine-grained" collection of descriptions about care and treatment of patients, covering areas like diseases, operations, treatments, drugs, and healthcare administration. SNOMED has been investigated as an ontological resource for biomedical text mining [23] and also has been used in patient-based similarity metric construction [121]. We select the CV on the SNOMED (version due to December, 2008) obtained from the Unified Medical Language System (UMLS) of NLM.

Universal Protein Knowledgebase
UniProtKB [56] is a repository for the collection of functional information on proteins with annotations developed by European Bioinformatics Institute (EBI). Annotations in UniProtKB are manually created and combined with non-redundant protein sequence database, which brings together experimental results, computed features and scientific conclusions. Mottaz *et al.* [125] design a mapping procedure to link the UniProt human protein entries and corresponding OMIM entries to the MeSH disease terminology. The vocabulary applied in our approach is selected on UniProt release 14.5 (due to December, 2008).

The terms extracted from these bio-ontologies are stored as bag-of-words and preprocessed for text mining. The preprocessing includes transformation to lower case, segmentation of long phrases, and stemming. After preprocessing, these vocabularies are fed into a Java program based on Apache Java Lucene API to index the titles and abstracts of MEDLINE publications relevant to human genes.

8.5.2 Vocabularies selected from subsets of ontologies

As mentioned, in some "fine-grained" bio-ontologies the concepts and terminologies are labeled in multiple hierarchies, denoted as sub-ontologies, to represent domain concepts at various levels of specificity. For instance, GO consists of three sub-ontologies: biological process, cellular component and molecular function. MeSH descriptors are arranged in 16 hierarchical trees at the top level. In SNOMED, the medical terminologies are composed of 19 higher level hierarchies. eVOC ontology contains 14 orthogonal vocabulary subsets, in which the terms contained are strictly non-overlapping. To investigate whether more specific

vocabularies can improve the effectiveness of the text mining model, we select terms from the sub-ontologies of GO, MeSH, SNOMED, and eVOC and compose the corresponding subset CVs. Considering the main objective as disease-associated gene identification, only the most relevant sub-ontologies (6 from eVOC, 7 from MeSH and 14 from SNOMED) are selected. To distinguish the gene-by-term profiles obtained from subset CVs with those obtained from complete CVs, we denote the former one as subset CV profile and latter one as complete CV profile.

8.5.3 Merging and mapping of controlled vocabularies

The strategy of incorporating multiple CVs may be alternatively achieved by merging terms of several vocabularies together. To investigate this, we merge the terms of all 9 complete CVs as a union of vocabulary and denote the corresponding gene-by-term text mining result as "merge-9 profile". Furthermore, we notice the lexical variants across multiple ontologies: a concept may be represented as different terms due to the diversities of professional expressions. For example, the MeSH term "denticles" is expressed as "dental pulp stones" in OMIM and as "pulp stone" in SNOMED. To resolve these variants, we refer the UMLS Metathesaurus to map terminological variants as unified concepts. We download the concept names and sources file (MRCONSO.RRF) from UMLS Metathesaurus, which provides the mapping of atoms (each occurrence of unique string or concept name within each source vocabulary) to unified concepts. In text mining, we build a synonym vocabulary to map and aggregate the occurrences of various synonym terms are mapped and aggregated as the indices of the unified concept. The obtained results are gene-by-concept profile which the features are the unique and permanent concept identifiers defined by UMLS Metathesaurus. Among the nine vocabularies adopted in our approach, only four of them (GO, MeSH, OMIM and SNOMED) are included in UMLS and resolved in the MRCONSO.RRF file. Therefore, to fairly compare the effect of concept mapping, we also create "merge-4" gene-by-term profile using the union of the four vocabularies in indexing. Then, we map the lexical variants as concepts and the result is denoted as "concept-4" profile. Moreover, to create a naive baseline, we also index the MEDLINE corpus without using any controlled vocabulary. All the terms appeared in the corpus are segmented as single words [1] and the results are expressed by these vast amount of words, denoted as "no-voc profile".

[1] We didn't consider the phrases of multiple words because there would be immense amount of combinations.

Table 8.2: Overview of the controlled vocabularies applied in the multi-view approach. The *Number of indexed terms* of controlled vocabularies reported in this table are counted on indexing results of human related publications only so their numbers are smaller than those in our earlier work [196], which were counted on all species appeared in GeneRIF. The *Number of terms in CV* are counted on the vocabularies independent to the indexing process.

No.	CV	Nr. terms CV	Nr. indexed terms
1	**eVOC**	1659	1286
2	eVOC anatomical system	518	401
3	eVOC cell type	191	82
4	eVOC human development	658	469
5	eVOC mouse development	369	298
6	eVOC pathology	199	166
7	eVOC treatment	62	46
8	**GO**	37069	7403
9	GO biological process	20470	4400
10	GO cellular component	3724	1571
11	GO molecular function	15282	3323
12	**KO**	1514	554
13	**LDDB**	935	890
14	**MeSH**	29709	15569
15	MeSH analytical	3967	2404
16	MeSH anatomy	2467	1884
17	MeSH biological	2781	2079
18	MeSH chemical	11824	6401
19	MeSH disease	6717	4001
20	MeSH organisms	4586	1575
21	MeSH psychiatry	1463	907
22	**MPO**	9232	3446
23	**OMIM**	5021	3402
24	**SNOMED**	311839	27381
25	SNOMED assessment scale	1881	810
26	SNOMED body structure	30156	2865
27	SNOMED cell	1224	346
28	SNOMED cell structure	890	498
29	SNOMED disorder	97956	13059
30	SNOMED finding	51159	3967
31	SNOMED morphologic abnormality	6903	2806
32	SNOMED observable entity	11927	3119
33	SNOMED procedure	69976	9575
34	SNOMED product	23054	1542
35	SNOMED regime therapy	5362	1814
36	SNOMED situation	9303	2833
37	SNOMED specimen	1948	742
38	SNOMED substance	33065	8948
39	**Uniprot**	1618	520
40	Merge-9	372527[2]	50687
41	Merge-4	363321	48326
42	Concept-4	1420118	44714
43	No-voc	-	259815

8.5.4 Text mining

We refer the mapping of genes and publications in Entrez GeneRIF and index a subset of MEDLINE literature repository (as of 10 December, 2008) that consists of 290,000 human gene-related publications. In the first step the MEDLINE documents are indexed and the doc-by-term (or doc-by-concept) vectors are constructed. In the second step, we averagely combine the document-by-term (document-by-concept) vectors as gene-by-term (gene-by-concept) vectors according to the GeneRIF mapping. The detail of the text mining process is presented in our earlier work [66, 196]. Table 8.2 lists all the CVs applied in our approach. Table 8.3 illustrates the overlapping terms among the nine complete CVs.

Table 8.3: The number of overlapping terms in different vocabularies and indexed terms. The upper triangle matrix shows the numbers of overlapping terms among vocabularies independent to indexing. The lower triangle matrix shows the numbers of overlapping indexed terms. The second horizontal row (from the top) are the numbers of the terms in vocabularies independent to indexing. The second vertical column (from the left) are the numbers of the indexed terms.

		eVOC	GO	KO	LDDB	MeSH	MPO	OMIM	SNOMED	Uniprot
		1659	37069	1514	935	29709	9232	5021	311839	1618
eVOC	1286	-	370	16	118	827	566	325	876	46
GO	7403	358	-	404	74	3380	1234	659	4772	325
KO	554	16	344	-	1	383	72	120	489	44
LDDB	890	118	74	1	-	346	275	205	498	16
MeSH	15569	784	2875	344	343	-	2118	1683	12483	373
MPO	3446	554	1177	72	271	2007	-	823	2729	146
OMIM	3402	322	655	119	205	1644	816	-	2275	161
SNOMED	27381	814	3144	380	492	8900	2508	2170	-	593
Uniprot	520	46	301	42	16	361	146	157	371	-

Preliminary result shows that the weighting scheme of terms also influences the performance of gene-by-term data in biological validation [196]. When the same vocabulary and the ranking algorithm are applied in prioritization, the IDF representation generally outperforms the TF-IDF and the binary representations. Therefore, in this article all the term profiles are represented in the IDF weighting scheme.

8.5.5 Dimensionality reduction of gene-by-term data by Latent Semantic Indexing

We have introduced the subset CVs method to reduce the number of terms expressing the genes. Alternatively, we also apply LSI to reduce the number of term features. On the one hand, the information expressed on vast numbers of terms is mapped to a smaller number of latent factors so the irrelevant information is reduced. On the other hand, we expect that LSI does not compromise the information required for prioritization and clustering. In implementation, we use the Matlab function *svds* to solve the eigenvalue problem of the sparse gene-by-term matrix of the whole human genome (22,743 human genes). To calculate the total variance on this huge matrix is very computational expensive, so we sort the eigenvalues obtained by the sparse eigenvalue decomposition. To determine the number of latent factors, we select the dimension where the corresponding smallest eigenvalue is less than 0.05% of the sum of all eigenvalues.

8.5.6 Algorithms and evaluation of gene prioritization task

The computational definition of gene prioritization is mentioned in our earlier work [3, 46, 172, 196]. We briefly introduce here for the completeness. Genes that are known relevant to the same disease are constructed as a disease-specific training set. A prioritization model is first built on this training set, then that model is used to rank a test set of candidate genes according to their similarity to the model. The performance is evaluated by checking the positions of the real relevant genes in the ranking of a test set. A perfect prioritization should rank the gene with the highest relevance to the biomedical concept, represented by the training set, at the highest position (at the top). The interval between the real position of that gene with the top is similar to the error. For a prioritization model, minimizing this error is equal to improving the ranking position of the most relevant gene and in turn it reduces the number of irrelevant genes to be investigated in biological experimental validation. So a model with smaller error is more efficient and accurate to find disease relevant genes and that error is also used as a performance indicator for model comparison [196]. The ranking of candidate genes is usually based on scores. Assuming larger score represents higher similarity towards the prioritization model, in benchmark study, one can label the real relevant genes as class "+1" and other irrelevant genes as class "-1" and plot the Receiver operating characteristic (ROC) curves to compare different models by the values of area under curve (AUC). The error of prioritization is thus equivalent to 1 minus the AUC value.

The methods to combine models for prioritization are roughly classified as two approaches: ensemble of rankings and fusion of sources (kernels).

Ensemble ranking

In ensemble ranking, the prioritization is first carried on each individual model and then multiple ranking results are combined. Since our main objective is to integrate the models, we use the same algorithm, standard correlation, as the base method to obtain ranking results on individual models. Using other algorithms, the results after model integration may not be significantly different and the computational complexity is more likely to be higher than the standard correlation algorithm.

The ranking results is integrated either as ranking orders (ratios) or as ranking scores. To compare them, we implement three integration algorithms. Two of them are basic operators to calculate the average or maximal value of multiple ranking scores. The third one is based on order statistics as defined in (8.1).

Kernel fusion for prioritization

The kernel fusion method for gene prioritization is proposed by De Bie *et al.* [46] as the 1-SVM problem defined in (8.3). We apply this 1-SVM method to combine kernels derived from the multi-view data for gene prioritization. Because the dimensionality of gene-by-term profile is high, we only use linear function to construct the kernel matrices. One of the main features of the 1-SVM is the sparsity of its solutions, which the dominant coefficient may be assigned on one or two kernel matrices. This property is useful to distinguish a small amount of relevant sources from large amount of irrelevant sources in data fusion. However, in biomedical application, the data sources are usually preprocessed and have high relevances w.r.t. the problem. Sparse solution may be too selective, in this case, to thoroughly combine the redundant and complementary information in these data sources. To balance the effect of sparse coefficients (most of θ_j are equal to 0) and non-sparse coefficients in model generalization, we try 3 different values of the regularization parameter θ_{min} to restrict the optimization process as a lowerbound of coefficient assigned on each kernel. When $\theta_{min} = 0$, there is no lowerbound and the optimization procedure will probably result in the sparse coefficients. When $\theta_{min} = 0.5/N$, each kernel is insured to have a minimum contribution in data fusion. When $\theta_{min} = 1/N$, the kernel matrices are averagely combined. In our implementation, the QCLP problem in (8.3) is solved by SeDuMi 1.2 [164].

Evaluation of prioritization

The prioritization result is evaluated by leave-one-out (LOO) method [196]. In each experiment, given a disease gene set which contains K genes, one gene, termed the "defector" gene, is deleted from the set of training genes and added to 99 randomly selected test genes (test set). We use the remained $K - 1$ genes (training set) to

build our prioritization model. Then, we prioritize the test set which contains 100 genes by the trained model and determine the ranking of that defector gene in test data. The prioritization performance is thus equivalent to the error (1 minus the AUC value).

8.5.7 Benchmark data set of disease genes

We validate the clustering results with the human disease benchmark data set of Endeavour [3], which consists of 620 relevant genes from 29 diseases. Genes from the same disease are constructed as a disease-specific training set used to evaluate the prioritization and clustering performance. For prioritization, we perform 620 rankings (with each gene left out once) on 99 candidate human genes randomly selected from the human genomic. The prioritization is repeated 20 times (with randomly permuted 99 random genes each time for each left out gene) and the average Error value is reported as the final result.

8.6 Results

8.6.1 Multi-view performs better than single view

According to the experimental result, integration of the *multi-view* data obtains significantly better performance than the best individual data. Among the different approaches we tried, the best performance is obtained by combining 9 CV profiles as kernels in the LSI reduced dimensionality (1-SVM+LSI, Error of AUC = 0.0335). This error is only a half of the best single CV profile (LDDB, 0.0792). The ROC curves of leave-one-out performance of different approaches are presented in Figure 8.3. Without LSI, 1-SVM data fusion reduces the error from 0.0792 to 0.0453. By coupling LSI with 1-SVM, the error is further reduced from 0.0453 to 0.0335. Considering the cost and effort of validating the false positive genes in lab experiments, the improvement from 0.0792 to 0.0335 is quite meaningful because it means that when prioritizing 100 candidate genes, our proposed method can save the effort of validating 4 false positive genes. The obtained result is also comparable to the performance of the existing systems. In the Endeavour system [3], the same disease gene benchmark dataset and the same evaluation method is implemented. Endeavour is different from our approach mainly in two aspects. Firstly, Endeavour combines one textual data (GO-IDF profile obtained from free literature text mining) with nine other biological data sources. Also, there is no dimensionality reduction used in it. Secondly, Endeavour applies order statics to integrate the models. The performance obtained in our paper is much better than Endeavour (Error=0.0833). Moreover, the performance is also better than the result obtained by De Bie *et al.* [46]. In their approach, they use the

same data sources as Endeavour and apply the 1-SVM MKL algorithm for model integration (best performance Error=0.0477, $\theta_{min} = 0.5/k$). Since the methods and the disease gene benchmark data are exactly the same, the improvement can only be attributed to the multi-view text mining and the LSI dimensionality reduction.

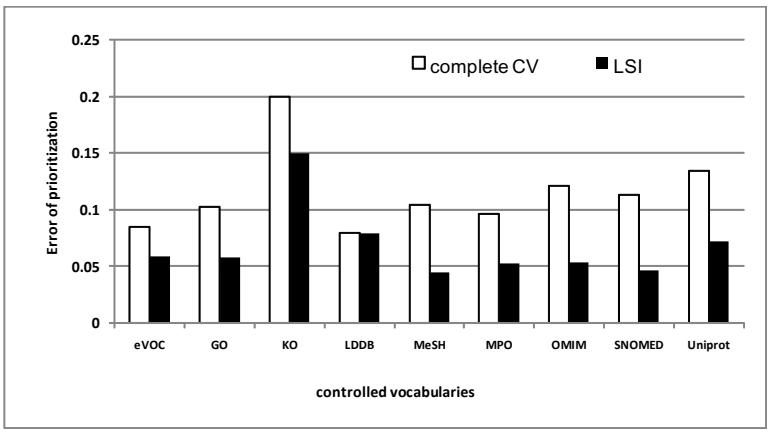


Figure 8.3: Prioritization results obtained by complete CV profiles and LSI profiles.

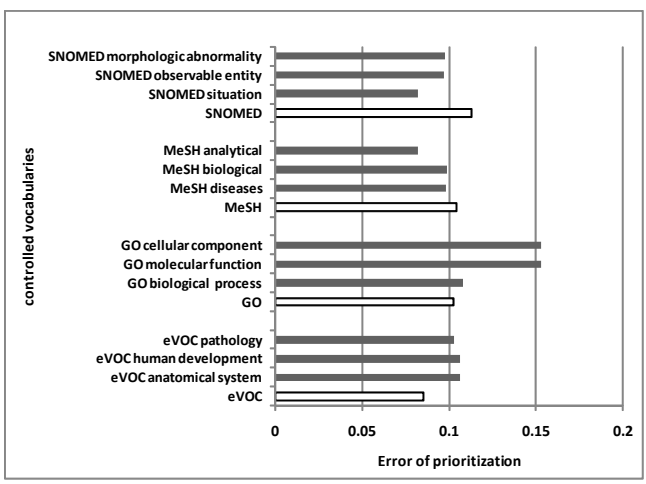


Figure 8.4: Prioritization results obtained by complete CV profiles and subset CV profiles.

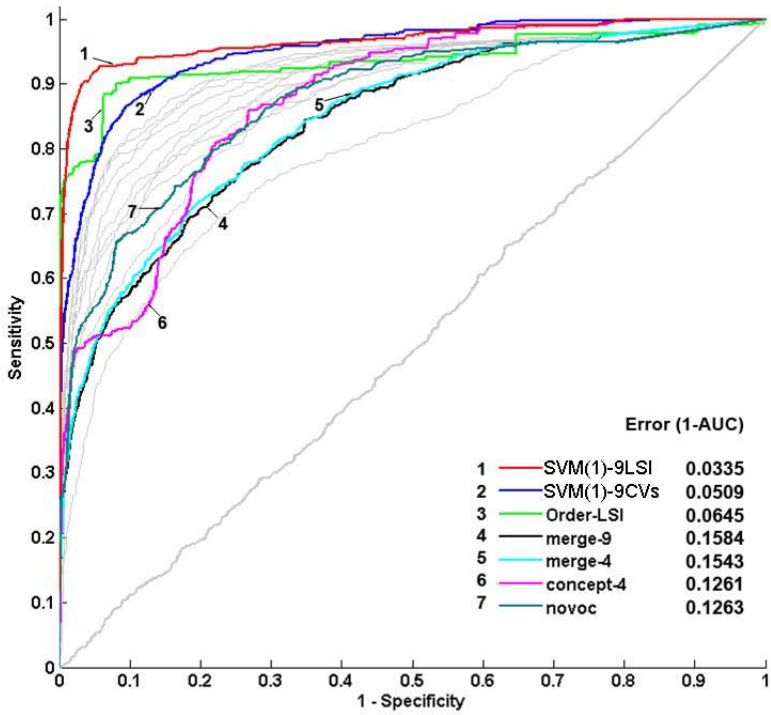


Figure 8.5: ROC curves of prioritization obtained by various integration methods. The light grey curves represent individual textual data. The near-diagonal curve is obtained by prioritization of random genes.

8.6.2 Effectiveness of multi-view demonstrated on various number of views

To further demonstrate the effectiveness of multi-view text mining, we evaluated the performance on various number of views. The number was increased from 2 to 9 and three different strategies were adopted to add the views. Firstly, we simulated a random strategy by enumerating all the combinations of views from the number of 2 to 9. The combinations of 2 out of 9 views is C_9^2, 3 out of 9 is C_9^3, and so on. We calculated the average performance of all combinations for each number of views. In the second and the third experiment, the views were added by two different heuristic rules. We ranked the performance of the nine views from high to low was LDDB, eVOC, MPO, GO, MeSH, SNOMED, OMIM, Uniprot, and KO. The second strategy combined best views first and increases the number from 2 to 9. In the third strategy, the irrelevant views were integrated first. The results obtained by these three strategies are presented in Figure 8.6. The performance of the random strategy increases steadily with the number of views involved in integration. In the best view first strategy, the performance increased and reached the ideal performance, then started to decrease when more irrelevant views are involved. The ideal performance of prioritization was obtained by combining the five best views (Error of AUC = 0.0431) by the 1-SVM method applied on averagely combined kernel. The generic integration method (order statistic) did not perform well on high dimensional gene-by-term data. The performance of integrating all CVs was comparable to the ideal performance, which shows that the proposed multi-view approach is quite robust to the irrelevant views. Furthermore, the merit in practical explorative analysis is that the near-optimal result can be obtained without evaluating each individual model. In the third strategy, because the combination starts from the irrelevant views first, the performance was not comparable to the random or the ideal case. Nevertheless, as shown, the performance of the multi-view approach was always better than the best single view involved in integration. Collectively, this experiment clearly illustrated that the *multi-view* approach is a promising and reliable strategy for disease gene prioritization.

