Multi-Instance Learning

Soumya Ray, Oregon State University   Stephen Scott, University of Nebraska
Hendrik Blockeel, K. U. Leuven

March 31, 2007

1 Byline

2 Synonyms

Multiple-Instance Learning

3 Definition

Multiple-Instance (MI) Learning is an extension of the standard supervised learning setting. In standard supervised learning, the input consists of a set of labeled instances each described by an attribute vector. The learner then induces a concept that relates the label of an instance to its attributes. In multiple-instance learning, the input consists of labeled examples (called "bags") consisting of *multisets* of instances, each described by an attribute vector, and there are constraints that relate the label of each bag to the unknown labels of each instance. The multiple-instance learner then induces a concept that relates the label of a bag to the attributes describing the instances in it. This setting contains supervised learning as a special case: if each bag contains exactly one instance, it reduces to a standard supervised learning problem.

4 Motivation and Background

The multiple-instance setting was introduced by Dietterich et al. [1] in the context of drug activity prediction. Drugs are typically molecules that fulfill some desired function by binding to a target. If we wish to learn the characteristics responsible for binding, a possible representation of the problem is to represent each molecule as a set of low energy shapes or *conformations*, and describe each conformation using a set of attributes. Each such bag of conformations is given a label corresponding to whether the molecule is active or inactive. To learn a classification model, an algorithm assumes that every instance in a bag labeled negative is actually negative, whereas at least one instance in a bag labeled positive is actually positive with respect to the underlying concept.

From a theoretical viewpoint, multiple-instance learning occupies an intermediate position between standard propositional supervised learning and first-order relational learning. Supervised learning is a special case of MI learning, while MI learning is a special case of first-order learning. It has been argued that the multiple-instance setting is a key transition between standard supervised and relational learning [2]. At the same time, theoretical results exist that show that, under certain assumptions, certain concept classes that are PAC-learnable in a supervised setting remain PAC-learnable in a multiple-instance setting. Thus, the MI setting is able to leverage some of the rich representational power of relational learners while not sacrificing the efficiency of propositional learners. Figure 1 illustrates the relationships between standard supervised learning, multiple-instance learning and relational learning.

Since its introduction, a wide variety of tasks have been formulated as multiple-instance learning problems. Many new algorithms have been developed, and well-known supervised learning algorithms extended, to learn MI concepts. A great deal of work has also been done to understand what

