Fast in-memory spectral clustering using a fixed-size approach

R. Langone*[1], R. Mall[2], V. Jumutc[1], J. A. K. Suykens[1]

[1]KU Leuven, ESAT-STADIUS, Kasteelpark Arenberg 10,
B-3001 Leuven (Belgium)
[2]Qatar Computing Research Institute (QCRI), Doha (Qatar)
*Corresponding author, e-mail: rocco.langone@esat.kuleuven.be

Abstract. Spectral clustering represents a successful approach to data clustering. Despite its high performance in solving complex tasks, it is often disregarded in favor of the less accurate k-means algorithm because of its computational inefficiency. In this article we present a fast in-memory spectral clustering algorithm, which can handle millions of datapoints at a desktop PC scale. The proposed technique relies on a kernel-based formulation of the spectral clustering problem, also known as kernel spectral clustering. In particular, we use a fixed-size approach based on an approximation of the feature map via the Nyström method to solve the primal optimization problem. We experimented on several small and large scale real-world datasets to show the computational efficiency and clustering quality of the proposed algorithm.

1 Introduction

Data clustering represents a valuable data analysis tool in modern applications of artificial intelligence. In many domains clustering is used to gain first insights in the data under investigation and to provide solutions to several real-life problems, from customer segmentation in marketing campaigns to fault detection in industries within a predictive maintenance strategy.

Spectral clustering (SC) [1, 2, 3, 4] is considered among the most successful clustering algorithms, mainly due to its ability of discovering nonlinear relationships in the data. A major drawback of SC is its cubic computational complexity and high memory cost. Several algorithms have been devised to scale SC, which include power iteration clustering [5], spectral clustering in conjunction with the Nyström approximation [6], incremental spectral clustering [7, 8, 9], parallel spectral clustering [10], kernel spectral clustering [11] etc.

Kernel spectral clustering or KSC represents a kernel-based formulation of SC and, in contrast to the other methods, allows to tackle the issues of selecting an appropriate number of clusters and predicting the memberships of new points using a kernel-based modeling approach. The KSC algorithm has been optimized to handle big network data by taking advantage of their inherent sparse format [12, 13]. In particular, a fast cosine kernel computation (based on the *Python* dictionary data type) and the usage of the out-of-sample extension property with a small representative training set have been exploited. Furthermore, in [14] various penalty-based reduced set techniques (including the Group Lasso, L_0

and $L_0 + L_1$ penalizations) have been proposed to reduce the time complexity of the expensive out-of-sample extension and to obtain a sparser model.

In this paper we propose an alternative strategy to cluster large-scale vector data by means of a fixed-size procedure, which was originally proposed in [15] and optimized in [16] only for classification and regression problems. The approach relies on the Nyström approximation [17] of the nonlinear mapping induced by the kernel matrix to solve the primal optimization problem. In particular, if we denote with N the total number of datapoints, we show how the solution to the primal optimization problem can be obtained by computing the eigenvalue decomposition of an $m \times m$ matrix, where $m \ll N$ indicates the dimension of the approximate explicit feature map. The latter is constructed by using a random subset of size m extracted from the entire dataset.

This paper is organized as follows. In Section 2 the standard KSC algorithm is briefly reviewed. Section 3 introduces the proposed approach, where a primal KSC model instead of a dual model is derived using an approximated explicit feature map. Section 4 reports the experimental results and finally some conclusions are draw in Section 5.

2 Kernel Spectral Clustering

Kernel spectral clustering (KSC [11]) is a formulation of the spectral clustering problem in the least squares support vector machines [15] learning framework. This setting brings two main advantages, namely a rigorous tuning procedure for the selection of a proper number of clusters and the prediction of the cluster memberships for unseen points using an out-of-sample extension property.