8.6.3 Effectiveness of multi-view demonstrated on disease examples

To explain why the improvements take place when combining multiple views, we show an example taken from prioritization of MTM1, a gene relevant to the disease Myopathy. In the disease benchmark data set, Myopathy contains 41 relevant genes so we build the disease model by using the other 40 genes and leave MTM1 out with 99 random selected genes for validation. In order to compare the rankings, only in the experiment for this example, the 99 random candidate genes are kept

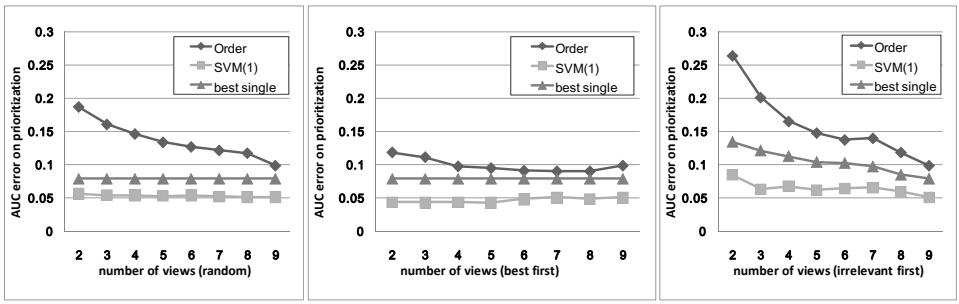


Figure 8.6: Multi-view prioritization with various number of views

identical for different views. In table 8.4, we list the ranking positions of MTM1 and the false positive genes in all 9 CVs. When using LDDB vocabulary, 3 "false positive genes" (C3orf1, HDAC4, and CNTFR) are ranked higher than MTM1. To investigate the terms causing this, we sort the correlation score of each term between the disease model and the candidate genes. It seems that C3orf1 is ranked at the top mainly because of the high correlation with terms "skeletal, muscle, and heart". HDAC4 is ranked at the second position because of terms "muscle, heart, calcium, and growth". CTNFR is ranked at the third position due to the terms like "muscle, heart, muscle weak, and growth". As for the real disease relevant gene MTM1, the high correlated terms are "muscle, muscle weak, skeletal, hypotonia, growth, and lipid". However, according to our knowledge, none of these three genes (C3orf1, HDAC4, and CNTFR) is actually known to cause any disease. Escarceller *et al.* [57] show that C3orf1 seems to be enhanced in heart and skeletal muscle, but there is no clue that it has directly relation with the disease. HDAC4 is found in the work of Little *et al.* "as a specific downstream sbstrate of CaMKIIdeltaB in cardiac cells and have broad applications for the signaling pathways leading to cardiac hypertrophy and heart failure" [111]. In the papers of Glenisson *et al.* [68] and Cohen *et al.* [36], HDAC4 is found to have a role in muscle, which means it might be a good candidate but has not yet been proved directly related to the disease. For CNTFR, it has been found that in heterozygotic mice inactivation of the CNTFR leads to a slight muscle weakness [143]. In the papers of Roth *et al.* [147] and De Mars *et al.* [47], CNTFR is shown related to muscular strength in human. Collectively, although these 3 genes found by LDDB vocabulary have not yet been reported as direct disease causing factors, the prioritization result is still meaningful because in literature they have many similar correlated terms with the disease model as the real disease causing gene does. Especially, according to literature, HDAC4 and CNFR seem to be nice candidates to muscular disorder. Though LDDB ranks 3 "false positive" genes higher than the real disease relevant gene, eVOC and GO rank the left out gene MTM1 as the top candidate gene. In eVOC, the most important correlated terms are "muscle, sever, disorder, ...". In GO, the terms are "muscle, mutation, family, sever, ...". When combining multi-

view data for prioritization, the ranking of LDDB is complemented by eVOC and GO thus MTM1 is ranked as the top gene.

Table 8.4: Prioritization results of MTM1 by different CV profiles

CV	Rank	Gene	correlated terms
LDDB	1	C3orf1	muscle, heart, skeletal
	2	HDAC4	muscle, heart, calcium, growth
	3	CNTFR	muscle, heart, muscle weak, growth
	4	**MTM1**	muscle, muscle weak, skeletal, hypotonia, growth, lipid
eVOC	**1**	**MTM1**	muscle, sever, disorder, affect, human, recess
MPO	1	HDAC4	muscle, interact, protein, domain, complex
	2	HYAL1	sequence, human, protein, gener
	3	WTAP	protein, human, sequence, specif
	4	FUT3	sequence, alpha, human
	...		
	15	**MTM1**	myopathy, muscle, link, sequence, disease, sever
GO	1	**MTM1**	muscle, mutate, family, gene, link, seqeuence, sever
MeSH	1	HYAL1	human, protein, clone, sequence
	2	LUC7L2	protein, large, human, function
	3	**MTM1**	myopathy, muscle, mutate, family, gene, missens
SNOMED	1	S100A8	protein, large, human, function
	2	LUC7L2	protein, large, human, function
	3	LGALS3	human, protein, express, bind
	...		
	23	**MTM1**	muscle, mutate, family, gene, link
OMIM	1	HDAC4	muscle, interact, protein, bind
	2	MAFK	sequence, protein, gene, asthma relat trait
	3	LUC7L2	protein, large, function, sequence
	4	SRP9L1	sequence, protein, length, function
	...		
	50	**MTM1**	muscle, family, gene, link, sequence, disease, sever, weak
Uniprot	1	**MTM1**	gene, protein, function
KO	1	S100A8	protein, bind, complex, specif, associ, relat
	2	PRF1	specif, protein, contain, activ
	...		
	56	**MTM1**	protein, large, specif, contain
Multi-view	**1**	**MTM1**	
	2	HDAC4	
	3	CNTFR	

8.7 Discussion and Summary

The strategy of model integration has already been used in several text mining applications. The concept of *multi-view* is proposed by Bickel and Scheffer [20] in web document clustering analysis to combine intrinsic view of web pages (text based similarity) and extrinsic view (citation link based similarity). In our paper, we borrow the term *multi-view* in the text mining context as gene profiles represented by different CVs. Neveol *et al.* [132] combine three different methods (dictionary lookup, post-processing rules and NLP rules) to identify MeSH main head-ing/subheading pairs from medical text. Chun *et al.* [34] develop an integrative system to extract disease-gene relations from MEDLINE. Jimeno *et al.* [88] combine three methods (dictionary look-up, statistical scoring, and MetaMap) to recognize disease names on a corpus of annotated sentences. Gaulton *et al.* [63] adopt 3 different ontologies and 8 data sources in the CAESAR system to annotate human genes as disease associated candidates. When annotating multiple data sources with different relevant terms from ontologies, each gene may get multiple scores of relevance with the input text query. CAESAR combines the scores using 4 basic methods: maximum, sum, normalized average, and a transformed score penalized by the number of genes annotated in a given data source. Our approach is different from CAESAR by exploiting all the relevant MEDLINE abstracts for indexing so the gene textual profiles are retrieved from vast amounts of gene-based free-text information available in the literature. Yamakawa *et al.* [190] combine 3 different sources (GO, Locuslink and HomoloGene) to create gene list annotated by GO terms. Then, a decomposition method (ETMIC situation decomposition) is applied to extract the multi-aspect information from the target gene list, resulting in several bipartite graphs describing the relationships between small subset of genes and GO terms. Their approach shares the same motivation as ours in obtaining more refined characteristics of genes separated in different aspects (views). However, their approach has not shown how multi-aspect gene annotations can improve the process of biomedical knowledge discovery. The main limitation of combining the LSI with data fusion is that the latent factors cannot be easily interpreted so it is hard to investigate the important terms in prioritization. The idea of *multi-view* text mining is not restricted to the selection of CVs. For example, instead of using the curated GeneRIF as the mapping of genes to publications, one can detect gene names expressed in the text automatically by natural language processing (NLP) and create new gene profiles according to this new mapping. One can also retrieve the literature co-occurences of genes and produce new *view* about the gene-to-gene interactions. Combining these views will probably lead to new insight about the relationship between diseases and genes.

Chapter 9

Endeavour MerKator: an open software for cross-species gene prioritization

9.1 Introduction

In the preceding chapter we used multiple CVs to extract information from MEDLINE and combined the multi-view text mining data for gene prioritization. The text mining data is a useful source, yet not complete, in the scheme of genomic data fusion shown in Figure 1.3. Textual information is known highly relying on the published results and the accuracy of prioritization may decrease if the publication data were "rolled back" to the moment before the critical publications were published. A more comprehensive approach is to combine textual information with biological data to leverage the effect between reliability and novelty, which has becomes a state-of-art method in candidate gene prioritization. Most of the existing approaches, as mentioned in the previous Chapter, are restricted to integrating information in a single species of interest. Recently, people start to collect phylogenetic evidences among multiple species to facilitate the identification of candidate genes in prioritization. Hutz *et al.* [81] develop CANDID algorithm to combine cross-species conservation measures and other genomic data sources to rank candidate genes that relevant to complex human diseases. In their approach, they adopt the NCBI's HomoloGene database which analyzes genes from 18 organisms and collects homologs using amino acid and DNA sequences. Liu *et al.* [112] investigate the effect of adjusting gene prioritization results by cross-species comparison. They identify the ortholog pairs between between

Drosophila melanogaster and Drosophila pseudoobscura by BLASTP and use this cross-species information to adjust the rankings of the annotated candidate genes in D. melanogaster. They find that a candidate gene with lower score in the main species (D. melanogaster) may be re-ranked higher if it exhibits strong similarity to orthologs in coding sequence, splice site location, or signal peptide occurrence [112]. According to the evaluation on the test set of 7777 loci of D. melanogaster, the cross-species model outperforms other single species models in sensitivity and specificity measures.

In this chapter we present an open software for gene prioritization called Endeavour MerKator, which is featured as the combination of cross-species homologies detection and genomic data fusion. The new software is developed on the databases of Endeavour [3] and the kernel fusion novelty detection methodology [46]. The developmental work mainly consists of two parts. Firstly, we create a "kernelized" version of Endeavour: we transform all the genomic data sources into kernel matrices and develop a java based 1-SVM MKL algorithm to prioritize the candidate genes by kernel fusion. Secondly, we extend the prioritization algorithm from single species to multiple species. To our knowledge, Endeavour MerKator is probably one of the first real bioinformatics softwares powered by kernel method. It is also one of the first cross-species prioritization softwares freely accessible online.

9.2 Conceptual overview of Endeavour MerKator

Comparing to its previous version, Endeavour MerKator is featured as the kernel based integration (1-SVM) of prioritization results. Moreover, it also incorporates cross-species information. The user specifies a group of training genes and candidate (test) genes in the organism of main interest, denoted as the *main organism*. MerKator identifies the homologs of the main organism genes from the *reference organisms*. The identification is based on NCBI's HomoloGene [185], which provides the mapping of homologs among the genes of 18 completely sequenced eukaryotic genomes. MerKator investigates the homolog mappings among 5 species, *Homo sapiens, Mus musculus, Rattus norvegicus, Drosophila melanogaster,* and *Caenorhabditis elegans*. For each gene in each organism, MerKator stores the homolog pair with the lowest ratio of amino acid differences (the *Stats-prot-change* field in HomoloGene database). Thus when the training and candidate set of the main organism genes are given, MerKator constructs the corresponding training and candidate sets in all reference organisms. Within each organism, MerKator prioritizes the candidate genes with the model built on training genes by the 1-SVM MKL algorithm [46] introduced in Chapter 3 and Chapter 8. Then MerKator combines multi-species prioritization results together as the final cross-prioritization result, where the mathematical formulations will be

Table 9.1: Genomic data sources adopted in Endeavour MerKator

data source	H.sapiens	M.musculus	R.norvegicus	D.melanogaster	C.elegans
Annotation GO	◯	◯	◯	◯	◯
Annotation-Interpro	◯	◯	◯	◯	◯
Annotation-EST	◯	◯	×	×	×
Sequence-BLAST	◯	◯	◯	◯	◯
Annotation-KEGG	◯	◯	◯	◯	◯
Expression-Microarray	2	3	1	2	1
Annotation-Swissprot	×	◯	◯	◯	◯
Text	◯	×	×	×	×
Annotation-Phenotype	×	×	×	◯	×
Annotation-Insitu	×	×	×	◯	×
Motif	◯	×	×	×	×
Interaction-Bind	◯	×	×	×	×
Interaction-Biogrid	◯	×	×	◯	◯
Interaction-Mint	×	×	×	◯	◯

introduced in the next section. The conceptual workflow of Merkator is depicted in Figure 9.1.

MerKator integrates multiple genomic data sources in multiple species for gene prioritization. The complete list of the data sources adopted in the current version is presented in Table 9.1.

9.3 Implementation of Endeavour MerKator

MerKator fully relies on the kernel fusion techniques to solve full genomic prioritization problems. In this section we will discuss several challenges in its implementation and present how we tackle these problems.

9.3.1 Offline and online processes of the software

MerKator separates the program into two parts, the offline process and the online process, to improve its efficiency on full genomic prioritization tasks. The main workflows of the offline and the online processes are shown in Figure 9.2. The main computational burden in MerKator is the kernel computation of various data sources in the full genomic scale, especially for the data which is represented in high dimensional space, such as Gene Ontology Annotation, Sequence String, Text mining, and so on. To tackle this difficulty, MerKator manages all the kernel matrices in the offline process using a Matlab-Java data exchange tool. In Matlab, the tool retrieves the genomic data from Endeavour database and construct the kernel matrices. The kernel matrices of the full genomic data may be very large so it is infeasible to manage them directly in the software. To solve this, we decompose

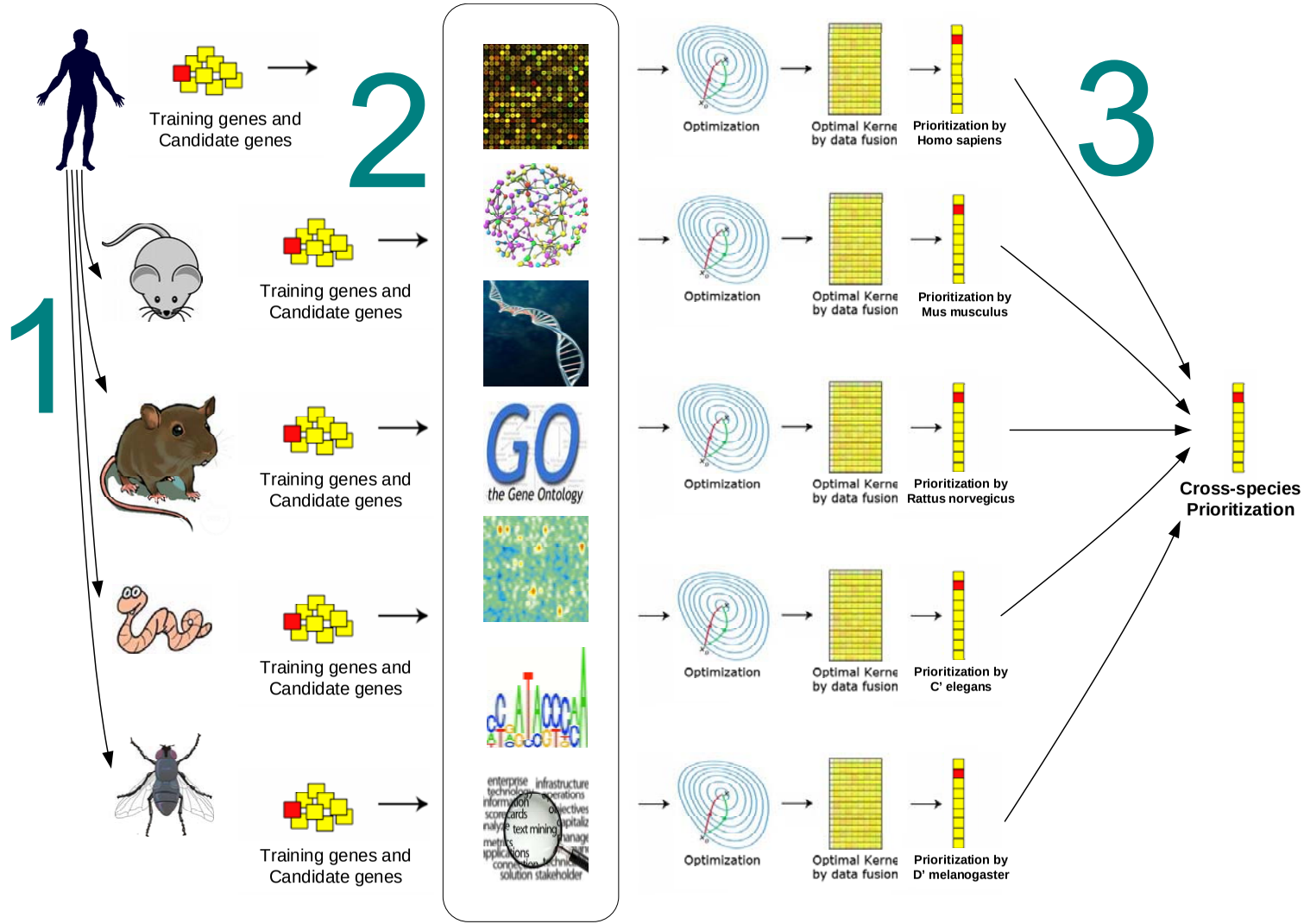


Figure 9.1: Conceptual overview of Endeavour MerKator software

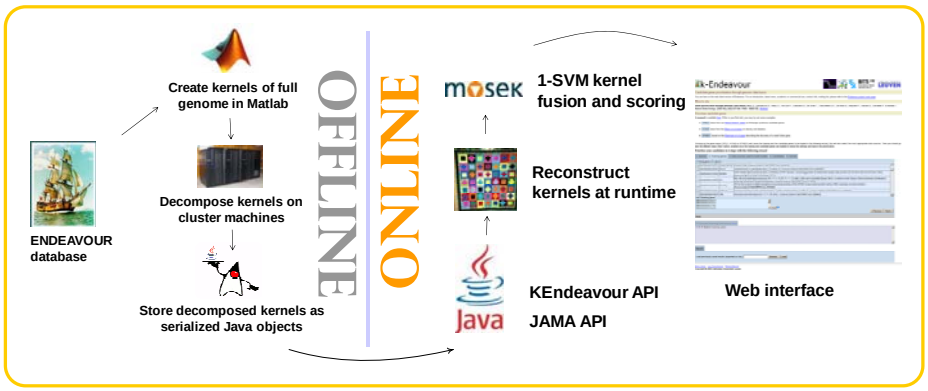


Figure 9.2: Offline and Online processes of Endeavour MerKator

all the kernel matrices by the ICD thus the dimensions of the decomposed kernel matrices are often smaller than the original data. The precision of the ICD is set as 95% of the matrix norm, given by

$$\frac{||K - K'||_2}{||K||_2} \leq 0.05, \tag{9.1}$$

where K is the original kernel matrix, K' is the approximated kernel matrix as the inner product of the ICD matrix. In this way the computational burden of kernel calculation is significantly reduced as the computation of the inner product of the decomposed matrices. The Matlab-Java tool creates Java objects on the basis of decomposed kernel matrices in Matlab and stores them as the serialized java objects. The computational burden of these processes, such as the kernel computation, decomposition and java object transformation, are quite heavy so they are all executed offline. For the online process, MerKator loads the decomposed kernel matrices from the serialized java objects, reconstructs the kernel matrices and solve the 1-SVM MKL optimization problem to prioritize the genes. Then the prioritization results are displayed on the web interface. In contrast with the offline process, the online process is lightweight where the main complexity is determined by the N number of training genes ($O(N^3)$). In our implementation, the optimization solver is based on the Java API of MOSEK [8] and it shows satisfying performance. Through this offline and online separation, we avoid the large kernel computation and make MerKator applicable for the full genomic prioritization tasks. Though in this separated structure the user cannot create kernel matrices with arbitrary parameters, in kernel fusion problems the final performance is not solely determined by a single kernel but is obtained as fusing all the pre-defined kernels, the software may still obtain near-optimal result.

9.3.2 Centering the kernel matrices

In MerKator, when the prioritization task involves data set of the full genomic size, some trivial operations become quite inefficient. To control the number of the false positive genes in 1-SVM, De Bie *et al.* suggest a strategy to center the kernel matrices of all training genes and test genes on the basis of the *iid* assumption. However, when the task is to prioritize the full genomic data, to center the full genome kernel matrices becomes very inefficient. In MerKator we use a strategy to split the full genomic data into different subsets. Let us assume that the full genome data contains N genes, which is split into several subsets containing M genes. Instead of centering the kernel matrix sizes of $N \times N$, we center the kernel matrix of size $J \times J$, where J is the number of genes in the union of the M candidate genes with the training genes. Because M is smaller than N, for each centered kernel matrix MerKator obtains the prioritization score of M candidate genes, so it need to iterate multiple times (denote as k) to calculate all the scores of N candidate genes. According to the *iid* assumption, if M is large enough then centering the kernel matrix of size $J \times J$ would be statistically equivalent to centering the kernel matrix of the full genome, thus the prioritization scores obtained from the k iterations can precisely approximate the values obtained from the full genome centering. Therefore, we may compare the prioritization scores of N genes obtained from different centered matrices and rank the full genome. All these assumptions come to one non-trivial question: how to select the appropriate M to trade off between the reliability of *iid* assumption and the computational efficiency? In MerKator, the M is determined via experiments.

Numerical experiment

The experiment is conducted on the text mining data source. In MerKator, the full human genome data contains 22743 protein coding genes. Firstly, we build the prioritization model on 20 training genes and score the 22743 genes by centering the linear kernel matrix of the full genome. The obtained values is regarded as the true prioritization scores, denoted as $\vec{f}$. We also calculate the overall computation time, denote as t. To benchmark the effect of M, we try 10 different values from 1000 to 10000. In each iteration, the 20 training genes are mixed with M randomly selected candidate genes and the prioritization scores of the candidate genes are computed by centering the small kernel matrix. In the next iteration, we select M new candidate genes until all the 22743 genes are prioritized. The prioritization scores obtained by centering this small kernel matrix are denoted as $\vec{f}'$, and the computation time is also compared. The difference (error) between the prioritization scores obtained in these two approaches represents how well the

Table 9.2: The approximation error and the computational time of using M randomly selected genes in kernel matrix centering

M	e	time(seconds)
22,743	0	11969
10000	0.015323701440092	4819.2
9000	0.022324135658694	3226.3
8000	0.028125449554702	2742.0
7000	0.048005271001603	2135.2
6000	0.041416998355952	1638.2
5000	0.048196878290559	1117.0
4000	**0.045700854755551**	**745.40**
3000	0.087474107488752	432.76
2000	0.098294618397952	191.95
1000	0.136241837096454	72.34

M candidate genes approximates the *iid* assumption of the full genome, given by

$$e = \frac{||f - f'||_2}{||f||_2}. \tag{9.2}$$

We use this difference to find the optimal M. According to the benchmark result presented in Table 9.2, large M value leads to small error but takes much longer time for the program to center the kernel matrix. In MerKator, we set the M to 4000.