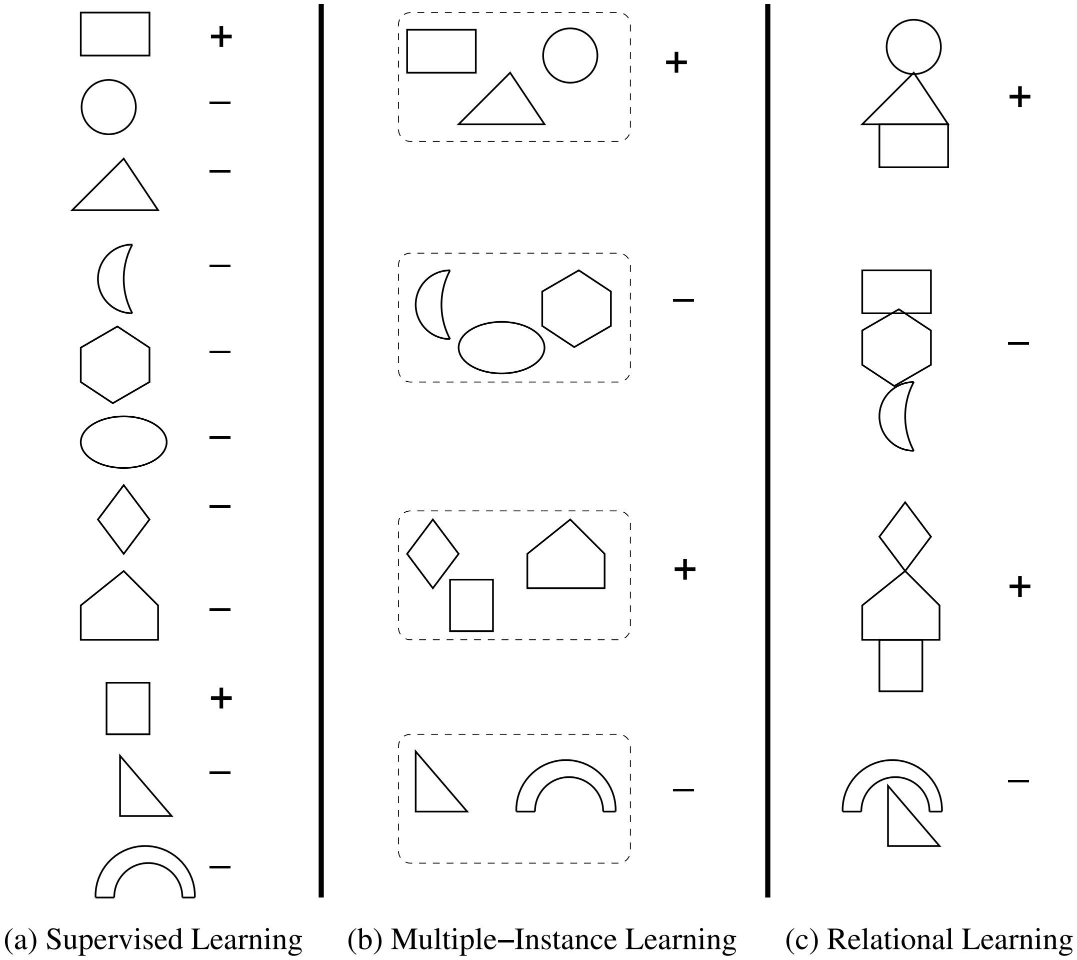

Figure 1: The relationship between supervised, multiple-instance and relational learning. (a) In supervised learning, each example (geometric figure) is labeled. A possible concept that explains the example labels shown is “the figure is a rectangle”. (b) In MI learning, bags of examples are labeled. A possible concept that explains the bag labels shown is “the bag contains at least one figure that is a rectangle.” (c) In relational learning, objects of arbitrary structure are labeled. A possible concept that explains the object labels shown is “the object is a stack of three figures and the bottom figure is a rectangle.”

Given: A set of bags $\{B_1, ...B_n\}$ each with label $\ell_i \in \{0,1\}$. Each B_i is a multiset of n_i instances, $B_i = \{B_{i1}, \ldots, B_{in_i}\}$.

Constraints: There exists a concept c such that:

- For every B_i with $\ell_i = 1$, $c(B_{ij}) = 1$ for at least one j, and
- For every B_i with $\ell_i = 0$, $c(B_{ij}) = 0$ for all j.

Do: Learn a concept that maps a bag B_i to its label ℓ_i.

Figure 2: Statement of the multiple-instance classification problem.

kinds of concepts can and cannot be learned efficiently in this setting. In the following sections, we discuss the theory, methods and applications of multiple-instance learning in more detail.

5 Structure of the Problem

The general multiple-instance classification task in shown in Figure 2. The multiple-instance regression task is defined analogously by substituting a real-valued response for the classification label. In this case, the constraint used by the learning algorithm is that the response of any bag is equal to the response of at least one of the instances in it, for example, it could be equal to the largest response over all the instances.

Notice the following problem characteristics:

- The number of instances in each bag can vary independently of other bags. This implies in particular that an MI algorithm must be able to handle bags with as few as one instance (this is a supervised learning setting) to bags with large numbers of instances.
- The number of instances in any positive bag that are "truly positive" could be many more than one—in fact, the definition does not rule out the case where *all* instances in a positive bag are "truly positive."
- The problem definition does not specify how the instances in any bag are related to each other.

6 Theory and Methods

In this section we discuss some of the key algorithms and theoretical results in multiple-instance learning. We first discuss methods and results for multiple-instance classification. Then we discuss work on multiple-instance regression.

6.1 Multiple-Instance Classification

Axis-Parallel Rectangles (APRs) are a concept class that early work in multiple-instance classification focused on. These generative concepts specify upper and lower bounds for all numeric attributes describing each instance. An APR is said to "cover" an instance if the instance lies within it. An APR covers a bag if it covers at least one instance within it. The learning algorithm tries to find an APR such that it covers all positive bags and does not cover any negative bags.