Given a set of N datapoints $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ to be clustered in k clusters, with $\mathbf{x}_i \in \mathbb{R}^d$, the primal KSC optimization problem related to N_{tr} training data is given by the following weighted kernel PCA formulation [11]:

$$\begin{aligned} \min_{\mathbf{w}^{(l)},\mathbf{e}^{(l)},b_l} \quad & \frac{1}{2}\sum_{l=1}^{k-1} \mathbf{w}^{(l)^T}\mathbf{w}^{(l)} - \frac{1}{2}\sum_{l=1}^{k-1} \gamma_l \mathbf{e}^{(l)^T} D^{-1}\mathbf{e}^{(l)} \\ \text{subject to} \quad & \mathbf{e}^{(l)} = \Phi\mathbf{w}^{(l)} + b_l \mathbf{1}_{N_{\text{tr}}}, l = 1,\ldots,k-1. \end{aligned} \tag{1}$$

Equation (1) means that one wants to find some directions $\mathbf{w}^{(l)}$ with minimal norm such that the weighted variances of the projections along these directions, i.e. $\mathbf{e}^{(l)^T} D^{-1}\mathbf{e}^{(l)}$ are maximized.

The symbols have the following meaning: $D^{-1} \in \mathbb{R}^{N_{\text{tr}} \times N_{\text{tr}}}$ denotes the inverse of the degree matrix D, which is diagonal with diagonal $\mathbf{d} = \Phi\Phi^T\mathbf{1}_{N_{\text{tr}}}$, Φ is the $N_{\text{tr}} \times d_h$ feature matrix $\Phi = [\varphi(\mathbf{x}_1)^T; \ldots; \varphi(\mathbf{x}_{N_{\text{tr}}})^T]$, $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ indicates the mapping to a high-dimensional feature space, b_l are bias terms. The $\mathbf{e}^{(l)} = [e_1^{(l)}, \ldots, e_i^{(l)}, \ldots, e_{N_{\text{tr}}}^{(l)}]^T$ indicate the clustering scores, that is the projections of the training data mapped in the feature space along the directions $\mathbf{w}^{(l)}$, and for a given point $\mathbf{x}_i$ can be computed as $e_i^{(l)} = \mathbf{w}^{(l)^T}\varphi(\mathbf{x}_i) + b_l$. Finally, in case of a new datapoint $\mathbf{x}_i^{\text{test}}$, the related clustering score can be obtained as

$e_i^{(l),\text{test}} = \mathbf{w}^{(l)^T}\varphi(\mathbf{x}_i^{\text{test}}) + b_l$, which corresponds to the out-of-sample property mentioned earlier. Since in general the feature map Φ is unknown and can be even infinite-dimensional (in case for instance of a Gaussian kernel), from the KKT conditions for optimality of the Lagrangian associated with (1) one can derive the following dual problem:

$$M_D D^{-1}\Omega\alpha^{(l)} = \lambda_l\alpha^{(l)} \tag{2}$$

where $\Omega = \Phi\Phi^T$ indicates the kernel matrix, M_D is a centering matrix and $\lambda_l = \frac{1}{\gamma_l}$. The solutions $\alpha^{(l)}$ allow to compute the clustering score for the i-th training point as $e_i^{(l)} = \sum_{j=1}^{N_{\text{tr}}}\Omega_{ij}\alpha_j^{(l)}$, without explicitly knowing the expression of the feature map. Finally, the clustering memberships can be obtained by taking the sign of the projections and using an Error Correcting Output Codes (ECOC) coding scheme, similarly to what can be used in case of a standard support vector machine classifier.

3 Proposed algorithm

When the number of datapoints N is large, there are two possible solutions to handle the clustering problem by means of the KSC algorithm: (i) select a small number of training data $N_{\text{tr}} \ll N$, train a KSC model by solving the dual of (1), compute the cluster memberships for the remaining points by means of the out-of-sample extension property; (ii) utilize a fixed-size approach by solving the primal problem, as proposed in [15] in case of classification and regression. In this paper we follow the second direction.