9.3.3 Missing values

In bioinformatics application, clinical and genomic datasets are often incomplete and the values can be imprecise. In MerKator, there are also a considerable number of missing genes in the genomic datasets used for prioritization (Table 9.2). In MerKator, the missing gene profiles are represented as zeros in the kernel matrices mainly for computational convenience. However, zero value still contains strong information so it may lead to imprecise prioritization score. In data fusion, the problem of missing value may have two cases: the first case is a gene profile is missing in part of the data sources; the second case is a gene profile is missing in all the data sources. In MerKator, we use a strategy illustrated in Figure 9.3 to combine kernel matrices with missing values. The combined kernel matrix obtained by this strategy is also PSD and the obtained prioritization scores only rely on the non-missing information among different data sources. This strategy may not be perfect, because in kernel centering the zero values still influence the similarity

Table 9.3: Ratio of missing genes with the number of protein-coding genes in MerKator

	H.sapiens	M.musculus	R.norvegicus	D.melanogaster	C.elegans
Annotation GO	26.67%	36.13%	51.51%	39.47%	53.37%
Annotation-Interpro	21.53%	29.96%	27.76%	35.12%	36.01%
Annotation-EST	11.26%	83.86%	-	-	-
Sequence-BLAST	0.00%	0.00%	0.00%	0.00%	0.00%
Annotation-KEGG	82.44%	86.24%	90.30%	90.64%	96.54%
Expression-MA1	40.43%	66.37%	27.76%	20.64%	19.69%
Expression-MA2	42.73%	69.14%	-	70.69%	-
Annotation-Swissprot	-	32.85%	79.46%	4.63%	71.24%
Text	24.20%	-	-	-	-

scores of other non-missing genes in the kernel matrices. A possible improvement is to refine it by centering kernel matrices only for non-missing values, but that might leads to a complicated procedure of centering different size kernels for different data sources and replacing back the missing profiles for kernel combination, and the final result may not be much different with the results obtained in our strategy.

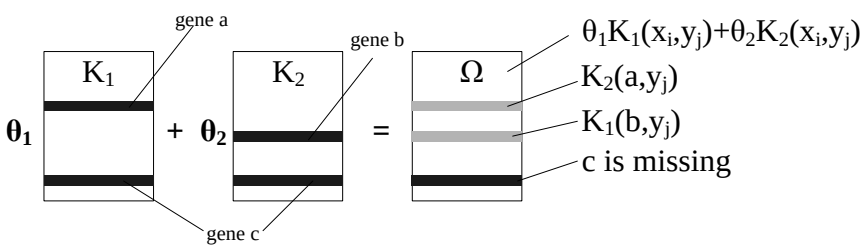


Figure 9.3: Kernel fusion with missing values. Suppose gene a and gene c are missing in the first kernel $K1$; gene b and gene c are missing in the second kernel $K2$; θ_1, θ_2 are the kernel coefficients. In the combined kernel Ω, we fill the missing values of gene a by the non-missing information in $K2$ and gene b by the non-missing information in $K1$. The gene c is missing in all data sources so it will not be prioritized in MerKator. For the case with more than 2 data sources, the coefficients of missing values are recalculated by considering non-missing sources only.

9.3.4 Cross-species information integration

This is an ongoing research and the content may change in the final version of the thesis

MerKator first uses the 1-SVM algorithm to prioritize genes in a single species, and then adopts a Noisy OR model [139] to integrate prioritization scores from multiple species. The integration model is given as follows. We assume the scenario of cross-species prioritization as depicted in Figure 9.4. Suppose the training set of the main species contains N_1 number of human genes as $\{H_1, ..., H_{N_1}\}$ and MerKator obtains the corresponding sets in reference species rat and mouse. The training set in rat contains N_2 number of genes which are denoted as $\{R_1, ..., R_{N_2}\}$ and the mouse set contains N_3 genes denoted as $\{M_1, ..., M_{N_3}\}$. Note that MerKator always selects the homolog with the highest similarity ratio of sequence, so it is a *many-to-one* mapping and N_2, N_3 are always smaller or equal to N_1. We define the homolog scores between human and rat training sets as $a_1, ..., a_{N2}$; Similarly, the homolog scores between human and mouse training sets are $b_1, ..., b_{N3}$. For a test set, each candidate gene of human is mapped to at most one rat gene and mouse gene, where the homolog score is respectively denoted as c_0 and d_0. First, we introduce some necessary utility parameters: $h1$ and $h2$ are the parameters describing the quality of homolog scores, given by:

$$h1 = \min\{c_0, \text{median}(a_1, a_2, ..., a_{N2})\},$$

$$h2 = \min\{d_0, \text{median}(b_1, b_2, ..., b_{N3})\}.$$

z_1 and z_2 are the parameters describing the ratio of the number of homologs in the reference species with the number of genes in the main species, given by

$$z_1 = \frac{N_2}{N_1}, \quad z_2 = \frac{N_3}{N_1}. \tag{9.3}$$

Next, we denote f_0 as the prioritization score of candidate gene H_0 ranked by the training set $\{H_1, ..., H_{N1}\}$; denote f_1 as the score of reference candidate gene R_0 prioritized by the reference training set $\{R_1, ..., R_{N2}\}$ and f_2 as the score of M_0 ranked by the set $\{M_1, ..., M_{N3}\}$. The prioritization score obtained by the 1-SVM is in the range of $[-1, +1]$, thus we scale it into $[0, +1]$. The adjustment coefficient adj is defined as:

$$adj = 1 - \prod_{\text{organism } i} (1 - h_i z_i f_i)\,. \tag{9.4}$$

The adj coefficient in (9.4) combines information from multiple species by the Noisy OR model and a larger adj means there is strong evidence from the homologs that the candidate gene is relevant to the model. This coefficient is used to adjust

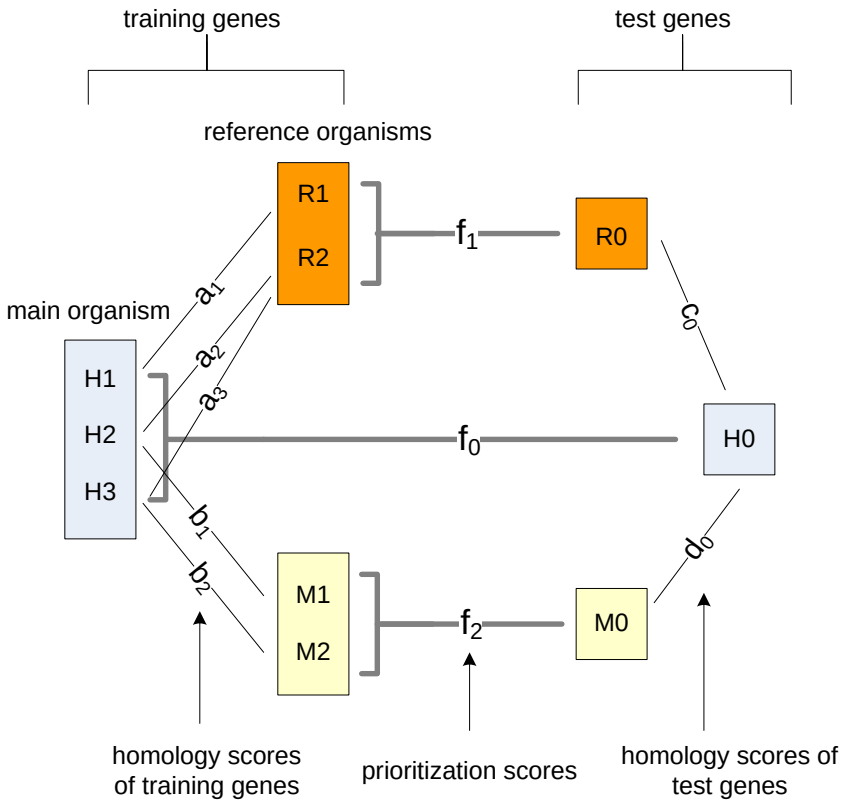


Figure 9.4: Integration of cross-species prioritization scores

f_0, the prioritization score of the main organism, and we have tried two different versions as follows:

$$\text{human non-special: } f_{\text{cross-species}} = 1 - (1 - f_0)(1 - adj), \tag{9.5}$$

and

$$\text{human special: } f_{\text{cross-species}} = 1 - \frac{(1 - f_0)(2 - adj)}{2}. \tag{9.6}$$

The *human non-special* version considers the human prioritization score as equivalent to the homology evidence and combines them as Noisy OR. In contrast, the *human special* version only adjusts the human prioritization score with the homology evidence by average. In the Noisy OR integration, the cross-species score is boosted up if either the main species or the homology evidence shows good prioritization score. In the average score integration, the cross-species score

compromises between the homology evidence and the main species score, and only is boosted up if both of them are good.

Bias of homology: good or bad?

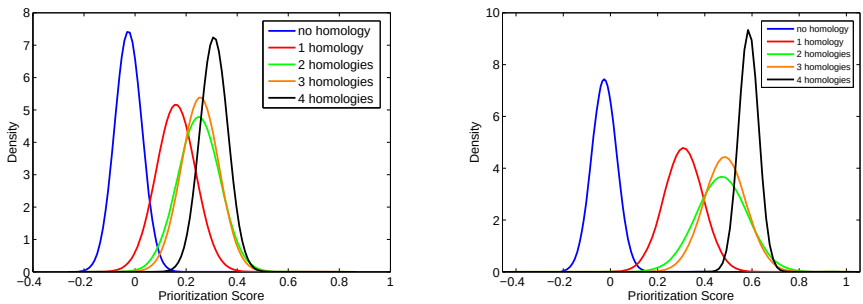


Figure 9.5: The bias of homolog in cross-species prioritization scores. The scores are obtained using the *Lymphoma* disease training set. Among the 22743 human genes, 7530 genes are not related to any homolog in the four reference organisms; 6188, 5516, 2370 and 1139 of them have 1 homolog, 2 homologs, 3 homologs and 4 homologs respectively. Their scores are fit to normal distributions and plotted in different colors. The left figure shows the human special scores computed by (9.6). The right figure shows the human non-special scores obtained in (9.5).

To show the result, we present full genome prioritization scores of human (as the main organism) combined with other four organisms. All the available data sources listed in Table 9.1 are used for prioritization. We first try the training set of *Lymphoma* disease (31 genes) and prioritize the full human genome (22743 protein-coding genes). We plot the histogram of the scores and fit the distribution [1]. We separate the scores into five groups according to the number of homologs to which a candidate gene is mapped. As shown in Figure 9.5, the cross-species prioritization scores computed by (9.5) and (9.6) are strongly affected by the number of homologs. This is because when there is no homolog found related to the candidate gene, the *adj* in (9.4) equals to 0. It increases monotonically when more homologs are found in different organisms because the chain product term is always smaller than 1. Larger *adj* values will boost up the prioritization score in (9.4) and (9.5). Apparently, the cross-species prioritization favors those genes which have homologs; It is also possible for a gene without homolog, which was ranked highly in a single organism, to get very low rank in the cross-species result. Whether to accept or to reject this *homolog bias* depends on the motivation of the cross-species

[1]The purpose is to intuitively show the relationship between the scores and the number of homologs. There is no hypothesis or conclusion made on the distribution estimation.

prioritization. Considering the case one may want to eliminate this bias, we modify the function in (9.4) and (9.5) by a redefining the adj parameter, denoted as adj^+, given by

$$adj^+ = \begin{cases} \text{median}(\{adj\}), & \text{if } j \text{ has no homolog} \\ 1 - \{\prod_{\text{organism } i} (1 - h_i z_i f_i)\}^{\frac{1}{k}} & \text{if } j \text{ has homolog.} \end{cases} \tag{9.7}$$

The first case of (9.7) means when there is no homolog related to gene j, adj^+ is equal to the median value of the set $\{adj\}$ who contains the values of homolog-related genes. In implementation, the $\{adj\}$ is calculated first, then we remove all the 1 values (genes without homolog) from the set and calculate the median. In the second case, when there is a homolog mapped to gene j, k is equal to the number of homolog gene that j is related. By taking the k-th exponential root, it removes the additional bias of the prioritization score caused by multiple homologs. We recompute the cross-species prioritization scores by this new adj^+ and plot the distribution of the scores in Figure 9.6.

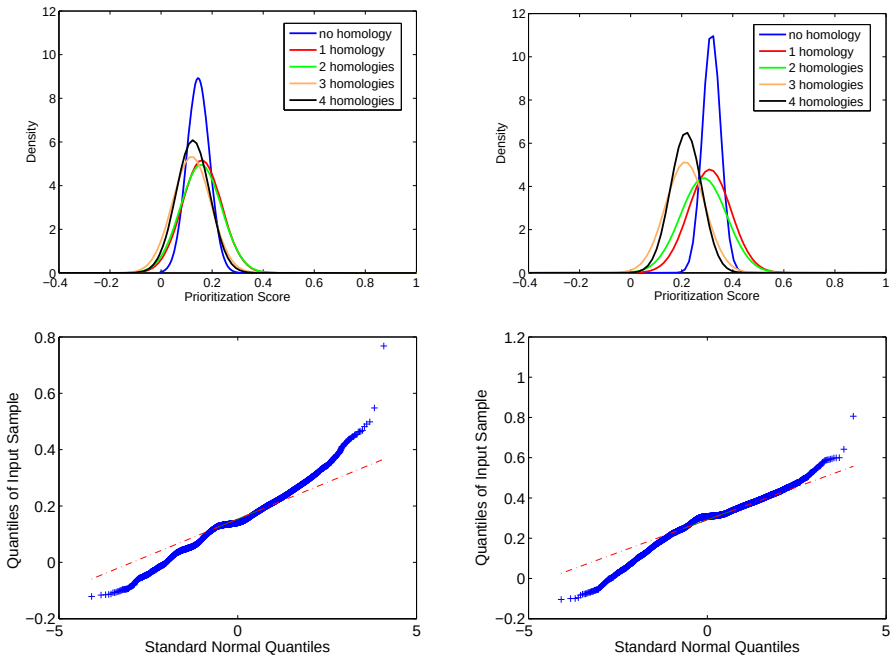


Figure 9.6: Eliminating the homolog bias by the new adj^+ parameter defined in (9.7). The scores are obtained using the *Lymphoma* disease training set (31 genes). The top left figure plots the human special scores computed by adj^+; the top right figure shows the human non-special scores; the qq-plot on the bottom left shows the sample quantiles of the human special scores of all genes versus the normal distribution; the qq-plot on the bottom right shows the human non-special scores versus normal distribution.

We also validate the adjustment on different training sets, mainly concerning the number of training genes. In Figure 9.7 we plot the results obtained using the *Leukemia* training set which contains 112 training genes. In Figure 9.8 we plot the results obtained using the *Myopathy* training set which contains 41 training genes. It seems that the modified formula correctly adjusts the homolog bias on training sets with different size.

Finally, we compare the ranking of the cross-species approach with the ranking obtained only in human. The comparisons are illustrated in Figure 9.9 through Figure 9.12. Results obtained by four different formulations are shown and compared with the human prioritization result.

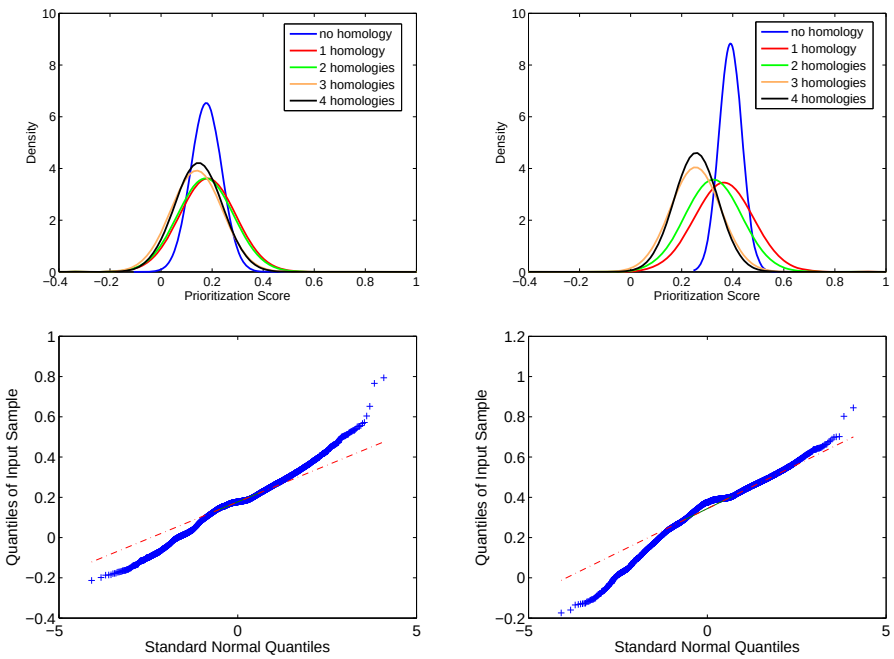


Figure 9.7: Eliminating the homolog bias by the new adj^+ parameter defined in (9.7). The scores are obtained using *Leukemia* disease training set (112 genes). The arrangement and description of figures are the same as in Figure 9.6.

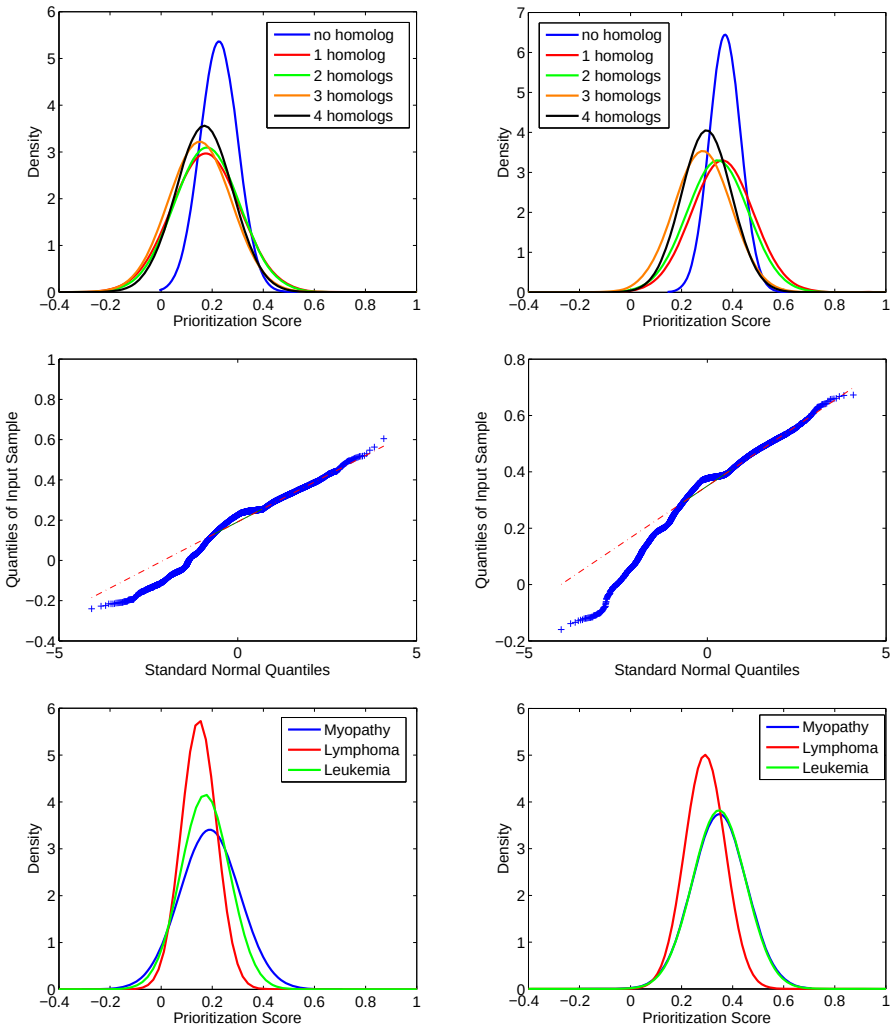


Figure 9.8: Eliminating the homolog bias by the new adj^+ parameter defined in (9.7). The scores are obtained using the *Myopathy* disease training set (41 genes). The top left and top right figures show the distribution of human special and human non-special scores; the middle figures show the qq-plot; the bottom figures compare the distributions of all scores obtained by three disease training sets, the left one illustrates the human special score and the right one depicts the human non-special score.

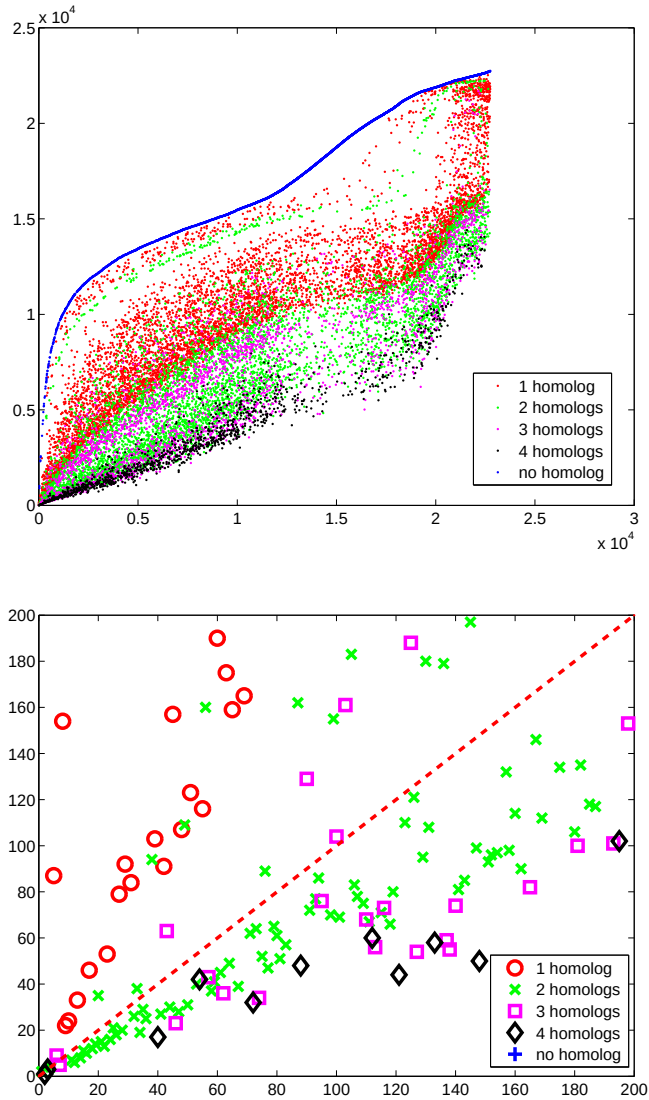


Figure 9.9: Comparison of the human based ranking with cross-species ranking using the human special homolog biased score. The top figure compares the rankings of all 22743 genes. It clearly shows that most of the blue dots lie at the upper-triangle, which means those genes get lower rank in cross-species approach because they have no homolog. The bottom figure compares the first 200 ranking positions. If more genes appear in the figure on the bottom, it means there are more agreements about the top ranked genes between the cross-species and the single organism prioritization. Moreover, the markers close the diagonal line represent those genes get identical rankings in the two approaches. As shown, there are a considerable amount of genes with similar rankings in the two approaches, and the genes who have more homologs often get higher ranking in the cross-species approach.