An algorithm called "iterated-discrimination" was proposed by Dietterich et al. [1] to learn APRs from multiple-instance data. This algorithm has two phases. In the first phase, it iteratively chooses a set of "relevant" attributes, and grows an APR using this set. This phase results in the construction of a very "tight" APR that covers just positive bags. In the second phase, the algorithm expands this APR so that with high probability, a new positive instance will fall within the APR.

The key steps of the algorithm are outlined below. Note that initially, all attributes are considered to be "relevant".

The algorithm starts by choosing a random instance in a positive bag. Let us call this instance I_1. The smallest APR covering this instance is a point. The algorithm then expands this APR by finding the smallest APR that covers any instance from a yet uncovered positive bag; call the newly covered instance I_2. This process is continued, identifying new instances $I_3, \dots, I_k$, until all positive bags are covered. At each step, the APR is "backfitted" in a way that is reminiscent of the later Expectation-Maximization (EM) approaches: each earlier choice is revisited, and I_j is replaced with an instance from the same bag that minimizes the current APR (which may or may not be the same as the one that minimized it at step j).

This process yields an APR that imposes maximally tight bounds on all attributes and covers all positive bags. Based on this APR, a new set of "relevant" attributes is selected as follows. An attribute's relevance is determined by how strongly it discriminates against negative instances, i.e. given the current APR bounds, how many negative instances the attribute excludes. Features are then chosen iteratively and greedily according to how relevant they are until all negative instances have been excluded. This yields a subset of (presumably relevant) attributes. The APR growth procedure in the previous paragraph is then repeated, with the size of an APR redefined as its size along relevant attributes only. The APR-growth and attribute selection phases are repeated until the process converges.

The APR thus constructed may still be too tight, as it fits narrowly around the positive bags in the dataset. In the second phase of the algorithm, the APR bounds are further expanded using a kernel density estimate approach. Here, a probability distribution is constructed for each relevant attribute using Gaussian distributions centered at each instance in a positive bag. Then, the bounds on that attribute are adjusted so that with high probability, any positive instance will lie within the expanded APR.

Theoretical analyses of APR concepts have been performed along with the empirical approach, using Valiant's "probably approximately correct" (PAC) learning model [3]. In early work [4], it was shown that if each instance was drawn according to a fixed, unknown, product distribution over the rational numbers, independently from every other instance, then an algorithm could PAC-learn APRs. Later, this result was improved in two ways [5]. First, the restriction that the individual instances in each bag come from a product distribution was removed. Instead, each instance is generated by an arbitrary probability distribution (though each instance in a bag is still generated independent and identically distributed (iid) according to that one distribution). Second, the time and sample complexities for PAC-learning APRs was improved. Specifically, the algorithm described in this work PAC-learns APRs in time

$$O\left(\frac{d^3n^2}{\epsilon^2}\log\frac{nd\log(1/\delta)}{\epsilon}\log\frac{d}{\delta}\right)$$

using

$$O\left(\frac{d^2n^2}{\epsilon^2}\log\frac{d}{\delta}\right)$$

labeled training bags. Here, d is the dimension of each instance, n is the (largest) number of instances per training bag and ϵ and δ are parameters to the algorithm. A variant of this algorithm was empirically evaluated and found to be successful [6].

Diverse Density [7, 8] is a probabilistic generative framework for multiple-instance classification. The idea behind this framework is that, given a set of positive and negative bags, we wish to learn a concept that is "close" to at least one instance from each positive bag, while remaining "far" from every instance in every negative bag. Thus, the concept must describe a region of instance space that is "dense" in instances from positive bags, and is also "diverse" in that it describes every positive bag. More formally, let

$$DD(t) = \frac{1}{Z}\left(\prod_i \Pr(t|B_i^+)\prod_i \Pr(t|B_i^-)\right),$$

where t is a candidate concept, B_i^+ represents the i^{th} positive bag, and B_i^- represents the i^{th} negative bag. We seek a concept that maximizes $DD(t)$. The concept generates the instances of a bag, rather than the bag itself. To score a concept with respect to a bag, we combine t's probabilities for instances using a function based on noisy-or [9]:

$$\Pr(t|B_i^+) \quad \propto \quad (1 - \prod_j (1 - \Pr(B_{ij}^+ \in t))) \tag{1}$$

$$\Pr(t|B_i^-) \quad \propto \quad \prod_j (1 - \Pr(B_{ij}^- \in t)) \tag{2}$$

Here the instances B_{ij}^+ and B_{ij}^- belonging to t are the "causes" of the "event" that "t is the target". The concept class investigated by Maron [8] is the class of generative Gaussian models, which are parameterized by the mean μ and a "scale" $s = \frac{1}{2\sigma^2}$:

$$\Pr(B_{ij} \in t) \propto e^{-\sum_k (s_k (B_{ijk} - \mu_k)^2)},$$

where k ranges over attributes. Figure 3 illustrates a concept that Diverse Density might learn when applied to an MI dataset.