The proposed algorithm, named fixed-size kernel spectral clustering or KSC-FS, is based on the following unconstrained formulation of the KSC primal objective:

$$\min_{\hat{\mathbf{w}}^{(l)},\hat{b}_l} \quad \frac{1}{2}\sum_{l=1}^{k-1}\hat{\mathbf{w}}^{(l)^T}\hat{\mathbf{w}}^{(l)} - \frac{1}{2}\sum_{l=1}^{k-1}\gamma_l(\hat{\Phi}\hat{\mathbf{w}}^{(l)} + \hat{b}_l\mathbf{1}_{N_{\text{tr}}})^T\hat{D}^{-1}(\hat{\Phi}\hat{\mathbf{w}}^{(l)} + \hat{b}_l\mathbf{1}_{N_{\text{tr}}}) \tag{3}$$

where $\hat{\Phi} = [\hat{\varphi}(\mathbf{x}_1)^T; \ldots; \hat{\varphi}(\mathbf{x}_{N_{\text{tr}}})^T] \in \mathbb{R}^{N_{\text{tr}}\times m}$ is the approximated feature matrix, $\hat{D} \in \mathbb{R}^{N_{\text{tr}}\times N_{\text{tr}}}$ denotes the corresponding degree matrix, and $\hat{\varphi} : \mathbb{R}^d \to \mathbb{R}^m$ indicates a finite dimensional approximation of the feature map $\varphi(\cdot)$. The m points needed to estimate the components of $\hat{\varphi}$ can be selected at random or by means of active sampling techniques such as the Renyi entropy criterion. In order to minimize (3) we can take the partial derivatives of the optimization function $J(\hat{\mathbf{w}}^{(l)}, \hat{b}_l)$ w.r.t. to the primal variables:

$$\begin{aligned}
\frac{\partial\mathcal{J}}{\partial\hat{\mathbf{w}}^{(l)}} = 0 \quad &\rightarrow \quad \hat{\mathbf{w}}^{(l)} = \gamma_l(\hat{\Phi}^T\hat{D}^{-1}\hat{\Phi}\hat{\mathbf{w}}^{(l)} + \hat{\Phi}^T\hat{D}^{-1}\mathbf{1}_{N_{\text{tr}}}\hat{b}_l) \\
\frac{\partial\mathcal{J}}{\partial\hat{b}_l} = 0 \quad &\rightarrow \quad \mathbf{1}_{N_{\text{tr}}}^T\hat{D}^{-1}\hat{\Phi}\hat{\mathbf{w}}^{(l)} = -\mathbf{1}_{N_{\text{tr}}}^T\hat{D}^{-1}\mathbf{1}_{N_{\text{tr}}}\hat{b}_l.
\end{aligned}$$

After some simple algebraic manipulations one obtains the following eigenvalue problem to solve:

$$R\hat{\mathbf{w}}^{(l)} = \lambda_l \hat{\mathbf{w}}^{(l)} \tag{4}$$

with $\lambda_l = \frac{1}{\gamma_l}$, $R = \hat{\Phi}^T \hat{D}^{-1} \hat{\Phi} - \frac{(\mathbf{1}_{N_{\text{tr}}}^T \hat{D}^{-1} \hat{\Phi})^T (\mathbf{1}_{N_{\text{tr}}}^T \hat{D}^{-1} \hat{\Phi})}{\mathbf{1}_{N_{\text{tr}}}^T \hat{D}^{-1} \mathbf{1}_{N_{\text{tr}}}}$ and $\hat{b}_l = -\frac{\mathbf{1}_{N_{\text{tr}}}^T \hat{D}^{-1} \hat{\Phi}}{\mathbf{1}_{N_{\text{tr}}}^T \hat{D}^{-1} \mathbf{1}_{N_{\text{tr}}}} \hat{\mathbf{w}}^{(l)}$. Notice that we now have to solve an eigenvalue problem of size $m \times m$, which can be done very efficiently by choosing m such that $m \leq_{\text{tr}} \ll N$. Furthermore, we can compute the diagonal of matrix $\hat{D}$ as $\hat{\mathbf{d}} = \hat{\Phi}(\hat{\Phi}^T \mathbf{1}_m)$, without constructing the (potentially) large matrix $\hat{\Phi}\hat{\Phi}^T$. Once we have solved problem (4), the cluster memberships can be obtained by applying the k-means algorithm on the projections $\hat{e}_i^{(l)} = \hat{\mathbf{w}}^{(l)^T} \hat{\varphi}(\mathbf{x}_i) + \hat{b}_l$ for training data and $\hat{e}_i^{(l),\text{test}} = \hat{\mathbf{w}}^{(l)^T} \hat{\varphi}(\mathbf{x}_i^{\text{test}}) + \hat{b}_l$ in case of test points.