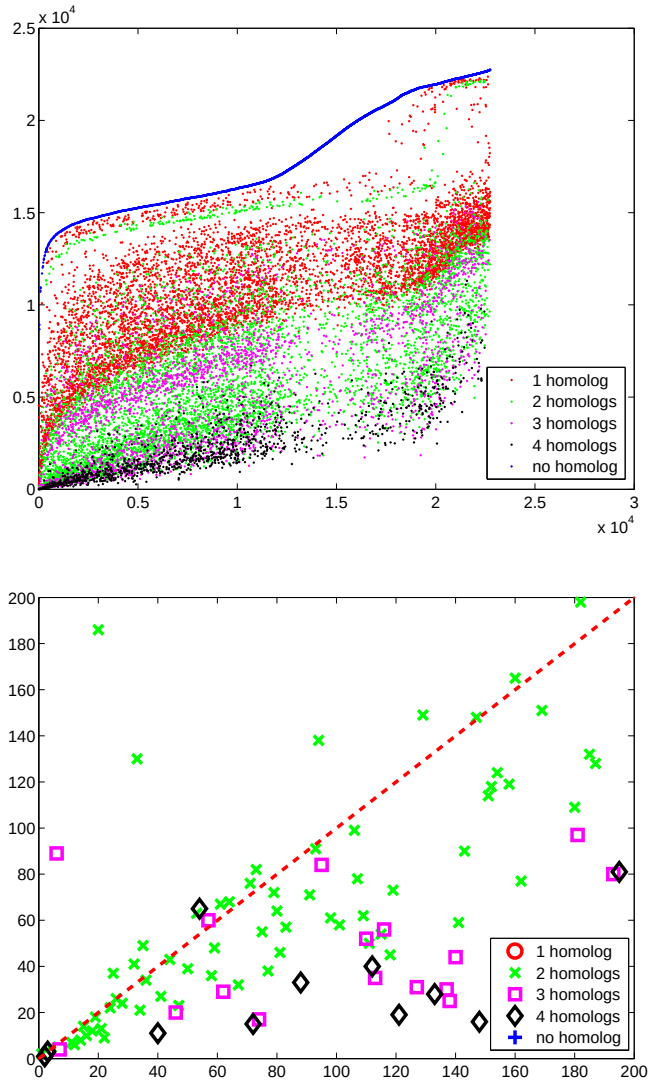


Figure 9.10: Comparison of the human based ranking with cross-species ranking using the human non-special homolog biased score. In the top figure, the distributions of markers are not so close to the diagonal as in Figure 9.9. This is because that this version doesn't regards human score as special so the rankings are less similar to the human rankings. In the bottom figure, there is rarely any genes who has no homolog and 1 homolog is ranked as the top 200.

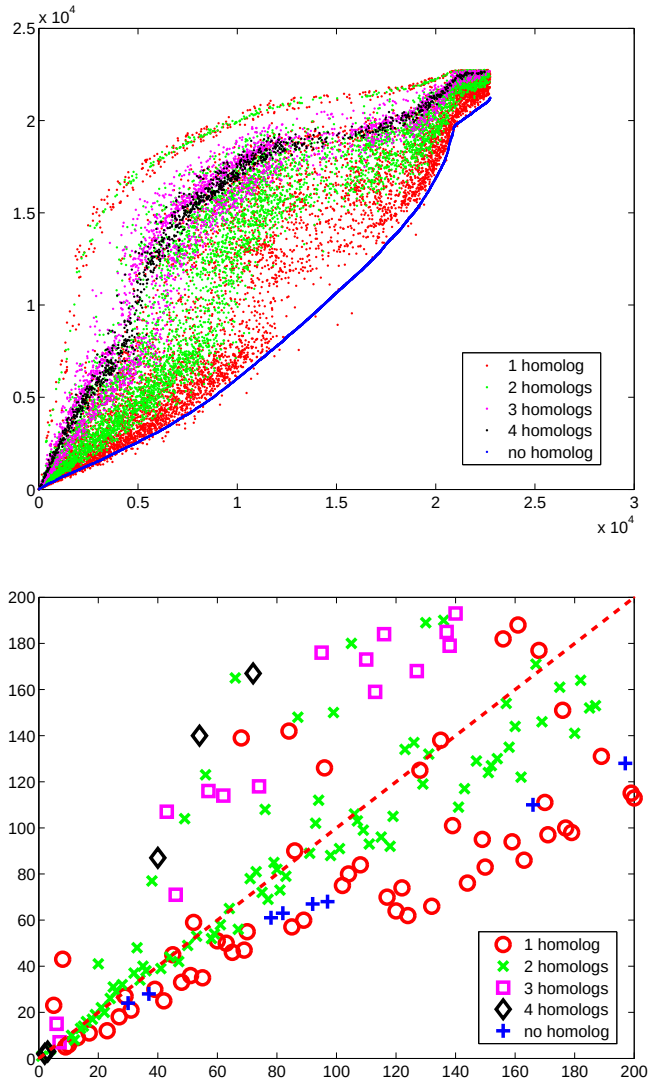


Figure 9.11: Comparison of the human based ranking with the cross-species ranking using the human special homolog unbiased score. Comparing with Figure 9.9, the main difference of this figure is that those genes without homolog also get similar rank, or even higher rank than the human organism in the cross-species approach. This result is coincident to our motivation to eliminate the homolog bias. However, the top 200 rankings still mainly consist of homolog-related genes. The unbiased score penalizes those genes who have more homologs, so in the figure many genes who have exactly 1 homolog are ranked higher in the cross-species approach. Those who get the same rankings with human only are mostly genes who have two homologs (the green markers near the diagonal line).

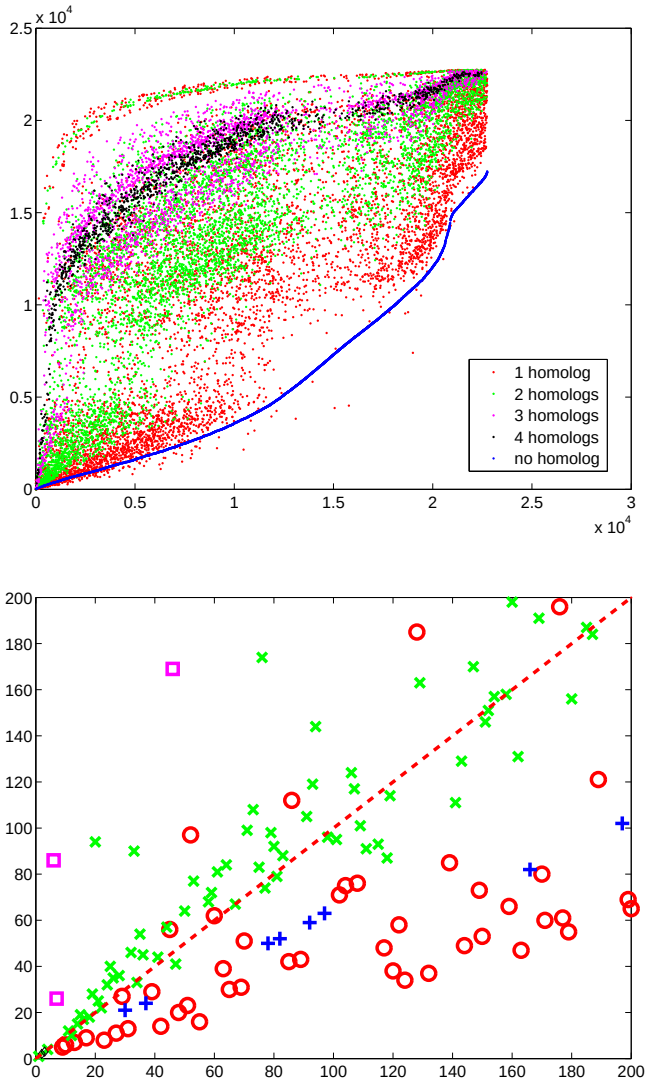


Figure 9.12: Comparison of the human based ranking with the cross-species ranking using the human non-special homolog unbiased score. The results in this case are quite similar to the human special unbiased one (Fig 9.11). The main difference in this figure is the distributions in the top figure are less close to the diagonal because the human scores are not considered as special in this case.

9.4 Summary

This chapter illustrated the Endeavour MerKator software to combine cross-species information and multiple genomic data sources to rank the candidate disease related genes. The software was developed using the same databases of Endeavour, but equipped with kernel fusion technique. To embed kernel methods in a real and large scale bioinformatics application, we tackled many challenges as some strategies are mentioned in this chapter. Our approaches may be concluded as the following three aspects:

- *Near optimal solution.* The performance of kernel based algorithms is strongly affected by the selection of hyper-parameters, such as the parameter of kernel function, the regularization parameter, etc. The optimal parameter should be selected by cross-validation, which may not be always feasible for a software oriented for biologists and medical researchers. Kernel fusion technique allows the developers to preselect the kernel parameters empirically. Then the overall performance does not rely on a single kernel parameter, so even when the optimal parameter is not involved, the fusion procedure still can leverage among several near optimal parameters and provides a near optimal result. For real applications, the 1% difference of performance is not so critical to the end users. In most cases, a successful application prefers much on the speed of solution than the very optimal parameter or model.

- *Missing values.* Real applications are full of missing values, which may raise difficulties for algorithms to obtain precise results. Estimating the missing values in kernel based applications is also an ongoing issue, however, we solved it in MerKator with a simple strategy as addressed in Section 9.3.3: the kernels are only combined for non-missing values. Though there are some approaches presenting formal descriptions and solutions for the missing value problem, they usually require much computational effort and may not appropriate to be implemented as online processes in MerKator. These methods may be useful to estimate missing values in the offline manner, but they still need to be validated by the benchmark data in the specific problems.

- *Combining evidences from multiple species.* We proposed a Noisy OR model to combine prioritization scores from multiple organisms. We tried four different formulations to calculate the final score. These formulations vary by the purpose to preserve or eliminate the strong bias caused by the homologies. The issue of multiple species prioritization is complicated, which may involve many factors such as the size of training set, the selection of data sources, the number of relevant homologies, and so on. Considering so many factors, it is almost infeasible to make hypothesis, or estimate the data model for the

final prioritization score. A practical solution is rather to avoid assumption about the data and regard it being generated by a blackbox. The integration methods are adjusted in the blackbox and the outputs are validated with benchmark data until satisfying performance is obtained.

Chapter 10

Clustering by heterogeneous data sources

10.1 Introduction

In this chapter we review and conclude a set of algorithms for data fusion based clustering. The goal of clustering is to group samples into homogeneous groups, where the homogeneity is usually measured as distances or proximities among the samples. Recently, many applications face the requirement of clustering with multiple data sources. This is because a single data source may has limited information, therefore, combining multiple sources may facilitate the comprehensive understanding of the problem. For instance, in order to investigate memory persistence (long term or short term memory) of bacteria, a bacterium is observed at different experimental conditions and developmental times [188]. Then these multiple experimental observations are combined in clustering analysis to quantify the "memory" of the bacterium in different developmental stages. In scientometrics, a strategy has been proposed to combine text mining data and bibliometrics data (hybrid clustering) to explore the structure mapping of journal sets [84] [85]. In bioinformatics, high-throughput techniques produce a large amount of genomic data. The challenge to endow clustering algorithms with the ability to retrieve correlated or complementary information about the underlying functional partitions of genes and proteins has attracted much interests [197] [9] [102] [141]. Unfortunately, though the machine learning community has already focused on data fusion for classification, the extension to unsupervised learning such as clustering, is still an unresolved and ongoing problem.

10.2 Clustering algorithms for data fusion

In clustering analysis, one may always need to try many candidate algorithms because there is no golden rule to determine the best algorithm. Though we have introduced two novel algorithms in Chapter 5 and Chapter 6, they may not always work well for all purposes. In this section, we review and conclude several existing algorithms and purpose a set of solutions for data fusion based clustering.

10.2.1 Ensemble clustering

Clustering algorithms for data fusion can be roughly divided into two main categories. The first approach is ensemble clustering, also known as clustering aggregation or consensus clustering, which combines multiple partitions into a consolidate partition by consensus function. Different ensemble algorithms mainly vary on the choice of consensus functions. Ensemble clustering is originally applied on a single source where various partitions are generated by different representations, dimensionality reduction techniques, and clustering algorithms. The strategy of the ensemble clustering can be extended to data fusion, where the main difference is that partitions are varied by data sources. An underlying assumption is: if the information contained in multiple sources is highly correlated, their partitions should also contain a *common agreement* thus a consolidated partition can also be obtained. Below we will outline some well know clustering ensemble algorithms.

HGPA, CSPA, MCLA

Strehl and Ghosh [162] formulate the optimal consensus as the partition that shares the most information with the partitions to combine, as measured by the Average Normalized Mutual Information. They use three heuristic consensus algorithms based on graph partitioning, called Cluster based Similarity Partition Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA) and Meta Clustering Algorithm (MCLA) to obtain the combined partition.

Quadratic Mutual Information Clustering (QMI)

Topchy *et al.* [171] formulate the combination of partitions as a categorical clustering problem. In their approach, a category utility function is adopted to evaluate the quality of a "median partition" as a summary of the ensemble. They prove that maximizing this category utility function implies the same clustering ensemble criterion as maximizing the generalized mutual information based on

quadratic entropy. Furthermore, the maximization of the category utility function is equivalent to the square error based clustering criterion when the number of clusters is fixed. The final consensus partition is obtained by applying the k-Means algorithm on the feature space transformed by the category utility function.

Evidence Accumulation Clustering with Average Linkage (EACAL)

Fred and Jain [59] introduce the concept of Evidence Accumulation Clustering (EAC) that maps the individual data partitions as a clustering ensemble by constructing a co-association matrix. The entries of the co-association matrix is interpreted as the votes on the pairwise co-occurrences of objects, which is computed as the number of occurrences each a pair of objects appear in the same cluster of an individual partition. Then the final consensus partition is obtained by applying Single Linkage Clustering (SL) and Average Linkage Clustering (AL) methods on the co-association matrix. According to their experiments, average linkage performs better than single linkage so in this paper we apply EACAL as the representative algorithm for comparison.

Adaptive cumulative Voting (AdacVote)

Ayad and Kamel [10] propose a "cumulative vote weighting method" to compute an empirical probability distribution summarizing the ensemble. The goal of this ensemble is to minimize the average squared distance between the mapped partitions and the combined partition. The cumulative voting method seeks an adaptive reference partition and incrementally updates it by averaging other partitions to relax the dependence of the combined partition on the selected reference. In the AdacVote they proposed, the partitions are combined in the decreasing order of their entropies.

10.2.2 Kernel fusion

An alternative approach of clustering by data fusion may achieved by fusing the similarity matrices. A parametric approach is first proposed by Lange and Buhmann [107]. If the similarity matrices are positive semi-definite, the data integration problem can be also formulated as a kernel fusion problem, where the main difference is the integration is done before clustering (early integration). In contrast, ensemble clustering aggregates partitions after clustering (late integration). Kernel method elegantly resolves the heterogeneities of data sources by representing them as same-sized kernel matrices. Moreover, if we assume that the importance of each data source is equivalent, we can combine the kernels in an average manner, thus the issue of data integration is transparent

to the clustering algorithm. The combined average kernel is a new data source, thus the partition can be obtained by standard clustering algorithms in kernel space. Furthermore, we can also apply a more machine-intelligent approach by coupling the optimization problem of kernel learning with the objective function of pattern analysis. In that case, the weights assigned on each data source can be adjusted adaptively during the clustering procedure, as the proposed OKKC and OKLC algorithms in the thesis.

OKKC

The algorithm is introduced in Chapter 5. It adaptively adjusts the coefficients of multiple kernel matrices to optimize the FDA objective for clustering.

OKLC

The algorithm is introduced in Chapter 6. It combines multiple kernel matrices and Laplacians in a spectral clustering objective. The optimal combination of kernels and the optimal combination of Laplacians are learned in two coupled optimization processes. This algorithm is extended upon OKKC and mainly proposed to integrated attribute-based data and graph-based data

KM and SC on equally combined kernels

When the kernels or Laplacians are combined equally, OKKC and OKLC reduce to the trivial KM and SC. The SC algorithm implemented in this thesis is based on the algorithm proposed by Ng *et al.* [136] and the Laplacian is computed on the averagely combined kernel.

Hierarchical Clustering

To apply the hierarchical clustering methods in feature space, we first transform the kernel matrix into a distance matrix by calculating the distances between feature vectors [155],

$$\| \phi(x) - \phi(z) \|^2 = \langle \phi(x), \phi(x) \rangle - 2\langle \phi(x), \phi(z) \rangle + \langle \phi(z), \phi(z) \rangle. \tag{10.1}$$

Then we apply the standard linkage clustering methods (SL, AL, Complete Linkage (CL), and Ward Linkage (WL)) on the transformed distance matrix and obtain the partitions by hierarchical clustering. In particular, the Ward linkage clustering algorithm based on average combination of kernels can be regarded as a special case of the WL method proposed by Janssens *et al.* [84], where the weights

assigned on data sources are determined empirically. In our paper, the weights are set as equal.

10.2.3 Clustering by data fusion

We establish a unified framework for clustering by data fusion, where various algorithms are categorized as two main approaches, clustering ensemble and kernel fusion, and their conceptual structures are shown in Figure 10.1. The software, which contains HGPA, CSPA, and MCLA algorithms, is available for download from the author's website. We implement QMI, EACAL, AdacVote, and all kernel fusion algorithms in MATLAB. In ensemble clustering, the individual partition can also be obtained by various clustering algorithms. However, to keep the discussion concise, all the individual data partitions, as well as the partitions in the ensemble clustering algorithms, are obtained by KM clustering in the original space or kernel KM clustering in the kernel space.

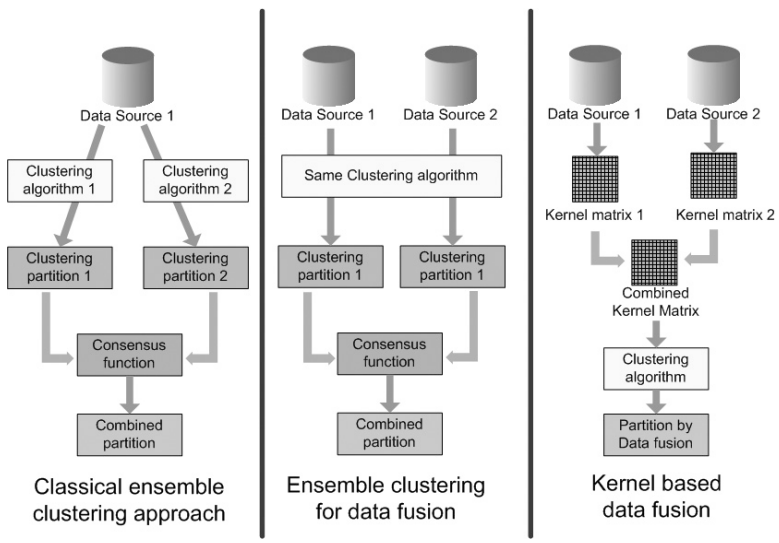


Figure 10.1: A conceptual overview of ensemble clustering and kernel fusion methods for data fusion

10.3 Experimental Results

10.3.1 Digit Recognition Data

We combine two UCI digit recognition data sources for clustering analysis. The first source is pen-based digit recognition data created by trainers input on a tablet with 500 by 500 pixels resolution. The second source is optical recognition data generated by the NIST OCR system [137]. In the literature, both data sources have been used separately as benchmark data for classification and clustering analysis. We combine them for clustering analysis for the first time. The assumption of combining them is that all samples within the same digit class are equivalent, and the data sources are two *facets* of the digit information measured in different experiments. For each clustering run, we randomly select 1500 samples (150 for each digit class) from each source and combine them for clustering. We repeat the run 50 times. To simplify the discussion, we use linear kernels only. Clustering by data fusion in nonlinear space is certainly a very interesting topic and it will be one of the main directions in our future work. To evaluate the performance of data fusion in a easy way, we use the labels of 10 digit classes from the original UCI repository as reference partitions. The two data sources are first clustered separately and their individual performances are shown in Figure 10.2 as dark bars. As shown, optical data is more informative because its RI and NMI score are significantly higher than those of the pen-based data. Then we combine these two sources for clustering, 4 algorithms obtain significant improvement (WL, KM, SC, and OKKC) compared to individual data. To show why kernel fusion improves the clustering performance, a figure showing the plot of similarity matrices of optical data, pen data, and the combined matrix is presented in Figure 10.3. As shown, the optical data has better quality for clustering than pen data. When merging the similarity matrices together, the within-class similarities are enhanced. Especially, as can be judged visually in the optical data, most of the error come from the digit 9 and the digit 0. In the contrast, in the pen data the partition between the digit 9 and the digit 0 is much clearer. After merging their similarity matrices together, the distinction of the digit 9 and the digit 0 is much more obvious than it is in the optical data source.