Diverse Density with k disjuncts is a variant of Diverse Density that has also been investigated [8]. This is a class of disjunctive Gaussian concepts, where the probability of an instance belonging to a concept is given by the maximum probability of belonging to any of the disjuncts.

EM-DD [10] is an example of a class of algorithms that try to identify the "cause" of a bag's label using Expectation-Maximization (EM). These algorithms sometimes assume that there is a single instance in each bag that is responsible for the bag's label (though variants using "soft EM" are possible). The key idea behind this approach is as follows: from each positive bag, we take a random instance and assume that this instance is the relevant one. We learn a hypothesis from these relevant instances and all negative bags. Next, for each positive bag, we replace the current relevant instance by the instance most consistent with the learned hypothesis (which will initially not be the chosen instance in general). We then relearn the hypothesis with these new instances. This process is continued until the set of chosen instances does not change (or alternatively, the objective function of the classifier reaches a fixed point). This procedure has the advantage of being computationally efficient, since the learning algorithm only uses one instance from each positive bag. This approach has also been used in multiple-instance regression, described later.

"Upgraded" supervised learning algorithms can be used in a multiple-instance setting by suitably modifying their objective functions. Below, we summarize some of the algorithms that have been derived in this way.

1. **Decision Tree induction** algorithms have been adapted to the MI setting [11]. The standard algorithm measures the quality of a split on an attribute by considering the class label distribution in the child nodes produced. In the MI case, this distribution is uncertain, because the true instance labels in positive bags are unknown. However, some rules have been identified that lead to empirically good MI trees: (a) use an asymmetric heuristic that favors early creation of pure positive (rather than negative) leaves, (b) once a positive leaf has been created, remove all other instances of bags covered by this leaf; (c) abandon the depth-first or breadth-first order in which nodes are usually split, adopting a best-first strategy instead (indeed, because of (b), the result of tree learning is now sensitive to the order in which the nodes are split).

2. **Artificial Neural Networks** have been adapted to the MI setting by representing the bag classifier as a network that combines several copies of a smaller network, which represents the instance classifier, with a smooth approximation of the *max* combining function [12]. Weight update rules for a backpropagation algorithm working on this network have been derived. Later work on multiple-instance neural networks has been performed independently by others [13].

3. **Logistic Regression** has been adapted to the MI setting by using it as an instance-based classifier and combining the instance-level probabilities using functions like softmax [14] and arithmetic and geometric averages [15].