In order to compute the approximated feature map, one can apply the Nyström method to solve numerically the Fredholm integral equation. In particular, the i-th component of the m dimensional feature map $\hat{\varphi}$ for a given point $\mathbf{x}$ can be calculated as follows [18]:

$$\hat{\varphi}_i(x) = \frac{1}{\beta_i^{(s)}} \sum_{j=1}^{m} u_{ji} K(\mathbf{x}_j, \mathbf{x}) \tag{5}$$

where $\beta_i^{(s)}$ and u_i are the eigenvalues and eigenvectors of the $m \times m$ kernel matrix $\hat{\Omega} = \hat{\Phi}\hat{\Phi}^T$, with $K(\mathbf{x}_i, \mathbf{x}_j) = \hat{\Omega}_{ij}$.

A Matlab implementation of the algorithm can be freely downloaded at: *http://www.esat.kuleuven.be/stadius/ADB/langone/softwareKSCFSlab.php*

4 Experimental Results

In this Section the results of the simulations performed on several real-world datasets (mostly) from the UCI machine learning repository are reported. In all the experiments we have used the following settings[1]: $m = 100$, $N_{\text{tr}} = 0.80N$, $N_{\text{test}} = 0.20N$, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{\sigma^2})$ (Gaussian kernel). The m datapoints are selected at random[2] and each simulation is repeated 30 times. For simplicity, the number of clusters k has been set equal to the number of classes, and a grid search procedure using the balanced angular fit (BAF [12]) as quality criterion has been used to select an optimal σ. The cluster quality is assessed using an internal quality metric, namely the Davies-Bouldin [19] criterion, and an external quality metric such as the the adjusted rand index (ARI [20]). In the latter case we follow the cluster assumption [21], according to which *if points are in the same cluster they are likely to be of the same class.*

[1] We have also experimented with $m = 500$, $m = 1000$ and $m = 5000$, but we found out that $m = 100$ represents a good trade-off between cluster quality and computation time.

[2] Also the usage of the Renyi entropy criterion has been investigated. In general we have observed that, compared to random sampling, the Renyi entropy sampling method leads to less variable outcomes (among the different runs) and a similar mean cluster quality.

Table 1 reports the performance of the proposed algorithm and the k-means approach in terms of execution time, ARI and DB index. In general, the KSC-FS approach performs better in terms of ARI and worse according to the DB index. Regarding the runtime, it is competitive in most of the datasets and outperforms k-means in case of the largest databases (i.e. *Susy* and *Higgs*).

Dataset	N	d	KSC-FS			K-means		
			ARI	DB	Time (s)	ARI	DB	Time (s)
Iris	150	4	**0.64**	0.85	0.012	0.57	**0.83**	0.005
Ecoli	336	8	0.50	1.57	0.023	0.50	**1.17**	0.010
Dermatology	366	33	**0.83**	**1.87**	0.017	0.69	1.91	0.013
Vowel	528	10	**0.12**	1.67	0.053	0.09	**1.60**	0.023
Libras	360	91	**0.32**	1.46	0.030	0.29	**1.32**	0.046
Pen Digits	10 992	16	**0.61**	1.63	0.064	0.57	**1.43**	0.161
Opt Digits	5 620	64	0.52	3.12	0.085	0.52	**1.93**	0.374
S1	5 000	2	**0.96**	**0.40**	0.046	0.89	0.49	0.019
S4	5 000	2	**0.66**	**0.67**	0.066	0.64	0.68	0.066
Spambase	4 601	57	**0.38**	3.87	0.020	0.22	**1.83**	0.100
Magic	19 020	11	**0.04**	3.28	0.093	0.006	**1.43**	0.078
Shuttle	58 000	9	0.29	2.00	0.368	**0.35**	**0.75**	0.212
Skin	245 057	3	**0.03**	**0.67**	0.415	-0.03	0.69	0.280
RCV1	20 242	1 960	**0.08**	2.03	1.139	0.008	**0.67**	1.140
Covertype	581 012	54	**0.07**	3.85	4.550	0.05	**1.89**	4.291
GalaxyZoo [22]	667 944	9	0.25	1.69	3.047	**0.27**	**1.12**	2.558
Susy	5 000 000	18	**0.12**	2.17	18.96	0.11	**2.08**	59.54
Higgs	11 000 000	28	**0.008**	3.34	27.11	0.006	**2.68**	129.7