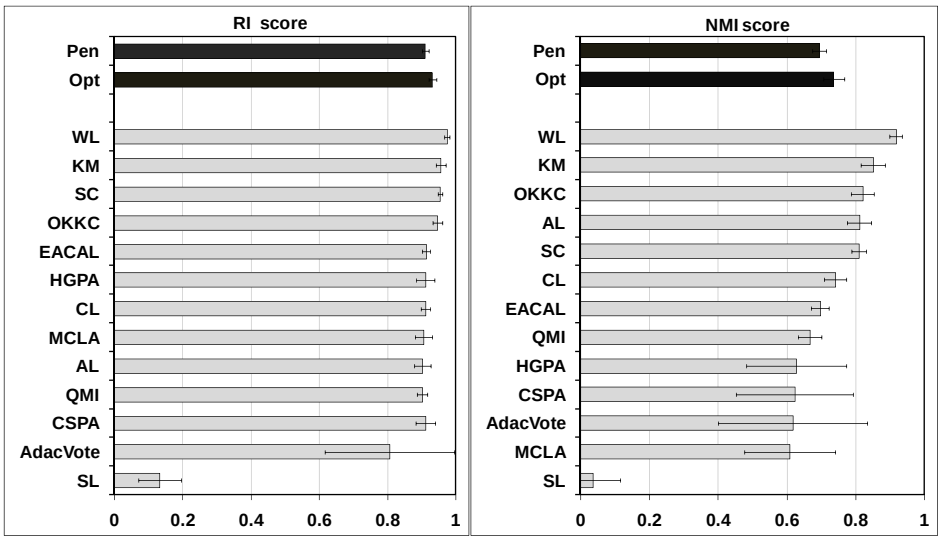


Figure 10.2: Performance of clustering on digit recognition data fusion

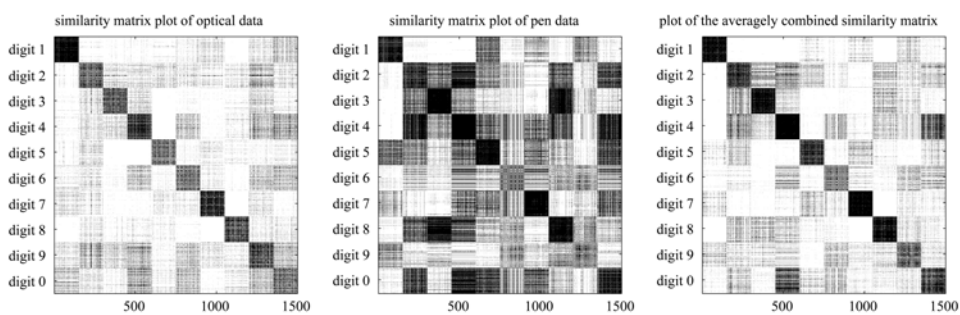


Figure 10.3: Plots of similarity matrices generated by 2 digit recognition data sources. The data is grouped from digit 1 to 0 so the distinctions among each clusters are more obvious. The dots in the figure are generated by setting a threshold 0.8 on similarity values.

Table 10.1: Statistical analysis of performance

	RI			NMI		
Best single data	0.9278	0.0123	-	0.7312	0.0317	-
WL	0.9783	0	1.17×10^{-18}	0.9248	0	9.17×10^{-27}
KM	0.9520	0.0146	2.24×10^{-14}	0.8434	0.0353	1.86×10^{-30}
SC	0.9507	0.0087	3.62×10^{-18}	0.8002	0.0272	3.14×10^{-20}
OKKC	0.9473	0.0143	7.51×10^{-11}	0.8206	0.0333	1.32×10^{-24}

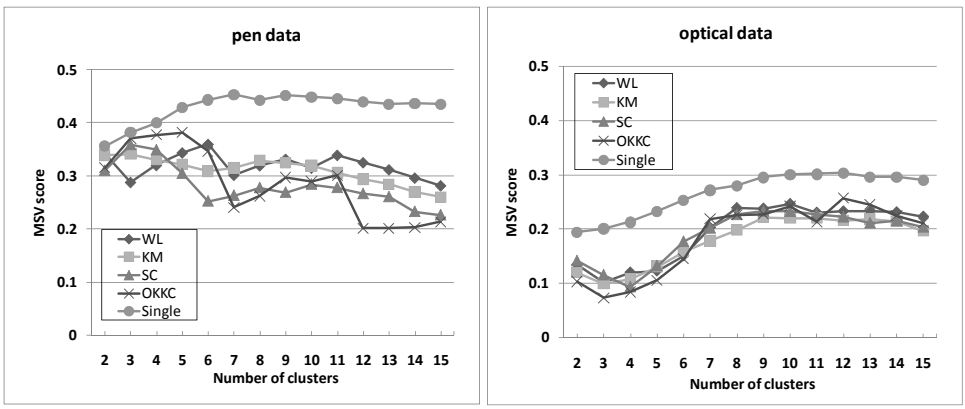


Figure 10.4: Evolution of MSV indices on digit recognition data sources with various number of clusters.

We apply paired t-test to evaluate the statistical significance of the data fusion performance. As shown in Table 10.1, the four data fusion approaches all significantly improve the performance over the best individual data.

The ground truth labels of data may not always be available in clustering problem. In a single data source, the model is often validated by internal validation indices such as Mean Silhouette Value (MSV), Modularity, and so on. To investigate whether internal validations are applicable on heterogeneous data sources, we plot the evolution of MSV with various cluster numbers on the two data sources, as shown in Figure 10.4. Comparing the two MSV curves, we make several statements about the issue of using internal validation indices in data fusion as follows:

- The scores obtained by internal validation cannot be compared across sources. This is because they are dependent to the scale and the structure of data, thus comparing indices with different scales and data structures is meaningless. For instance, in Figure 10.4 the MSV value of the optical data is always smaller than the pen data, which does not indicate that the clustering result on the optical data is worse than the one on the pen data.

- Within a single data source, good clustering partition obtained by data fusion not necessarily leads to better internal validation index. Most internal validation indices explain how good "the data" is fit to "the partition". When the data sources are integrated, "the data" is no longer any of the original data sources, but becomes the combined data (or the combined kernel). So, **in data fusion context, when observing a better internal validation score measured on a single data source, it is probably just a coincidence rather than an improvement.** In Figure 10.4, the MSV values of data fusion clustering are always lower than the one of single data clustering, however, we have clearly shown in Figure 10.3 that merging two data sources leads to more distinguishing patterns.

- Still, we are able to utilize the internal indices to judge the trend of clustering quality with different number of clusters. The optical data, as shown previously, has better clustering performance and the trend of the MSV curve indicates the correct optimal cluster number. Moreover, the four different data fusion algorithms show consistent trends on the optical data. On the contrary, the pen data is comparably less informative and the trends of algorithms are much different.

10.3.2 Disease gene clustering by multi-view text mining

We have investigated the biological problem of disease-causing gene identification in Chapter 8 and 9 as a prioritization approach. In prioritization, we assume that the similarity among genes causing the same disease is higher than the similarity with random genes. In the case of multiple diseases, the disease gene identification problem can also be regarded as a clustering problem. The assumption is extended as that the similarity of genes causing the same disease (within-disease-cluster similarity) is higher than the similarity of genes causing different diseases (between-disease-cluster similarity). Therefore, we expect these genes to demonstrate some "natural partitions" by the type of diseases. Thus, we extend the same benchmark data from the prioritization problem to the clustering problem.

The benchmark data set is identical to the one used in the prioritization problem as discussed in Chapter 8. To investigate the clustering performance, we enumerate all the pairwise combinations of the 29 diseases (resulting in 406 binary clustering tasks), as shown in Fig 10.5. All the 406 tasks are repeated 50 times (resulting in 20,300 runs of clustering per kernel). In each repetition, the average performance of all the 406 tasks is used as the performance indicator of this run. Then the mean value of the 50 repetitions is reported as final result.

We first investigate the clustering performance obtained by integrating multi-view text mining data constructed from the 9 controlled vocabularies (discussed in

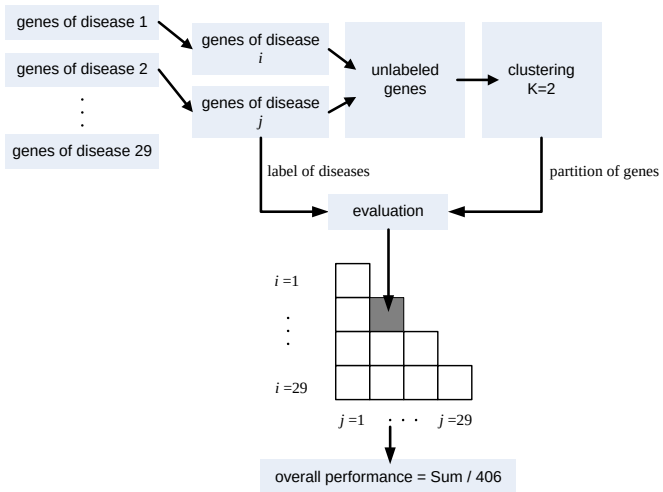


Figure 10.5: Conceptual scheme of clustering disease relevant genes

Chapter 8). Three integration methods (WL, OKKC, EACAL) obtain significant better performance than the best single view data (Table 10.2).

Table 10.2: Clustering performance by individual controlled vocabulary and multi-view approach

Individual CV	RI	NMI	Integration as multi-view(9 CV)	RI	NMI
LDDB	**0.7586** ± 0.0032	**0.5290**± 0.0032	Ward linkage	**0.8236** ± 0	**0.6015** ± 0
OMIM	0.7216 ± 0.0009	0.4606 ± 0.0028	OKKC(θ_{min}=0)	0.7624 ± 0.0067	0.5395 ± 0.0147
Uniprot	0.7130 ± 0.0013	0.4333 ± 0.0091	EACAL	0.7741 ± 0.0041	0.5542 ± 0.0068
eVOC	0.7015 ± 0.0043	0.4280 ± 0.0079	MCLA	0.7596 ± 0.0021	0.5268 ± 0.0087
MPO	0.7064 ± 0.0016	0.4301 ± 0.0049	QMI	0.7458 ± 0.0039	0.5084 ± 0.0063
MeSH	0.6673 ± 0.0055	0.3547 ± 0.0097	OKKC(θ_{min}=1/k)	0.7314 ± 0.0054	0.4723 ± 0.0097
SNOMED	0.6539 ± 0.0063	0.3259 ± 0.0096	AdacVote	0.7300 ± 0.0045	0.4093 ± 0.0100
GO	0.6525 ± 0.0063	0.3254 ± 0.0092	CSPA	0.7011 ± 0.0065	0.4479 ± 0.0097
KO	0.5900 ± 0.0014	0.1928 ± 0.0042	Complete linkage	0.6874 ± 0	0.3028 ± 0
			Average linkage	0.6722 ± 0	0.2590 ± 0
			HGPA	0.6245 ± 0.0035	0.3015 ± 0.0071
			Single linkage	0.5960 ± 0	0.1078 ± 0
			Merging all 9 CVs	0.6321 ± 0.0038	0.2830 ± 0.0079
			Merging 4 CVs (GO, eVOC, MeSH, SNOMED)	0.6333 ± 0.0053	0.2867 ± 0.0085
			Concept mapping(GO, eVOC, MeSH, SNOMED)	0.6241 ± 0.0056	0.2644 ± 0.0111
			novoc	0.5630	0.0892
			random	0.5103	0.0312

Table 10.3: Statistical analysis of the performance obtained on disease relevant genes clustering

	RI			NMI		
Best single data	0.7586	0.0032	-	0.5290	0.0032	-
WL	0.8236	0	7.97×10^{-47}	0.6015	0	6.45×10^{-46}
EACAL	0.7741	0.0041	3.39×10^{-31}	0.5542	0.0068	1.69×10^{-33}
OKKC	0.7624	0.0067	1.32×10^{-5}	0.5395	0.0147	1.89×10^{-5}

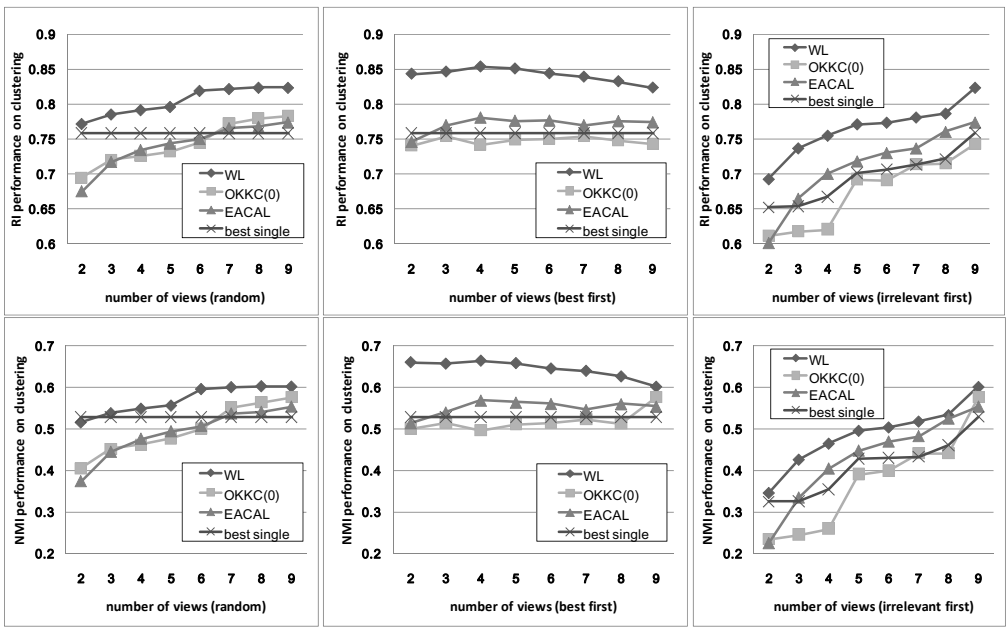


Figure 10.6: Multi-view clustering with various number of views

We apply paired t-test to evaluate the statistical significance of the data fusion performance. As shown in Table 10.3, the three presented data fusion approaches significantly improve the performance over the best individual data.

We then compare the performance of clustering integration on various number of views. Similar to the experiment in prioritization, we try three different approaches to combine various number of views. We present an example of clustering genes relevant to two diseases: breast cancer and muscular dystrophy. Each disease contains 24 relevant genes and there is no overlapping gene among them. We list the confusion tables and the mis-partitioned genes of each individual view in Table 10.4. As it is shown, all individual views produce several mis-partitioned genes. When these views are combined and clustered by the ward linkage method, all the genes are correctly partitioned by their disease labels.

Table 10.4: Clustering breast cancer and muscular dystrophy relevant genes by different CVs and the multi-view approach.

CV	Breast Cancer	Muscular Dystrophy	mis-partitioned genes
LDDB	22	2	RP11-49G10.8, FKTN
	0	24	
eVOC	23	1	RP11-49G10.8, FKTN
	7	17	LMNA, COL6A1, MYF6, CHEK2, SGCD, FKRP, DM
MPO	23	1	RP11-49G10.8
	1	23	SGCD
GO	23	1	RP11-49G10.8
	7	17	LMNA, COL6A1, MYF6, CHEK2, SGCD, FKRP, DM
MeSH	23	1	RP11-49G10.8
	2	22	SGCD,COL6A3
SNOMED	24	0	
	6	18	LMNA,COL6A1,MYF6,TRIM32,SGCD,DMD
OMIM	24	0	
	1	23	SGCD
Uniprot	24	0	
	4	20	MYF6,CHEK2,SGCD,FKRP
KO	19	5	SLC22A18, RP11-49G10.8, FKTN, PABPN1, CAPN3
	6	18	PPM1D, MYF6, SGCD, FKRP, COL6A3, DYSF
Multi-view	24	0	
(WL)	0	24	

10.4 Summary

In this chapter, we investigated *clustering by data fusion*, which is a new problem with the potential of many interesting and useful applications. We proposed a computational framework to combine heterogeneous data sources for clustering. This framework consists of two main approaches, ensemble clustering and kernel fusion. In the ensemble clustering approach we discussed some important existing algorithms and extended them to data fusion problems. In the kernel fusion approach, we created several new clustering algorithms in the Hilbert space. For KM clustering, we proposed the OKKC to weigh different data sources by learning the optimal kernel. For hierarchical clustering, we extended the classical linkage methods into kernel distance space and implemented four algorithms based on average combination of kernel matrices. For spectral clustering, we proposed the OKLC to combine kernels and Laplacians. Most of the algorithms in the framework are easy to implement, furthermore, the framework is also extendable to create novel and complex algorithms. For instance, a new weighted clustering method is developed on the basis of ANMI and it shown quite effective in journal set analysis [113].

We demonstrated the effectiveness of the proposed computational framework on two real applications, where 13 different data fusion algorithms were cross-compared. Using data fusion, the performance was significantly improved than the best individual data source.

Chapter 11

Combining lexical information and citation links in journal sets analysis

11.1 Introduction

In recent approaches of journal Scientometrics, to construct structural mappings of journal sets, people retrieve two different types of information (lexical similarities of journal-by-term data and citation-links of journal-by-journal data) and combine them in clustering analysis. The advantage of combining text-based similarity and citation-based link is first shown on the mapping of a small set of papers [27][28][67][86]. Later, the effectiveness of integration is demonstrated on a large scale data set containing thousands of papers retrieved from different journals [85][83][114]. By investigating documents in a broader range of science fields, the obtained structural mapping is shown useful to detect and monitor the emerging and converging fields of scientific research. In particular, to find groups of scientific journals that are similar in terms of lexical and citation information, we applied the OKKC algorithm to combine lexical similarities and citation-links to cluster the journals [114]. The clustering result of kernel integration is evaluated by two internal criteria (the MSV [148] and the modularity [134][133]) and two external criteria (the RI and the NMI). According to our experiments [114], when combining lexical data and citation data as kernels, the overall performance is restricted by the best individual kernel. In this chapter, we further improve the approach by combining the lexical similarities of journals as kernels and the citation-links as graph Laplacians. Then these multiple kernels and Laplacians are

combined by the OKLC algorithm and the clustering result is evaluated with the labels of ESI.

11.2 Background: hybrid clustering for journal set analysis

In scientometrics, information about journals is categorized at two different levels: the term similarities are measured at lexical level and the citations among journals are investigated in bibliometrics. Some bibliometric or scientometric research has been carried out on clustering or mapping of scientific publications. The widely used method of co-citation clustering was introduced independently by Small [157, 158] and [119]. Cross-citation based cluster analysis for science mapping has to be distinguished from the previous method; while the former one is usually based on links connecting individual documents, the latter approach requires aggregation of documents to units like journals or subject fields among which cross-citation links are established. Some advantages of this method (for instance, the possibility to analyze directed information flows) are contrasted by possible biases caused, for instance, by the use of predefined units (journals, subject categories etc.) implying already certain structural classification. Journal cross-citation clustering was used, among others by [109], [110], and [25], while [48] applied subject co-citation analysis to visualize the structure of science and its dynamics.

The integration of lexical similarities and citation links has also attracted lost of interests in other fields such as search engine design (i.e., Google combines text and link to calculate the similarities of its user's queries [31]). The combination of link-based clustering with a textual approach has also been suggested as early as around 1990 to improve efficiency and usability of co-citation and co-word analysis. The aim was to compensate the apparently low recall of co-citation analysis concerning current work and to allow contrasting the empirical characterisation of trend with immediacy by means of co-citation analysis ([27][28], [199]). The combination of link-based and textual methods also make it possible to cluster objects whenever links are weak or even missing (e.g. in the case of poorly cited or uncited papers). The present paper is based on a new approach of combined citation/lexical-based clustering [85], which forms a hybrid solution in two respects; on one hand, it combines the two traditional scientometric methods, as already mentioned, and it uses individual papers to cluster the journals in which those appeared. Furthermore, the lexical component is used here to label the obtained journal clusters.

Hybrid clustering has also been applied in various document analysis applications [124][73][184][20] as well as science mapping researches [67] [84] [114]. Though all these approaches fall in the paradigm to combine the lexical/ citation information,

the concrete algorithms applied therein are quite diverse. In the web document analysis, Modha & Spangler [124] integrate the similarity matrices measured in terms, out-links and in-links as a weighted linear combination. The data partition is obtained from the combined similarity matrix using a toric-k-means algorithm. He *et al.* [73] incorporate information of three different aspects (hyperlink, textual information and co-citation) to cluster the web documents using a graph-cut algorithm. Bickel & Scheffer (2004) investigate the web documents and combine their intrinsic views (page content) with their extrinsic views (anchor texts of inbound hyperlinks). Three different methods (generic EM clustering, k-means and agglomerative clustering) are applied to combine the views as hybrid clustering. Being distinguished from the web page analysis, Glenisson *et al.* [67] combine the textual analysis and the bibliometrics to improve the performance of journal publication clustering. Janssens *et al.* [84] proposes an un-biased combination of textual content and citation links on the basis of Fisher's inverse chi-square for the ward's linkage based agglomerative clustering.

11.3 Processing of journal data

11.3.1 Data sources and data processing

The raw experiment data contains more than six million published papers from 2002 to 2006 (i.e., articles, letters, notes, reviews, etc.) indexed in the Web of Science (WoS) database provided by Thomson Scientific. Citations received by these papers have been determined for a variable citation window beginning with the publication year, up to 2006, on the basis of an item-by-item procedure using special identification-keys made up of bibliographic data elements, which were extracted from first-author names, journal title, publication year, volume and first page. To resolve the ambiguities, the journals are checked for the name changes and the papers are also merged according to the changes. The journals not covered in the entire period (from 2002 to 2006) have been omitted. Two criteria are applied to select the journals for clustering: Firstly, only the journals that have at least 50 publications from 2002 to 2006 are investigated, others are removed from the data set. Secondly, only those that have more than 30 citations from 2002 to 2006 are kept, others are also removed. We selected 7 categories (1424 journals) as the journal set clustering data in this paper. The distributions of ESI labels of these 1424 journals are balanced because we want to avoid the affect of the skewed distributions in clustering evaluation.