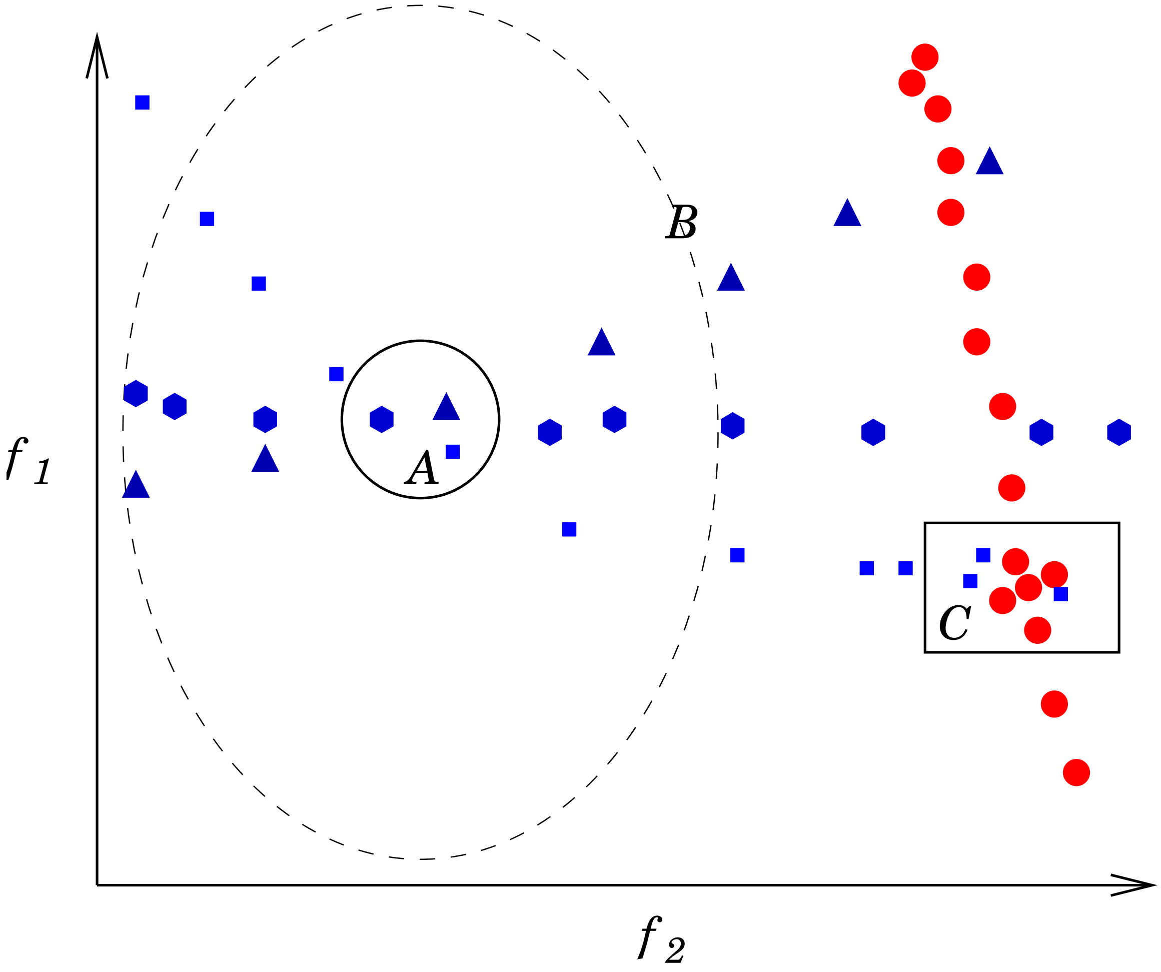

Figure 3: An illustration of the concept that Diverse Density searches for on a simple MI dataset with three positive bags and one negative bag, where each instance (represented by the geometric figures) is described by two attributes, f_1 and f_2. Each type of figure represents one bag, i.e. all triangles belong to one bag, all circles belong to a second bag and so forth. The bag containing the red circles is negative, while the other bags are positive. Region C is a region of high density, because several instances belong to that region. Region A is a region of high "Diverse Density", because several instances *from different positive bags* belong to that region, and no instances from negative bags are nearby. Region B shows a concept that might be learned if the learning algorithm assumed that all instances in every positive bag are positive. Figure adapted from Maron (1998) .

4. The **k-Nearest Neighbor** algorithm has been adapted to the MI setting by using set-based distance metrics, such as variants based on the Hausdorff distance. However, this alone does not solve the problem – it is possible for a positive bag to be mistakenly classified negative if it contains a "true negative" instance that happens to be much closer to negative instances in other negative bags. To solve this, a "Citation-kNN" [16] approach has been proposed that also considers, for each bag B, the labels of those bags for which B is a nearest neighbor.

5. **Support Vector Machines** have been adapted to the MI setting in several ways. In one method, the constraints in the quadratic program for SVMs is modified to account for the fact that certain instance labels are unknown but have constraints relating them [17]. In another method, new kernels are designed for MI data by modifying standard supervised SVM kernels [18] or designing new kernels [19]. The modification allows these MI kernels to distinguish between positive and negative bags if the supervised kernel could distinguish between ("true") positive and negative instances.

6. **Rule learning algorithms** have been adapted to the MI setting in two ways. One method has investigated upgrading a supervised rule-learner, the RIPPER system [20], to the MI setting by modifying its objective function to account for bags and addressing several issues that resulted. Another method has investigated using general purpose relational algorithms, such as FOIL [21] and TILDE [22], and providing them with an appropriate inductive bias so that they learn MI concepts. Further, it has been observed that techniques from multiple instance learning can also be used inside relational learning algorithms [23].