Table 1: **Clustering results on real-world datasets**. Comparison of the proposed KSC-FS approach against the k-means algorithm. In case of the KSC-FS method, the runtime comprises both training and test stages.

5 Conclusions

In this paper we have presented an efficient and accurate in-memory clustering algorithm. The proposed technique uses a fixed-size approach based on an approximation of the feature map (via the Nyström method) to solve the primal optimization problem characterizing a kernel spectral clustering model. A number of experiments performed on well-known real-world datasets confirm the usefulness of the proposed algorithm.

Acknowledgments

EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grant iMinds Medical Information Technologies SBO 2015 IWT: POM II SBO 100031 Belgian Federal Science Policy Office: IUAP P7/19 2012-2017).

References

[1] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

[2] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, Cambridge, MA, 2002. MIT Press.

[3] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[4] Hongjie Jia, Shifei Ding, Xinzheng Xu, and Ru Nie. The latest research progress on spectral clustering. *Neural Computing and Applications*, 24(7-8):1477–1486, 2014.

[5] Frank Lin and William W. Cohen. Power iteration clustering. In *ICML*, pages 655–662, 2010.

[6] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, February 2004.

[7] Huazhong Ning, Wei Xu, Yun Chi, Yihong Gong, and Thomas S. Huang. Incremental spectral clustering with application to monitoring of evolving blog communities. In *SDM*. SIAM, 2007.

[8] A. M. Bagirov, B. Ordin, G. Ozturk, and A. E. Xavier. An incremental clustering algorithm based on hyperbolic smoothing. *Computational Optimization and Applications*, 61(1):219–241, 2014.

[9] R. Langone, O. M. Agudelo, B. De Moor, and J. A. K. Suykens. Incremental kernel spectral clustering for online learning of non-stationary data. *Neurocomputing*, 139(0):246–260, September 2014.

[10] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and E.Y. Chang. Parallel spectral clustering in distributed systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):568–586, March 2011.

[11] C. Alzate and J. A. K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):335–347, February 2010.

[12] R. Mall, R. Langone, and J. A. K. Suykens. Kernel spectral clustering for big data networks. *Entropy (Special Issue on Big Data)*, 15(5):1567–1586, 2013.

[13] R. Mall, R. Langone, and J. A. K. Suykens. Self-Tuned Kernel Spectral Clustering for Large Scale Networks. In *IEEE International Conference on Big Data*, pages 1–9, 2013.

[14] R. Mall, S. Mehrkanoon, R. Langone, and J. A. K. Suykens. Optimal reduced sets for sparse kernel spectral clustering. In *Proc. of the International Joint Conference on Neural Networks (IJCNN 2014)*, pages 2436 – 2443, July 2014.

[15] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[16] K. De Brabanter, J. De Brabanter, J. A. K. Suykens, and B. De Moor. Optimized fixed-size kernel models for large data sets. *Computational Statistics & Data Analysis*, 54(6):1484–1504, June 2010.

[17] C. Baker. *The numerical treatment of integral equations*. Clarendon Press, Oxford, 1977.

[18] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[19] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, April 1979.

[20] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 1(2):193–218, 1985.

[21] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[22] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. C. Nichol, M. J. Raddick, A. Szalay, D. Andreescu, P Murray, and J. Vandenberg. Galaxy zoo 1: data release of morphological classifications for nearly 900000 galaxies. *Mon. Not. R. Astron. Soc.*, (410):166–178, 2011.