11.3.2 Kernels of lexical information

Text mining analysis The titles, abstracts and keywords of the journal publications are indexed by a Jakarta Lucene based text mining program using no controlled vocabulary. The index contains 9,473,061 terms and we cut the Zipf curve of the indexed terms at the head and the tail to remove the rare terms, stopwords and common words. These words are known as usually irrelevant, also noisy for clustering purpose. After the Zipf cut, 669,860 meaningful terms are used to represent the journal in a vector space model where the terms are attributes and the weights are calculated as four weighting schemes: TF-IDF, IDF, TF and binary. The paper-by-term vectors are then aggregated to journal-by-term vectors as the representations of the lexical data. Therefore, we have obtained four sub-models as the textual data sources varied by the weighting scheme of terms.

11.3.3 Laplacians of citations

Cross-Citation

Two papers are defined as cross-citation if one cites another, where the value is equivalent to the frequency of citations. We ignore the direction of citations by symmetrizing the cross-citation matrix.

Binary Cross-Citation

To penalize the large amount of cross-citations in the journals with a lot of publications, we use binary values (1 or 0) to represent whether there is (or no) citation between two journals, termed as binary cross-citation.

Co-Citation

Two papers from different journals are defined as co-citation if they are cited simultaneously by a paper from the third journal. The co-citation frequency of the two papers equals to the number of other papers that cite them simultaneously.

Bibliographic coupling

Two papers from different journals are considered as bibliographic coupling when they cite the same paper from a third journal. The coupling frequency equals to the number of papers they simultaneously cite together. The citations among papers are also aggregated to journal level.

Thus, we obtain 8 different information sources about the journals. Four of them are text-mining based attribute data and we construct the corresponding kernel matrices using linear kernel function, denoted as $K_{tfidf}, K_{idf}, K_{tf}, K_{bn-text}$. Four citation sources represent relationships among journals as graphs and we construct corresponding normalized Laplacians, denoted as $\tilde{L}_{crosscit}, \tilde{L}_{biblio}, \tilde{L}_{cocit}, \tilde{L}_{bncit}$.

11.3.4 ESI labels

We refer the ESI categories created by Thomson Scientific as the ground-truth labels of journal assignment. The 7 ESI labels and the distributions of the selected journals are presented in Figure 11.1.

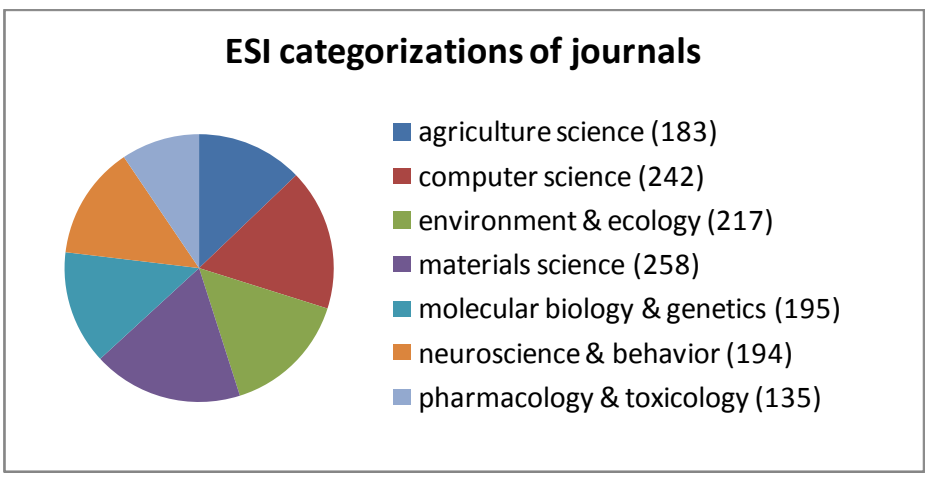


Figure 11.1: ESI categorizations and distributions of the 1424 journals selected for clustering

11.4 Experimental Setup

Kernel-Laplacian integration: The proposed OKLC algorithm is featured by the simultaneous integration of kernels and Laplacians. An alternative solution is to obtain the clustering partitions on kernels and Laplacians and ensemble their label information. To combine the labels, we apply a representative ensemble clustering algorithm, EACAL [59], which has been discussed in the previous Chapter. The clustering labels of the lexical data are produced by KM clustering and the labels of citation data are obtained by the SC. We also apply the KL

algorithm to combine a single kernel with a single Laplacian. The best 4 results out of 16 combinations are presented for comparison.

Integration as kernels: Four algorithms are applied to combine the eight sources as kernels for clustering. The KM is applied on the averaged kernel matrix. The OKKC is used to combine the kernel matrices in a weighted approach. The WL is adopted on the averagely combined kernel matrix. The EACAL is applied to ensemble the labels of the eight sources obtained by the kernel KM.

Integration as graphs: Analogously, we integrate all sources as graphs and cluster them by three algorithms: Normalized cut spectral clustering on averaged Laplacian (SC), Normalized cut on weighted combination of Laplacians (the gradient descent method to optimize the fusion of Laplacians), and the EACAL applied to ensemble the labels obtained by SC.

Individual source: To investigate whether the data fusion approach can bring advantage than the single data source, the eight adopted information sources are all partitioned by the KM and the SC individually as the baseline performance.

11.5 Results

We use the NMI to evaluate the clustering results of all approaches and the results are shown in Figure 10.2. Kernel-Laplacian integration performs significantly better than other comparing integration methods. Among them, we compare the OKLC with the best four KL results obtained on single kernel and Laplacian, the OKLC performs even better. The effect of the regularization parameter θ_{min} is also significant: the regularized integration (OKLC(0.5), 0.7954 ± 0.0245) performs the best, the average combination of kernels and Laplacians performs the second best (OKLC(1), 0.7762 ± 0.0320). We plot the optimal weights learned by these two versions in Figure 10.3. The best performance obtained on the single data source is the spectral clustering applied on the Term Frequency - Inverse Document Frequency (TF-IDF) Laplacian (SC, 0.7472 ± 0.0193). The complementary information contained in heterogeneous representations of lexical and citation data is helpful to make the data fusion clustering outperforming the best individual source. Collectively, among all the methods exhaustively compared on this problem, the proposed algorithm OKLC appears to be the best solution. For kernel based integration, the EACAL and the WL also perform good since their performance is better than the best individual kernel clustering result (green bars). The performance of the pure kernel fusion algorithm OKKC is similar to the best individual kernel, which is consistent with the results in our previous work [114]. This also demonstrates the significance of combining kernels with Laplacians in the present study. Another advantage of the OKLC is its ability to estimate the optimal cluster number. As shown in Figure 11.4, the eigenvalues are taken from

the generalized eigenvalue problem when the OKLC and KL algorithms converge. Though these eigenvalues are already affected by the pre-defined cluster number K because the algorithm requires K as the input to iterate, comparing the scales of eigenvalues obtained by different K values, one can still estimate the optimal cluster number.

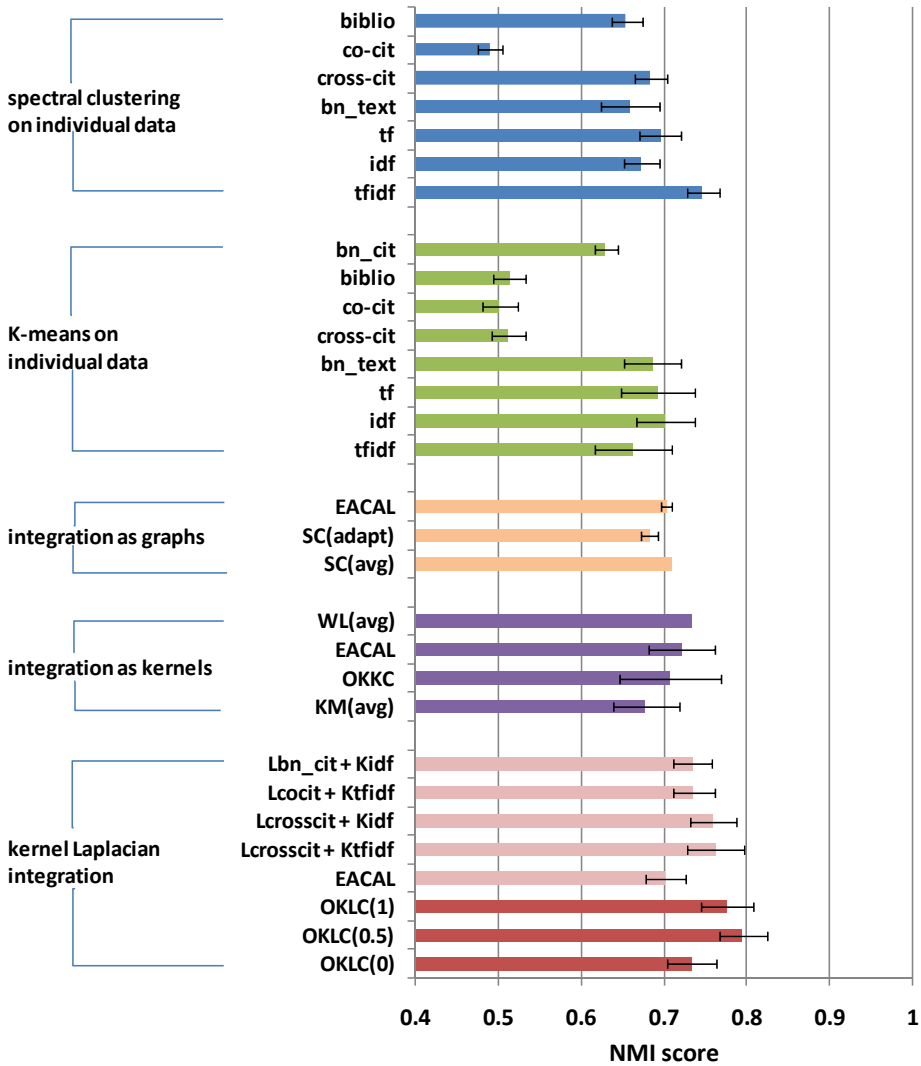


Figure 11.2: Comparison of the performance of the OKLC and other competing algorithms in journal set clustering. The error bar is the standard deviation of 20 random repetitions. For the regularized OKLC, the minimal boundary is set as 0.5, which means the minimal weight of each data source is 0.125. The regularization parameter λ in the OKLC is set as 10^{-4}.

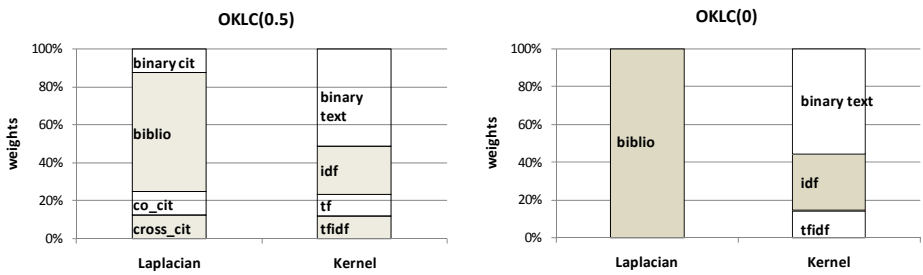


Figure 11.3: The optimal weights learned from the OKLC for kernels and laplacians integration. The figure on the left shows the weights obtained by the regularized integration when the minimal boundary is set as 0.5; The figure on the right shows the weights obtained by the non-regularized integration.

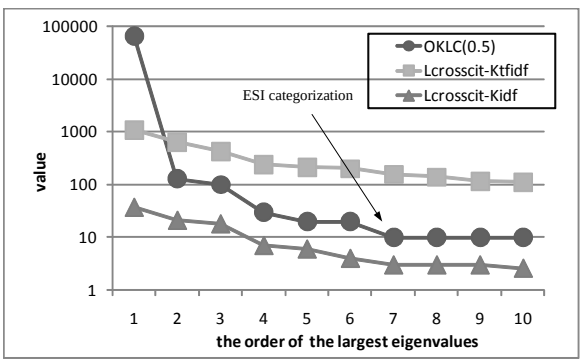


Figure 11.4: Comparison of the largest eigenvalues of the generalized eigenvalue problem in the OKLC(0.5) and the two best single KL algorithms. In spectral clustering, the largest eigenvalues are often used to indicate the optimal cluster number. This figure clearly shows that the elbow of the eigenvalues obtained from OKLC correctly indicates the same cluster number as the ESI categorizations. When combining the cross citation Laplacian with the IDF kernel, the single KL algorithm also indicates the correct number. In another case (combining the cross citation Laplacian with the TFIDF kernel), the eigenvalues of single KL algorithm is not very informative for optimal cluster number prediction.

11.6 OKLC-*light*

The OKLC algorithm optimizes the Rayleigh quotient objective by solving a complex problem consists of three coupled optimization procedures. To investigate whether it can be simplified, we propose another algorithm to combine kernels and Laplacians in a 2-layer alternative minimization framework. We use a different notation of Laplacian, $\hat{L}$, given by

$$\hat{L} = D^{-1/2} W D^{-1/2}, \tag{11.1}$$

where D is a diagonal matrix where the entry at (i, i) position is the sum of the entries of row i in the affinity matrix W. Based on this definition, the objective of the spectral clustering is equivalent to maximizing the term trace$(P^T \hat{L} P)$. Therefore, the objective of the OKLC in (6.7) can be rewritten as

$$\begin{aligned} \max_{A} \text{trace} \quad & \left(A^T \hat{L} A + A^T X^{\Phi T} X^{\Phi} A \right) \\ \text{s.t.} \quad & A^T A = I_k. \end{aligned} \tag{11.2}$$

To optimize this new objective, we may directly apply the spectral version of OKKC algorithm to combine the kernels and the new Laplacians $\hat{L}$ in a 2-layer alternative minimization framework. The novelty is that two different types of matrices are combined additively. We denote the new algorithm based on this new objective as OKLC-*light*.

We apply the OKLC-*light* on the identical journal data used in the previous section. The comparison of the OKLC and the OKLC-*light* on three different coefficient regularization parameters is listed in Table 11.1. Interestingly, the performance of OKLC-*light* is also quite good, in particular, it outperforms OKLC in unregularized integration ($\theta_{min} = 0$). It only performs slightly worse in the case of $\theta_{min} = 0.5$ but the difference is really small. When combing the kernels and the Laplacians averagely ($\theta_{min} = 1$), their performance is almost the same.

Table 11.1: Comparison of the OKLC and the OKLC-*light* on journal set clustering

	OKLC	OKLC-*light*
$\theta_{min} = 0$	0.7331 ± 0.0294	$\mathbf{0.7637} \pm 0.036$
$\theta_{min} = 0.5$	$\mathbf{0.7954} \pm 0.0285$	0.7734 ± 0.0367
$\theta_{min} = 1$	0.7762 ± 0.032	0.7726 ± 0.0156

We also plot the optimal coefficients of kernels and Laplacians learned by OKLC-*light* in Figure 11.5. In conclusion, both OKKC and OKLC-*light* obtain significant improvements over the single data source. In the present problem, the effect of the algorithmic structure is rather small (ration or sum); the most significant effect comes from the combination of kernels and Laplacians for clustering.

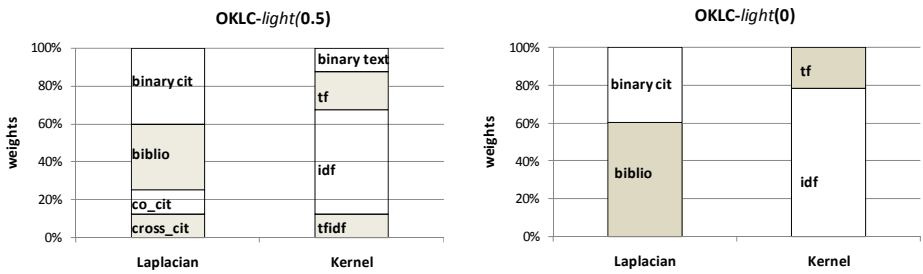


Figure 11.5: The optimal weights learned by the OKLC-*light*. The figure on the left shows the weights obtained by the regularized integration when the minimal boundary is set as 0.5; The figure on the right shows the weights obtained by the non-regularized integration.

11.7 Summary

In this Chapter we illustrated a scientometrics application to combine multiple lexical data based kernels and citation based Laplacians for journal sets clustering. The effectiveness of the kernel-Laplacian clustering was clearly demonstrated. We tried two different versions of OKLC and obtained similar performance. The OKLC-*light* is much simpler than the OKLC and their performance only has 1% difference so it is more preferable than the OKLC in the present problem. Moreover, the interpretations based on the coefficients learned by the two algorithms are much different. Therefore in our opinion, in complicated statistical learning problem, over-concerning about the interpretations is not meaningful because two disparate interpretations may only result in a very small difference in performance. The presented application also inspires us to consider different types of data representations in learning. Kernels are useful, but may not work for all the purposes. To model the problem, a unified objective is desirable, however, when the performance is not good, one may need to consider treating data sources separately as different representations or via multi-objective optimization. The proposed algorithm OKKC and its simplified version OKLC-*light* can be applied to a broad type of applications. For example, in genome data analysis, to combine the interaction data (Protein Protein Interaction (PPI) networks) with the attribute data (microarray expression, annotation data, etc.) for clustering would be a quite novel and interesting approach.

Chapter 12

Conclusions and Future Research

12.1 Conclusions

The exquisite nature of combining various senses in human cognition motivated our approach to incorporate multiple sources in data mining. The research described in this thesis covers a number of topics which are relevant to supervised and unsupervised learning by kernel based data fusion. The discussion of these topics were distinguished in four different aspects: theory, algorithm, application and software.

In the theoretical part, we reviewed the mathematical objectives of several popular supervised and unsupervised techniques in machine learning. The Rayleigh quotient objective was shown as equivalent as the objective of the kernel fusion method based on LS-SVM MKL. The equivalence of the objectives enabled us to plug the LS-SVM MKL in unsupervised learning problems to develop various new algorithms. To tackle the computational burden of large scale problems, we investigated several optimization techniques to simplify the problem. One of the main findings was that the SIP formulation of LS-SVM MKL is very efficient, being much faster than, while showing identical solution as other formulations on the benchmark data.

In the algorithmic part, we presented three kernel fusion algorithms. The first two were proposed for clustering analysis: the OKKC method combined multiple kernels for clustering, which was suitable for attribute-base data integration; the OKLC method combined kernels and Laplacian in clustering, which was proposed

for fusing attribute-based data and interaction-based data. The third algorithm, WMKCCA, extended the conventional unweighted canonical correlation method to a weighted version. We noticed that the CCA problem is shown related to Linear Discriminant Analysis (LDA) and their kernel versions are also connected [45], so it may be possible to have a MKL interpretation for CCA. In our discussion we rather focused the effect of dimensionality reduction in CCA and presented some preliminary results to classify the data in the weighted canonical spaces.

We applied our proposed algorithms in several real bioinformatics and scientometrics applications. We applied the 1-SVM MKL algorithm to combine multiple text mining views and multiple genomic data sources in disease candidate gene prioritization. We also proposed a framework, including the OKKC, the OKLC, and a set of other popular algorithms, to combine heterogeneous data sources in clustering analysis. This clustering framework was applied to partition disease relevant genes using multi-view text mining data. It was also applied in a scientometrics application to combine text mining and bibliometric data to obtain mapping fields of science and technology.

Finally, we solidified our efforts in kernel based gene prioritization as the Endeavour MerKator software. MerKator is a real kernel based software designed to prioritize candidate genes in terms of combining evidence from multiple species and multiple genomic data sources. It handles large amount of data at the full genome scale. Many challenges were encountered in the design and development stage of MerKator and some of our strategies are presented in this thesis. The software will be freely accessible online and it will play a useful role for biologists and bioinformaticians in their researches.

12.2 Future research

The research presented in this thesis opens the doors to interesting perspectives in several directions.

12.2.1 Kernel based sensor fusion

Data fusion approaches enable us to inference the structure of a system model with multiple signal sources extracted from heterogeneous measurements or distributed sensors. Apart from the bioinformatics and the scientometrics applications already mentioned in this thesis, it also has a wide range of applications in many other scientific and technological fields, for example, in digital communication, speech and acoustic signal processing, as well as in system identification and control. The multi-sensor approach is motivated as to acquire a system model with better precisions and robustness, for example, in robot control, the ultimate goal is

to develop robots able to do a wide variety of tasks in unconstrained settings, with a degree of robustness and adaptability similar to that of humans. Robots typically use different sensor modalities (vision, audio, range sensors, etc.) to capture information from the environment. It is clear that different sensors provide different kinds of information and no sensor works perfectly in all real-world applications. How to effectively utilise the positive side of each sensor and avoid its negative side becomes critical for the deployment of mobile robots in the real world. To reach this goal, sensor technology and data fusion algorithms have been a hot research topic and played a key role in the acquisition of more accurate and reliable information for the last two decades. However, several big challenges still remain such as how to cope with non-linear and non-Gaussian situations, how to create adaptive multi-sensor data fusion algorithms in an uncertain and dynamically changing environment, how to characterize the uncertainties in sensor measurements, and so on [80]. Recently, SVM-based control scheme has been applied to control robot in noisy environment [33][87]. Since kernel methods appear as principled approaches to deal with nonlinearity, the notion of kernel fusion is also well grounded in theory and able to accommodate the variety of the sensor data structure and the intrinsic dynamic of the problem, moreover, the efficient LS-SVM MKL algorithm enable us to handle large amount of data that is received continuously, therefore, the methods introduced in this thesis may be applicable to develop a new paradigm of sensor fusion based control and system identification.