A large scale empirical analysis of several such propositional supervised learning algorithms and their MI counterparts has been performed [14]. This analysis concludes that: (a) no single MI algorithm works well across all problems. Thus, different inductive biases are suited to different problems, (b) some multiple-instance algorithms consistently perform better than their supervised counterparts but others do not (hence for these biases there seems room for improvement), and (c) assigning a larger weight to false positives than to false negatives is a simple but effective method to adapt supervised learning algorithms to the multiple-instance setting. It was also observed that the advantages of multiple instance learners may be more pronounced if they would be evaluated on the task of labeling individual instances rather than bags.

Along with "upgrading" supervised learning algorithms, a **theoretical analysis of supervised learners** learning with MI data has been carried out [24]. In particular, the multiple-instance problem has been related to the problem of learning in the presence of classification noise (i.e. each training example's label is flipped with some probability $< 1/2$). This implies that any concept class that is PAC-learnable in the presence of such noise is also learnable in the multiple-instance learning model when each instance of a bag is drawn iid. Since many concept classes are learnable under this noise assumption (using e.g. *statistical queries* [25]), Blum and Kalai's result implies PAC learnability of many concept classes. Further, they improved on previous learnability results [5] by reducing the number of training bags required for PAC learning by about a factor of n with only an increase in time complexity of about $\log \log n/\epsilon$.

Besides these positive results, a **negative learnability result** describing when it is hard to learn concepts from multiple-instance data is also known [5]. Specifically, if the instances of each bag are allowed collectively to be generated according to an arbitrary distribution, learning from multiple-instance examples is as hard as PAC-learning disjunctive normal form (DNF) formulas from single-instance examples, which is an open problem in learning theory that is believed to be hard. Further, it has been showed that if an efficient algorithm exists for the non-iid case that outputs as its hypothesis an axis-parallel rectangle, then NP = RP (Randomized Polynomial time, see e.g. Papadimitriou [26]), which is very unlikely.

Learning from structured multiple-instance data has received some attention [27]. In this work, each instance is a graph, and a bag is a set of graphs (for example, a bag could consist of certain subgraphs of a larger graph). To learn concepts in this structured space, the authors use a modified form of the Diverse Density algorithm discussed above. As before, the concept being searched for is a point (which corresponds to a graph in this case). The main modification is the use of the size of the maximal common subgraph to estimate the probability of a concept—i.e., the probability of a concept given a bag is estimated as proportional to the size of the maximal common subgraph between the concept and any instance in the bag.

6.2 Multiple-Instance Regression

Regression problems in an MI setting have received less attention than the classification problem. Two key directions have been explored in this setting. One direction extends the well-known standard linear regression method to the MI setting. The other direction considers extending various MI classification methods to a regression setting.

In **Multiple-Instance Linear Regression** [28] (referred to as multiple-instance regression in the cited work), it is assumed that the hypothesis underlying the data is a linear model with Gaussian noise on the value of the dependent variable (which is the response). Further, it is assumed that it is sufficient to model one instance from each bag, i.e. that there is some *primary* instance which is responsible for the real-valued label. Ideally, one would like to find a hyperplane that minimizes the squared error with respect to these primary instances. However, these instances are unknown during training. The authors conjecture that, given enough data, a good approximation to the ideal is given by the "best-fit" hyperplane, defined as the hyperplane that minimizes training set squared error by fitting one instance from each bag such that the response of the fitted instance most closely matches the bag response. This conjecture will be true if the non-primary instances are not a better fit to a hyperplane than the primary instances. However, exactly finding the "best-fit" hyperplane is intractable. It is shown that the decision problem "Is there a hyperplane which perfectly fits one instance from each bag?" is NP-complete for arbitrary numbers of bags, attributes and at most three instances per bag. Thus, the authors propose an approximation algorithm which iterates between choosing instances and learning linear regression models that best fit them, similar to the EM-DD algorithm described earlier.