12.2.2 Integration of expression data and interaction networks for disease genes identification

We presented an application of combining attribute-based lexical data and interaction-based citation networks in journal set mapping. The improvement was shown significantly by comparing our results in two publications. The improvement also could inspire us to apply the analogue strategy to combine microarray expression data and PPI networks in bioinformatics applications. The OKLC algorithm proposed in the thesis may technically solve the clustering problem, thus the main difficulty is to collect high quality data sources and to find good benchmark data for evaluation. The prioritization problem on graph or network data is more challenging because of the two reasons. Firstly, there is few established method for graph or network ranking. In literature, only two relevant works address this issue: the Laplacian smoothing method proposed by Agarwal [5] and the enhanced page rank method introduced by Agarwal and Chakrabarti [4]. To create novel graph ranking algorithms using other technique is also possible, such as the label propagation [189]. The second challenge is the integration of kernels and graphs. Though in Figure 1.3 we have depicted the conceptual roadmap to combine disparate representations of genomic data, in some cases to combine kernels with graphs directly may not preferable because the sparse

graphs are more likely to be overwhelmed by the dense kernels. To tackle this issue, one may need to prioritize them separately and combine the ranking scores using ensemble methods introduced in Chapter 8.

12.2.3 A joint framework of data fusion with feature selection

Feature selection is an important technique to select a small number of important features from a vast amount of irrelevant features to improve the accuracy of prediction or classification. Unfortunately, kernel method do not perform feature selection automatically because it uses quadratic regularization. To equip SVM with feature selection, typical approaches include ad-hoc methods such as *recursive feature elimination* or L1 regularization such as *Kernelized Lasso*. To combine the kernel based feature selection with the kernel based data fusion is a direction worth further investigation. Nevertheless, for a specific learning problem, one need to be aware about the objectives of feature selection and kernel fusion because in some cases, the problem can be much simplified. For example, suppose the objective function of feature selection is to minimize the function f and the objective of data fusion is to minimize the function g. Denote X_i as the i-th data set, thus the goal of feature selection is to find a subset data X_i' to minimize $f(X_i')$. The goal of data fusion is to find a linear parametric combination of X_i, denoted as $\sum_{i=1}^{p} \theta_i X_i$ and $\theta_i \geq 0$ to minimize $g(\theta_i, X_i)$. The joint objective is then given by

$$J = \min\{g(\theta_1, X_1') + g(\theta_2, X_2') + ... + g(\theta_p, X_p')\}. \tag{12.1}$$

Actually, the data fusion problem in the objective as (12.1) can be factored out, if the following conditions are satisfied:

$$\min g(\theta_i, X_i) = \min \theta_i g(X_i), \tag{12.2}$$

$$\min g(X_i) = \min f(X_i). \tag{12.3}$$

These two conditions are actually quite common. The condition in (12.2) states that the linear coefficients of the data sources can be factored out of the objective, which is satisfied by all the L_∞-norm MKL problems presented in this thesis. The condition in (12.3) implies that the objective of feature selection and data fusion is equivalent, which is often true in the problems such as disriminant analysis, regression analysis, and so on. Then, the complicated problem in (12.1) can be equivalently rewritten as

$$J = \min\{\theta_1 f(X_1') + \theta_2 f(X_2') + ... + \theta_p f(X_p')\}, \tag{12.4}$$

which is basically an integration of two independent steps: to select features on each individual data source, next, to combine the feature selection results as data fusion. We also notice that in some real problems, the complicated approaches

may not necessarily lead to better performance. In contrast, simple method works quite well. For example, in the prioritization application presented in Chapter 8, the feature selection (Latent Semantic Indexing (LSI)) and the objective of kernel fusion 1-SVM are unrelated. Though the combination of inequivalent objectives seems to be questionable [1], it works quite well and it significantly improves the performance. In the contrary, a counter example might refer to the NAML algorithm proposed for clustering analysis. NAML combines LDA based feature selection, kernel fusion and clustering in a triple-level optimization procedure. All these three components are optimized towards the same objective function. However, according our benchmark results (not shown in the thesis) it does not perform well in combining heterogeneous data sources.

12.2.4 Other relevant topics

The MKL approach is also possible to advance in many further directions. For instance, throughout this paper the kernels are all combined as a linear additive model. The linear model can be further enriched by nonlinear operations, such as the multiplication of kernels, the inverse of kernels, etc., and leads to new interpretations in data fusion. Another relevant problem is to incorporate prior knowledge with multiple kernel learning. In some approaches, the prior knowledge is represented as additional constraints in the objective, whereas some other approaches embed prior knowledge in semi-supervised learning. How to involve prior knowledge in kernel methods is still an ongoing problem.

12.3 Closing remarks

The work presented in this thesis was done in the Bioinformatics group of SCD, ESAT, K.U.Leuven, where the use of statistical modeling is distinguished as two cultures [29] which strongly influence and compete with each other. It was a stimulating and interesting experience for me to feel the collisions. The topics discussed in this thesis apparently belong to the algorithmic modeling culture (the opposite one is the data modeling culture, named by Leo Breiman [29]). All the effort in this thesis starts with an algorithmic objective; there is few hypothesis and assumption about the data; the generalization from training data to test data relies on the *i.i.d.* assumption in machine learning. We consider the data being generated by a complex and unknown black box with an input $\vec{x}$ and an output $\vec{y}$. Our goal is then to find a function $f(\vec{x})$ – an algorithm that operates on $\vec{x}$ to predict the response $\vec{y}$. The blackbox is then validated and adjusted in terms of the predictive accuracy.

[1] It is yet an unsolved problem to perform feature selection for ranking

Bioinformatics is an active field full of the collisions between the two cultures. On one hand, some approaches spend much effort on making assumptions about the phenomena under study [40]. The parameters are estimated and the conclusions are made fully on the assumed model of the data. However, Terabytes of data are generated by high-throughput techniques at an awesome rate; In data fusion these terabytes are multiplied by the number of sources and the number of species; In this case, it seems not easy to find a data model to explain this mixture. On the other hand, there is an emphasis on assuming less and less via generalized additive models [40]. Admittedly, the excellent performance of these algorithmic approaches may attribute to the intrinsic complexity of their structures, for instance, the nonlinear transfer functions of neurons, the backpropagation architecture, the primal-dual trick of nonlinearity, and so on. One may also argue that the interpretability of these methods is poor and the interpretation is exactly the essence of biological and medical data analysis. However, as we have shown, for a complex problem two disparate models may result in only 1% difference on the performance. So interpretation is not always important and reliable for complicated problems.

The topics discussed in this thesis mostly come from the heated area in machine learning. This is a field where things change rapidly and new methods always look better than the old ones. The Neural network works better than the logistic regression, the SVM works better than the neural network, so in a near future new methods may also outperform SVM. Despite of the rapid innovation of algorithms, the true merit lies in the fact that we obtain more accurate predictions on more complicated problems, which may finally lead us to the ultimate goal of computational intelligence. To end this thesis, I would like to quote the statement, published by Leo Breiman nine years ago, about the three important changes in the advances of statistical modeling [29]:

- Roshomon: Good prediction may be ascribed to multiple models while each model tells a different story. One should focus on the performance of the prediction rather than over concerning the interpretation of a specific model.
- Occam: Make good prediction first, then understand why, and try to simplify the model.
- Bellman: The curse of dimensionality is now ubiquitous, embrace it and enjoy its good side.

Appendix A

List of algorithms

The list of main algorithms developed to this thesis.

no.	Algorithm	Purpose	Chapter index
1	Linf-norm 1-SVM MKL	ranking	3
2	L2-norm 1-SVM MKL	ranking	3
3	Linf-norm Vapnik's SVM MKL (QP version)	classification	3
4	L2-norm Vapnik's SVM MKL (QP version)	classification	3
5	Linf-norm LS-SVM MKL (QP version)	classification	3
6	L2-norm LS-SVM MKL (QP version)	classification	3
7	Linf-norm LS-SVM MKL (QP version, λ estimation)	classification	3
8	L2-norm LS-SVM MKL (QP version, λ estimation)	classification	3
9	Linf-norm Vapnik's SVM MKL (SIP version)	classification	4
10	L2-norm Vapnik's SVM MKL (SIP version)	classification	4
11	Linf-norm LS-SVM MKL (SIP version)	classification	4
12	L2-norm LS-SVM MKL (SIP version)	classification	4
13	Linf-norm LS-SVM MKL (SIP version, λ estimation)	classification	4
14	L2-norm LS-SVM MKL (SIP version, λ estimation)	classification	4
15	OKKC (Linf, QP version)	clustering	5
16	OKKC (Linf, QP version, λ estimation)	clustering	5
17	OKKC (Linf, SIP version)	clustering	5
18	OKKC (Linf, SIP version, λ estimation)	clustering	5
19	OKKC (L2, QP version)	clustering	5
20	OKKC (L2, QP version, λ estimation)	clustering	5
21	OKKC (L2, SIP version)	clustering	5
22	OKKC (L2, SIP version, λ estimation)	clustering	5
23	NAML (Linf)	clustering	5
24	NAML (L2)	clustering	5
25	OKLC (3-level, Linf)	clustering	6
26	OKLC (3-level, Linf, λ estimation)	clustering	6
27	OKLC (3-level, L2)	clustering	6
28	OKLC (3-level, L2, λ estimation)	clustering	6

no.	Algorithm	Purpose	Chapter index
29	OKLC (2-level, Linf)	clustering	11
30	OKLC (2-level, Linf, λ estimation)	clustering	11
31	OKLC (2-level, L2)	clustering	11
32	OKLC (2-level, L2, λ estimation)	clustering	11
33	WMKCCA	canonical correlation	7
34	EACAL	clustering	10
35	AdacVote	clustering	10
36	QMI	clustering	10
37	SL,AL,CL,WL	clustering	10
38	SC	clustering	10

The Matlab code of all algorithms is available online at

`http://homes.esat.kuleuven.be/~syu/thesis.html`

Appendix B

List of applications and applied algorithms

Applications discussed in this thesis and the applied algorithms listed in Appendix A.

no.	Application	Algorithms in Appendix A	Chapter index
1	multi-view disease gene prioritization	1,2	3,8
2	multi-view disease gene clustering	15,34,35,36,37,38	10
3	clinical data integration	3,4,5,6,7,8	3
4	numerical experiments	1-14	4
5	cross-species gene prioritization	1	9
6	clustering by digit recognition data fusion	15,34,35,36,37,38	10
7	journal set data hybrid clustering	15,25,34,37,38	11

Bibliography

[1] E.A. Adie, R.R. Adams, K.L. Evans, D.J. Porteous, and B.S. Pickard. Speeding disease gene discovery by sequence based candidate prioritization. *BMC Bioinformatics*, 6(55), 2005.

[2] E.A. Adie, R.R. Adams, K.L. Evans, D.J. Porteous, and B.S. Pickard. Suspects: enabling fast and effective prioritization of positional candidates. *Bioinformatics*, 22:773–774, 2006.

[3] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.C. Tranchevent, B. De Moor, Marynen. P., B. Hassan, P. Carmeliet, and Y. Moreau. Gene prioritization through genomic data fusion. *Nature Biotechnology*, 24(5):537–544, 2006.

[4] A. Agarwal and S. Chakrabarti. Learning random walks to rank nodes in graphs. *in Proc. of the ICML'07*, 2007.

[5] S. Agarwal. Ranking on graph data. *in Proc. of the ICML'06*, 2006.

[6] S. Akaho. A kernel method for canonical correlation analysis. *in Proc. of the International Meeting of Psychometric Society 2001*, 2001.

[7] C. Alzate. *Support Vector Methods for Unsupervised Learning*. PhD thesis, K.U.Leuven, 2009.

[8] E.D. Andersen and K.D. Andersen. The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. *High Perf. Optimization*, pages 197–232, 2000.

[9] S. Asur, D. Ucar, and S. Parthasarathy. An ensemble framework for clustering proteinprotein interaction networks. *Bioinformatics*, 23(13):i29–i40, 2007.

[10] H.G. Ayad and M.S. Kamel. Cumulative voting consensus method for partitions with a variable number of clusters. *IEEE Trans. PAMI*, 30(1):160–173.

[11] S Aymè. Bridging the gap between molecular genetics and metabolic medicine: access to genetic information. *European Journal of Pediatrics*, 159(15):S183–S185, 2000.

[12] F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Dept. of Statistics, University of California, 2005.

[13] F.R. Bach and M.I. Jordan. Kernel independent component analysis. *Journal of Machine Research*, 3:1–48, 2003.

[14] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *in Proc. of 21st international conference on Machine Learning*, 2004.

[15] D.J. Barry. *Design Of and Studies With a Novel One Meter Multi-Element Spectroscopic Telescope*. PhD thesis, Georgia State University, 1995.

[16] S. Becker. Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural System*, 7:7–31, 1996.

[17] Y. Bengio, O. Delalleau, N. Le Roux, J.F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel pca. *Neural Computation*, 16(10):2197ÍC2219.

[18] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. What is the nearest neighbor in high dimensional spaces? *in Proceeding of 17-th Int'l Conf. Database Theory (ICDT'99)*, 1540:217–235, 1999.

[19] R. Bhatia. *Matrix Analysis*. Springer-Verlag, New York, 1997.

[20] Steffen Bickel and Tobias Scheffer. Multi-View Clustering. *Proceeding of IEEE data mining Conference*, 2004.

[21] C. M. Bishop. *Pattern recognition and machine learning*. Springer Press, 2006.

[22] C. Blake and C. Merz. Uci repository of machine learning database, 1998.

[23] O. Bodenreider. Lexical, terminological, and ontological resources for biological text mining. *Text mining for biology and biomedicine*, 2006.

[24] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Computational Learning Theory*, pages 144–152, 2002.

[25] K.W. Boyack, K. Börner, and R. Klavans. Mapping the structure and evolution of chemistry research. *Scientometrics*, 79(1):45–60, 2009.

[26] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[27] R. R. Braam, H. F. Moed, and A.F.J. Van Raan. Mapping of science by combined cocitation and word analysis, part 1: Structural aspects. *Jounral of the American Society of Information Science*, 42(4):233–251, 1991.

[28] R. R. Braam, H. F. Moed, and A.F.J. Van Raan. Mapping of science by combined cocitation and word analysis, part ii: Dynamical aspects. *Jounral of the American Society of Information Science*, 42(4):252–266, 1991.

[29] L. Breiman. Random forests. *Machine Learning*, 2001.

[30] L. Brieman. Bagging predictors. *Machine Learning*, 1996.

[31] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-1):207–117, 1998.

[32] J. Chen, Z. Zhao, J.P. Ye, and H. Liu. Nonlinear adaptive distance metric learning for clustering. *in Proc. of ACM SIGKDD 07*, 2007.

[33] J. S. Chiou and K. Y. Wang. Application of a hybrid controller to a mobile robot. *Simulation Modeling Pratice and Theory*, 16(7):783–795, 2008.

[34] H. W. Chun. Extraction of gene-disease relations from MEDLINE using domain dictionaries and machine learning. *in Proceeding of ACM KDD*, 2007.

[35] Widen Clinic. http://www.widenclinic.com/health.html.

[36] T. J. Cohen, T. Barrientos, Z. C. Hartman, S. M. Garvey, G. A. Cox, and T. P. Yao. The deacetylase HDAC4 controls myocyte enchancing factor-2-dependent structural gene expression in response to neural activity. *The FASEB Journal*, 23:99–106, 2009.

[37] Confucius. The analects. http://classics.mit.edu/Confucius/analects.html, 500 B.C.

[38] Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[39] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*.

[40] D.R. Cox. Comment to "statistical modeling: The two cultures". *Statistical Science*, 16:199–231, 2001.

[41] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[42] I. Csiszar and G. Tusnady. Information geometry and alternating minimization procedures. *Statistics and Decisions*, Suppl. 1:205–237, 1984.

[43] A. Daemen, O. Gevaert, F. Ojeda, A. Debucquoy, Sempoux C. Suykens, J. A. K., J. P. Machiels, K. Haustermans, and B. De Moor. A kernel-based integration of genome-wide data for clinical decision support. *Genome Medicine*, 1:39, 2009.

[44] D. Dash and M. J. Druzdzel. Robust independence testing for constraint-based learning of causal structure. *in Proceedings of the 19th conference on Uncertainty in Artificial Intelligence*, pages 167–174, 2003.

[45] T. De Bie. *Semi-supervised Learning based on Kernel methods and Graph cut algorithms*. PhD thesis, K.U.Leuven, 2005.

[46] T. De Bie, L.C. Tranchevent, L. Van Oeffelen, and Y. Moreau. Kernel-based data fusion for gene prioritization. *Bioinformatics*, 23(13):i125–i132, 2007.

[47] G. De Mars, A. Windelinckx, G. Beunen, G. Delecluse, J. Lefevre, and M. A. Thomis. Polymorphisms in the CNTF and CNTF receptor genes are associated with muscle strength in men and women. *J. Appl. Physiol.*, 102(5):1824–1831, 2007.

[48] F. De Moya-Anegón, B. Vargas-Quesada, Z. Chinchilla Rodriguez, E. Corera-Álvarez, F. J. Munoz Fernández, and V. Herrero-Solana. Visualizing the marrow science. *Jounral of the American Society of Information Science*, 58(14):2167–2179, 2007.

[49] L. De Raedt and L. Dehaspe. Clausal discovery. *Machine Learning*.

[50] L. De Raedt and W. Van Laer. Inductive constraint logic. *Lecture Notes in Computer Science*.

[51] I.S. Dhillon, Y.Q. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. *in Proceeding of ACM KDD 04*, 2004.

[52] C. Ding and X. He. K-means clustering via principal component analysis. *in Proc. of ICML 2004*, 2004.

[53] C. Ding and X. He. Linearized cluster assignment via spectral ordering. *in Proc. of ICML 2004*, 2004.

[54] H. Drucker, R. Schapire, and P. Simard. Improving performance in neural networks using a boosting algorithm. *Advances in Neural Information Processing Systems*, 1993.

[55] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification (2nd Edition)*. John Wiley & Sons Inc., 2001.

[56] D B Emmert, P J Stoehr, G Stoesser, and G N Cameron. The european bioinformatics institute (ebi) databases. *Nucleic Acids Research*, 26(1):3445–3449, 1994.

[57] M Escarceller, R Pluvinet, L Sumoy, and X Estivill. Identification and expression analysis of C3orf1, a novel human gene homologous to the Drosophila RP140-upstream gene. *DNA Seq*, 11(3-4):335–338, 2000.

[58] L. Franke, H.V. Bakel, L. Fokkens, E.D. de Jong, M. Egmont-Petersen, and C. Wijmenga. Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am J Hum Genet*, 78(6):1011–1025, 2006.

[59] A.L.N. Fred and A.K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. PAMI*, 27(6):835–850.

[60] Y. Freund and R. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[61] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. *in Proceedings of the 14th ICML*, pages 125–133, 1997.

[62] E. Fromont, R. Quiniou, and M.-O. Cordier. Learning rules from multisource data for cardiac monitoring. *Lecture notes of computer science, AIME 2005*, 2005.

[63] Mohlke K.L. Gaulton, K.J. and T. Vision. A computational system to select candidate genes for complex human traits. *Bioinformatics*, 23(9):1132–1140, 2007.

[64] Gevaert08. *A Bayesian network integration framework for modeling biomedical data*. PhD thesis, Katholieke Universiteit Leuven, 2008.

[65] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Trans Neural Networks*, 13(3):780–784, 2002.

[66] P. Glenisson, B. Coessens, S. Van Vooren, J. Mathys, Y. Moreau, and B. De Moor. Txtgate: profiling gene groups with text-based information. *Genome Biology*, 5(6):R43, 2004.

[67] P. Glenisson, W. Glänzel, F. Janssens, and B. De Moor. Combining full text and bibliometric information in mapping scientific disciplines. *Information Processing and Management*, 41(6):1548–1572.

[68] W. Glenisson, V. Castronovo, and D. Waltregny. Histone deacetylase 4 is required for TGFbeta1-induced myofibroblastic differentiation. *Biochim Biophys Acta.*, 1773(10):1572–1582, 2007.

[69] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Scholkopf. Kernel methods for measuring independence. *Journal of Machine Research*, 6:2075–2129, 2005.

[70] L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Computer-Aided Design*, 11(9):1074–1085, 1992.

[71] D. R. Hardoon and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[72] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Press, New York, 2001.

[73] X. He, H. Zha, C. H. Q. Ding, and H. D. Simon. Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis*, (1):19–45, 2002.

[74] R. Hettich and K.O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35, 1993.

[75] L. Holmstrom and F. Hoti. Application of semiparametric density estimation to classification. *in Proc. of ICPR'04*, pages 371–374, 2004.

[76] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1986.

[77] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

[78] Jodi House. www.cycrout.com/jillianpwinslow/brain.jpg.

[79] C.M. Hsu and M.S. Chen. On the design and applicability of distance functions in high-dimensional data space. *IEEE Trans. on Knolwedge and Data Engineering*, 21(4):523–536, 2009.

[80] H. Hu and J. Q. Gan. Sensors and data fusion algorithms in mobile robotics. *Technical Report: CSM-422, Department of Computer Science University of Essex, ISSN 1744 - 8050*, 2005.

[81] J.E. Hutz, A.T. Kraja, H.L. McLeod, and M.A. Province. Candid: A flexible method for prioritizing candidate genes for complex human traits. *Genetic Epidemiology*, 32:779–790.

[82] A.K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice-Hall College Div, 1988.

[83] F. Janssens, W. Glänzel, and B. De Moor. Hybrid clustering for validation and improvement of subject-classification schemes. *Information Processing and Management*, to appear.

[84] F. Janssens, W. Glänzel, and Bart De Moor. Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis. *in Proc. of 13th ACM SIGKDD*, 2007.

[85] F. Janssens, W. Glänzel, and Bart De Moor. A hybrid mapping of information science. *Scientometrics*, 75(3):607–631, 2008.

[86] F. Janssens, J. Leta, W. Glänzel, and B. De Moor. Towards mapping library and information science. *Information Processing and Management, special Issue on Informatics*, 42(6):1614–1642.

[87] L. Jiang, M. Deng, and A. Inoue. Support vector machine-based two-wheeled mobile robot motion control in a noisy environment. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 222(7):733–743, 2008.

[88] Antonio Jimeno, Ernesto Jimenez-Ruiz, Vivian Lee, Sylvain Gaudan, Rafael Berlanga, and Dietrich Rebholz-Schuhmann. Assessment of disease named entity recognition on a corpus of annotated sentences. *BMC Bioinformatics*, 9(S3), 2008.

[89] J. Kaliski, D. Haglin, C. Roos, and T. Terlaky. Logarithmic barrier decomposition methods for semi-infinite programming. *International Transactions in Operations Research*, 4:285ÍC303, 1997.

[90] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The kegg databases at genomenet. *Nucleic Acids Res.*, 30(1):42–46, 2002.

[91] R. Kannan, S. Vempala, and A. Vetta. On clusterings - good, bad, and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

[92] J Kelso, J Visagie, G Theiler, A Christoffels, S Bardien-Kruger, D Smedley, D Otgaar, G Greyling, V Jongeneel, M McCarthy, T Hide, and W Hide. evoc: A controlled vocabulary for gene expression data. *Genome Research*, 13:1222–1230, 2003.

[93] J.R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58:433–451.

[94] H.C. Kim, S. Pang, H.M. Je, and S.Y. Bang. Pattern classification using support vector machine ensemble. *in Proc. of 16th International Conference on Pattern Recognition*, pages 160–163, 2002.

[95] S.-J. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel fisher discriminant analysis. *in Proc. of the 23rd International Conference on Machine Learning*, 2006.

[96] A. Klami and S. Kaski. Generative models that discover dependencies between two data sets. *in Proc. of IEEE Machine Learning for Signal Processing XVI*, pages 123–128, 2006.

[97] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg. Non-sparse multiple kernel learning. *in NIPS 08 Workshop: Kernel Learning - Automatic Selection of Optimal Kernels*, 2008.

[98] J. Kovač-Striko and K. Veselić. Trace minimization and definiteness of symmetric pencils. *Linear algebra and its applications*, 216:139–158, 1995.

[99] M. Kowalski, M. Szafranski, and L. Ralaivola. Multiple indefinite kernel learning with mixed norm regularization. *in Proc. of the 26th International Conference on Machine Learning*, 2009.

[100] K. Krishnan and J.E. Mitchell. Semi-infinite linear programming approaches to semidefinite programming problems. *Fields Institute Communications Series*, 37:123–142, 2003.

[101] A. Krogh and J. Vedelsby. Neural network ensembles, cross-validation and active learning. *Advances in Neural Information Processing Systems*, 1995.

[102] R. Kustra and A. Zagdanski. Data-fusion in clustering microarray data: Balancing discovery and interpretability. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 99(1), 2007.

[103] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2000.

[104] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Reserach*, 5:27–72, 2004.

[105] G.R.G. Lanckriet, N. Cristianini, M.I. Jordan, and W.S. Noble. *Kernel Methods in Computational Biology*, chapter Kernel-based integration of genomic data using semideinite programming. MIT Press, 2004.

[106] G.R.G. Lanckriet, T. De Bie, N. Cristianini, M.I. Jordan, and W.S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, 2004.

[107] T. Lange and J.M. Buhmann. Fusion of similarity data in clustering. *in Proc. of NIPS 2005*, 2005.

[108] C. Leslie, E. Eskin, J. Weston, and W.S. Noble. The spectrum kernel: a string kernel for svm protein classification. *in Proc. of the Pacific Symposium on Biocomputing 2002*, 2002.

[109] L. Leydesdorff. Can scientific journals be classified in terms of aggregated journal-journal citation relations using the journal citation reports? *Jounral of the American Society of Information Science*, 57(5):601–613, 2006.

[110] L. Leydesdorff and I. Rafols. A global map of science based on the isi subject categories. *Jounral of the American Society of Information Science*, 60(2):348–362, 2009.

[111] G. H. Little, Y. Bai, T. Williams, and C. Poizat. Nuclear calcium/calmodulin-dependent protein kinase IIdelta preferentially transmits signals to histone deacetylase 4 in cardiac cells. *J Biol Chem.*, 282(10):7219–7231, 2007.

[112] Q. Liu, K. Crammer, F.C.N. Pereira, and D.S. Roos. Reranking candidate gene models with cross-species comparison for improved gene prediction. *BMC Bioinformatics*, 9(433).

[113] X. Liu, S. Yu, Y. Moreau, W. Glänzel, B. De Moor, and F Janssens. Weighted hybrid clustering by combining text mining and bibliometrics on large-scale journal database. *Internal Report*, 2009.

[114] X. H. Liu, S. Yu, Y. Moreau, B. De Moor, W. Glänzel, and F. Janssens. Hybrid clustering of text mining and bibliometrics applied to journal sets. *in Proc. of SIAM Data Mining Conference 09*, 2009.

[115] J. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, Heidelberg, 1987.

[116] N. Lopez-Bigas and C.A. Ouzounis. Genome-wide indentification of genes likely to be involved in human genetic disease. *Nucleic Acids Research*, 32:3108–3114, 2004.

[117] J. Ma, J. L. Sancho-Gómez, and S. C. Ahalt. Nonlinear multiclass discriminant analysis. *IEEE Signal Processing Letters*, 10(7):196–199, 2003.

[118] X. Mao, T. Cai, J. G. Olyarchuk, and L. Wei. Automated genome annotation and pathway identification using the kegg orthology (ko) as a controlled vocabulary. *Bioinformatics*, 21(19):3787–3793, 2005.

[119] I. V. Marshakova. System of connections between documents based on references (as the science citation index). *Nauchno-Tekhnicheskaya Informatsiya Seriya*, 2(6):3–8, 1973.

[120] V A McKusick. *Mendelian Inheritance in Man. A Catalog of Human Genes and Genetic Disorders*. Johns Hopkins University Press, Baltimore, 1998 (12th edition).

[121] G. B. Melton, S. Parsons, F. P. Morrison, A. S. Rothschild, M. Markatou, and G. Hripcsak. Inter-patient distance metrics using snomed ct defining relationships. *Journal of Biomedical Informatics*, 39(6):697–705, 2006.

[122] S. Mika, G. Rätsch, J. Weston, and B. Schölkopf. Fisher discriminant analysis with kernels. *IEEE Neural Networks for Signal Processing IX*, pages 41–48, 1999.

[123] S. Mika, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Trans. on PAMI.*, 25(5):623–628, 2003.

[124] D. S. Modha and W. S. Spangler. Clustering hypertext with applications to web searching. *in Proceedings of the 7th ACM on Hypertext and Hypermedia*, pages 143–152, 2000.

[125] A. Mottaz, Y. L. Yip, P. Ruch, and A. L. Veuthey. Mapping proteins to disease terminologies: from uniprot to mesh. *BMC Bioinformatics*, 9(S5), 2008.

[126] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *The journal of Logic Programming*.

[127] J. W. Myers. Learning bayesian network from incomplete data with stochastic search algorithms. *in Proceedings of the 15th conference on Uncertainty in Artificial Intelligence*, 1999.

[128] Greek Myths. Pygmalion. urlhttp://en.wikipedia.org/wiki/Pygmalion_mythology.

[129] I. Nakić and K. Veselić. Wielandt and ky-fan theorem for matrix pairs. *Linear algebra and its applications*, 369(1):177–93, 2003.

[130] C. J. Needham, J. R. Bradford, A. J. Bulpitt, and D. R. Westhead. A primer on learning in bayesian networks for computational biology. *PLOS computational biology*, 3(8), 2007.

[131] Y. Nesterov and A. Nemirovskii. Interior-point polynomial algorithms in convex programming. *SIAM Studies in Applied Mathematics, SIAM*, 1994.

[132] A. Neveol. Multiple approaches to fine-grained indexing of the biomedical literature. *In Proceeding of PSB*, 2007.

[133] M.E.J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103(23):8577–8582.

[134] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 68(2), 2004.

[135] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. *In Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[136] A.Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *in Proceeding of NIPS 14*, 2001.

[137] National Institute of Standards and Technology. Nist form-based handprint recognition system (release 2.2). accessible via http://www.itl.nist.gov/iaui/894.03/databases/defs/nist_ocr.html.

[138] A.L.I. Oliveira, F.B.L. Neto, and S.R.L. Meira. Improving rbfdda performance on optical character recognition through parameter selection. *in Proc. of ICPR'04*, pages 625–628, 2004.

[139] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.

[140] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, San Mateo, CA, 1998.

[141] K. Pelckmans, S. Van Vooren, B. Coessens, J.A.K. Suykens, and B. De Moor. Mutual spectral clustering: Microarray experiments versus text corpus. *PMSB 2006*, 2006.

[142] C. Perez-Iratxeta, M. Wjst, P. Bork, and M.A. Andrade. G2d: a tool for mining genes associated with disease. *BMC Genetics*, 6(45), 2005.

[143] H. Plun-Favreau, G. Elson, M. Chabbert, J. Froger, O. deLapeyrière, E Lelièvre, C. Guillet, J. Hermann, J. F. Gauchat, H. Gascan, and S. Chevalier. The ciliary neurotrophic factor receptor α component induces the secretion of and is required for functional responses to cardiotrophin-like cytokine. *EMBO J.*, 20:1692–1703, 2001.

[144] M. Ramoni and P. Sebastiani. Robust learning with missing data. *Machine Learning*, pages 147–170, 2000.

[145] R. Reemtsen. Some other approximation methods for semi-infinite optimization problems. *Jounral of Computational and Applied Mathematics*, 53:87–108, 1994.

[146] N.J. Risch. Searching for genetic determinants in the new millennium. *Nature*, 405:847–856, 2000.

[147] S. M. Roth, E. J. Metter, M. R. Lee, B. F. Hurley, and R. E. Ferrell. C174T polymorphism in the CNTF receptor gene is associated with fat-free mass in men and women. *J. Appl. Physiol.*, 95(4):1425–1430, 2003.

[148] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65.

[149] K. C. Santosh, B. Lamiroy, and J.-P. Ropers. Inductive logic programming for symbol recognition. *in Proc. of the 10th International Conference on Document Analysis and Recognition*, pages 1330–1334, 2009.

[150] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. *in Proc. of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, pages 416–426, 2001.

[151] B. Scholköpf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[152] B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, Cambridge, USA, 2002.

[153] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, (5):1299–1319, 1998.

[154] H. Shatkay and R. Feldman. Mining the biomedical literature in the genomic era: An overview. *Journal of Computational Biology*, 10(6):821–855, 2003.

[155] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, Cambridge, UK, 2004.

[156] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, 2000.

[157] H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Jounral of the American Society of Information Science*, 24(4):265–269, 1973.

[158] H. Small. Cited documents as concept symbols. *Social Studies of Science*, 8(3):327–340, 1978.

[159] C.L. Smith, C A Goldsmith, and Eppig J T. The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biology*, 6(1):R7, 2004.

[160] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Jounral of Machine Learning Research*, 7:1531–1565, 2006.

[161] G.W. Stewart and J.-G. Sun. *Matrix perturbation theory*. Academic Press, Boston, 1990.

[162] A. Strehl and J. Ghosh. Cluster ensembles: A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.

[163] J.M. Stuart. A gene-coexpression network for global discovery of con-served genetic modules. *Science*, 302:249–255, 2003.

[164] J.F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[165] C. D. Sutton. Classification and regression trees, bagging, and boosting. *Handbook of Statistics*, 2005.

[166] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Publishing Co. Pte. Ltd., Singapore, 2002.

[167] R. Taton. La premiÍÍre note mathÍęmatique de gaspard monge (juin 1769). *Rev. Histoire Sci. Appl.*, 19:143–149.

[168] D.M.J. Tax and R.P.W Duin. Support vector domain description. *Pattern Recognition Letter*, 20:1191–1199, 1999.

[169] S. Theodoridis and K. Koutroumbas. *Pattern Recognition (2nd Edition)*. Elsevier Science (USA), 2003.

[170] N. Tiffin, J.F. Kelso, A.R. Powell, H. Pan, V.B. Bajic, and W.A. Hide. Integration of text- and data-mining using ontologies successfully selects disease gene candidates. *Nucleic Acids Research*, 33(5):1544–1552, 2005.

[171] A. Topchy, A.K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. PAMI*, 27:1866–1881.

[172] L. Tranchevent, R. Barriot, S. Yu, S. Van Vooren, P. Van Loo, B. Coessens, B. De Moor, S. Aerts, and Y. Moreau. ENDEAVOUR update: a web resource for gene prioriti-zation in multiple species. *Nucleic Acids Res.*, 36(1):W377–W384, 2008.

[173] K. Tretyakov. Methods of genomic data fusion: An overview. 2006.

[174] K. Tsuda, H.J. Shin, and B. Schölkopf. Fast protein classification with multiple networks. *Bioinformatics*, Suppl 2:59–65, 2005.

[175] F.S. Turner, D.R. Clutterbuck, and C.A.M. Semple. Pocus: mining genomic sequence annotation to predict disease genes. *Genome Biology*, 4(11), 2003.

[176] M.A. Van Driel, K. Cuelenaere, P.P.C.W. Kemmeren, J.A.M. Leunissen, H.G. Brunner, and G. Vriend. Geneseeker: extraction and integration of human disease-related information from web-based genetic databases. *Nucleic Acids Research*, W33:758–761, 2005.

[177] S. Van Vooren, B. Thienpont, B. Menten, F. Speleman, B. De Moor, J. R. Vermeesch, and Y. Moreau. Mapping biomedical concepts onto the human genome by mining literature on chromosomal aberrations. *Nucleic Acids Research*, 35(8):2533–2543, 2007.

[178] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[179] V. Vapnik. *The Natural of Statistical Learning Theory (2nd Edition)*. Springer-Verlag, New York, 1999.

[180] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, (2):264ÍC280, 1971.

[181] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[182] D. Wagner and F. Wagner. Between min cut and graph bisection. *In Proc. of the 18th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 744–750, 1993.

[183] F. Wang, C. Ding, and T. Li. Integrated kl(k-means-laplacian) clustering: A new clustering approach by combining attribute data and pairwise relations. *in Proceeding of SIAM SDM 09*, 2009.

[184] Y. Wang and M. Kitsuregawa. Evaluating contents-link coupled web page clustering for web search results. *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 490–506, 2002.

[185] D.L. Wheeler, T. Barrett, D.A. Benson, S.H. Bryant, K. Canese, V. Chetvernin, D.M. Church, M. Dicuccio, R. Edgar, S. Federhen, M. Feolo, L.Y. Geer, W. Helmberg, Y. Kapustin, O. Khovayko, D. Landsman, D.J. Lipman, T.L. Madden, D.R. Maglott, V. Miller, J. Ostell, K.D. Pruitt, G.D. Schuler, M. Shumway, E. Sequeira, S.T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, R.L. Tatusov, T.A. Tatusova, L. Wagner, and E. Yaschenko. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 36:D13–21, 2008.

[186] C. Williams and M. Seeger. The effect of the input density distribution on kernel-based classifiers. *in Proc. of 17th International Conference of Machine Learning*, 2000.

[187] R M Winter, M Baraitser, and J M Douglas. A computerised data base for the diagnosis of rare dysmorphic syndromes. *Journal of Medical Genetics*, 21:121–123, 1984.

[188] D.M. Wolf, L.F. Bodin, G. Price, J. Keasling, and A.P. Arkin. Memory in microbes: Quantifying history-dependent behavior in a bacterium. *PLOS one*, 3(2):e1700, 2008.

[189] M. Wu. *Label propagation for classification and ranking*. PhD thesis, Michigan State University, 2007.

[190] H. Yamakawa. Multi-aspect gene relation analysis. *In Proceeding of PSB*, 2005.

[191] J.P. Ye, S.W. Ji, and J.H. Chen. Multi-class discriminant kernel learning via convex programming. *Jounral of Machine Learning Research*, 9:719–758, 2008.

[192] S. Yu, W. Liu, X.H. Glänzel, B. De Moor, J.A.K. Suykens, and Y. Moreau. Clustering by multiple kernels and laplacians. *Internal Report, submitted to SIAM DM 10*, 2009.

[193] S. Yu, B. De Moor, and Y. Moreau. Adaptive kernel k-means clustering for heterogenous data fusion. *Internal Report, Submitted to IEEE Trans. Pattern Analysis and Machine Intelligence, under revision*, 2008.

[194] S. Yu, Y. Moreau, and B. De Moor. Learning with heterogeneous data sets by weighted multiple kernel canonical correlation analysis. *in Proc. of the IEEE Machine Learning for Signal Processing (IEEE MLSP)*, 2007.

[195] S. Yu, L.C. Tranchevent, S. Leach, R. Barriot, T. De Bie, B. De Moor, and Y. Moreau. Cross-species gene prioritization by genomic data fusion. *Internal Report*, 2009.

[196] S. Yu, S. Van Vooren, L. Tranchevent, B. De Moor, and Y. Moreau. Comparison of vocabularies, representations and ranking algorithms for gene prioritization by text mining. *Bioinformatics*, 24(16):i119–i125, 2008.

[197] Z. Yu, H.S. Wong, and H. Wang. Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics*, 23(21):2888–2896, 2007.

[198] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. *in Proc. of NIPS 14*, 2001.

[199] M. Zitt and E. Bassecoulard. Development of a method for detection and trend analysis of research fronts built by lexical or co-citation analysis. *Scientometrics*, 30(1):333–351, 1994.

Curriculum vitae

Shi Yu was born in Anhui, China, on November 6th, 1977. In year 1998 he obtained Bachelor of Mechanical and Electrical Engineering from China Textile University (present Donghua University) with the thesis titled "Simulation and software development of MCS-51 single chip processor ". Afterwards he worked in several E-commerce and E-government companies in Shanghai as a programmer, software engineer and project manager. He found his interests in *business intelligence*, known as data mining and machine learning in the academic terms. He restarted his studies in Katholieke Universiteit Leuven where he received a Master in Artificial Intelligence (magna cum laude) in September 2005. The thesis was titled as "Interpretation of gene profiles by biomedical text mining". From December 2005 he enrolled in a Ph.D program at the Department of Electrical Engineering (ESAT), in the lab of Signals, Identification, System Theory and Automation (SCD/SISTA). Since then he worked in the bioinformatics group, under the supervision of Prof. Yves Moreau and Prof. Bart De Moor.

Publications by the author

Journal Papers

- **S. Yu**, S. Van Vooren, L.-C. Tranchevent, B. De Moor, Y. Moreau, "Comparison of vocabularies, representations and ranking algorithms for gene prioritization by text mining", *Bioinformatics*, vol. 24, no. 16, pp. i119-125, 2008.
- **S. Yu**, L.-C. Tranchevent, B. De Moor, Y. Moreau, "Gene prioritization and clustering by multi-view text mining", *BMC Bioinformatics*, in publication, 2009.
- L.-C.Tranchevent, R. Barriot, **S. Yu**, S. Van Vooren, P. Van Loo, B. Coessens, B. De Moor, Y. Moreau, ENDEAVOUR update: a web resource for gene prioritization in multiple species. *Nucleic Acids Research*, vol. 36, no. 1, pp. W377- W384, 2008.
- X. Liu, **S. Yu**, F. Janssens, W. Glänzel, B. De Moor, Y. Moreau, "Weighted hybrid clustering by combining text mining and bibliometrics on large scale journal database", *Journal of American Information and Science Technology*, in publication, 2009. **(equally contributed author)**

Submitted papers and manuscripts in preparation

- **S. Yu**, L.-C. Tranchevent, X. Liu, W. Glänzel, J. Suykens, B. De Moor and Y. Moreau, "Optimized data fusion for kernel K-means clustering", *Internal Report 08-200, ESAT-SISTA, K.U.Leuven, 2008, Lirias number: 242275, submitted for publication.*
- **S. Yu**, L.-C. Tranchevent, S. Leach, R. Barriot, T. De Bie, B. De Moor, Y. Moreau, "Cross-species gene prioritization by genomic data fusion", *Manuscript in preparation*, 2009.

- **S. Yu**, X. Liu, W. Glänzel, B. De Moor, Y. Moreau, "Clustering with multiple kernels and Laplacians", *Internal report, K.U.Leuven, submitted for publication*, 2009.

- **S. Yu**, T. Falck, A. Daemen, J. Suykens, B. De Moor and Y. Moreau, "Non-sparse kernel fusion and its applications in genomic data integration", *Internal Report, K.U.Leuven, Lirias number: 248322, submitted for publication*, 2009.

International Conference Papers

- X. Liu, **S. Yu**, Y. Moreau, B. De Moor, W. Glänzel, F. Janssens, "Hybrid clustering of text mining and bibliometrics applied to journal sets", in Proceeding of SIAM Data Mining (SDM) Conference, 2009. Liu X. Yu S., Moreau Y., De Moor B., Glanzel Y., Janssens F. Hybrid clustering of text mining and bibliometrics applied to journal sets. *in Proceeding of SIAM Data Mining (SDM) Conference*, 2009. **(equally contributed author)**

- **S. Yu**, B. De Moor, Y. Moreau, "Clustering by heterogeneous data fusion: framework and applications", *in Workshop of learning from multiple data sources, NIPS*, 2008.

- **S. Yu**, B. De Moor, Y. Moreau, "Learning with heterogenous data sets by Weighted Multiple Kernel Canonical Correlation Analysis", *in Proceeding of Machine Learning for Signal Processing XVII, IEEE*, pp. 81-86, 2007.

- **S. Yu**, S. Van Vooren, B. Coessens, B. De Moor, "Interpreting gene profiles from biomedical literature mining with self organization maps", *in Lecture Notes in Computer Science: Advances in Neural Networks - ISNN 2006: Third International Symposium on Neural Networks*, pp. 635-641, 2006.

- X. Liu, **S. Yu**, Y. Moreau Y, B. De Moor, W. Glänzel, F. Janssens, "Hybrid Clustering by Integrating Text and Citation based Graphs in Journal Database Analysis", *accepted for publication in ICDM-09 Workshop on Mining Multiple Information Sources*, 2009.

Arenberg Doctoral School of Science, Engineering & Technology
Faculty .....
Department ...
Research group ...
Address ...

KATHOLIEKE UNIVERSITEIT LEUVEN

ASSOCIATIE K.U.LEUVEN