Another direction has explored **extending MI classification algorithms** to the regression setting. This approach [29] uses algorithms like Citation-kNN and Diverse Density to learn real-valued concepts. To predict a real value, the approach uses the average of the nearest neighbor responses or interprets the Gaussian "probability" as a real number for Diverse Density.

Recent work has analyzed the Diverse Density-based regression in the *on-line* model [30, 31]. In the on-line model, learning proceeds in *trials*, where in each trial a single example is selected adversarially and given to the learner for classification. After the learner predicts a label, the true label is revealed and the learner incurs a *loss* based on whether its prediction was correct. The goal of the on-line learner is to minimize loss over all trials. On-line learning is harder than PAC learning in that there are some PAC-learnable concept classes that are not on-line learnable.

In the regression setting above [32], there is a point concept, and the label of each bag is a function of the distance between the concept and the point in the bag closest to the target. It is shown that similar to Auer et al.'s lower bound, learning in this setting using labeled bags alone is as hard as learning DNF. They then define a *multiple-instance membership query* (MI-MQ) in which an adversary defines a bag $B = \{p_1, \ldots, p_n\}$ and the learner is allowed to ask an oracle for the label of bag $B + \vec{v} = \{p_1 + \vec{v}, \ldots, p_n + \vec{v}\}$ for any d-dimensional vector $\vec{v}$. Their algorithm then uses this MI-MQ oracle to on-line learn a real-valued multiple-instance concept in time $O(dn^2)$.

7 Applications

In this section, we describe domains where multiple-instance learning problems have been formulated.

Drug activity was the motivating application for the multiple instance representation [1]. Drugs are typically molecules that fulfill some desired function by binding to a target. In this domain, we wish to predict how strongly a given molecule will bind to a target. Each molecule is a three-dimensional entity and takes on multiple shapes or *conformations* in solution. We know that for every molecule showing activity, at least one its low energy conformations possesses the right shape for interacting with the target. Similarly, if the molecule does not show drug-like activity, none of its conformations possess the right shape for interaction. Thus, each molecule is represented as a bag, where each instance is a low energy conformation of the molecule. A well-known example from this domain is the MUSK dataset. The positive class in this data consists of molecules that smell "musky". This dataset has two variants, MUSK1 and MUSK2, both with similar numbers of bags, with MUSK2 having many more instances per bag.

Content Based Image Retrieval is another domain where the MI representation has been used [7, 33]. In this domain, the task is to find images that contain objects of interest, such as tigers, in a database of images. An image is represented by a bag. An instance in a bag corresponds to a segment in the image, obtained by some segmentation technique. The underlying assumption is that the object of interest is contained in (at least) one segment of the image. For example, if we are trying to find images of mountains in a database, it is reasonable to expect most images of mountains to have certain distinctive segments characteristic of mountains. A multiple-instance learning algorithm should be able to use the segmented images to learn a concept that represents the shape of a mountain and use the learned concept to collect images of mountains from the database.

The **identification of protein families** has been framed as a multiple-instance problem [19]. The objective in that work is to classify given protein sequences according to whether they belong to the family of thioredoxin-fold proteins. The given proteins are first aligned with respect to a motif that is known to be conserved in members of the family. Each aligned protein is represented by a bag. A bag is labeled positive if the protein belongs to the family, and negative otherwise. An instance in a bag corresponds to a position in a fixed length sequence around the conserved motif. Each position is described by a vector of attributes; each attribute describes a properties of the amino acid at that position, and smoothed using the same properties from its neighbors.

Text Categorization is another domain that has used the MI representation [17, 14]. In this domain, the task is to classify a document as belonging to a certain category or not. Often, whether the document belongs to the specified category is the function of a few passages in the document. These passages are however not labeled with the category information. Thus, a document could be represented as a set of passages. We assume that each positive document (i.e., that belongs to the specified category) has at least one passage that contains words that indicate category membership. On the other hand, a negative document (that does not belong to the category) has no passage that contain words indicating category membership. This formulation has been used to classify whether MEDLINE documents should be annotated with specific MeSH terms [17] and to determine if specific documents should be annotated with terms from the Gene Ontology [14].

Time-series data from hard drives have been used to define a multiple-instance problem [34]. The task here is to distinguish drives that fail from others. Each hard drive is a bag. Each instance in the bag is a fixed-size window over timepoints when the drive's state was measured using certain attributes. In the training set, each drive is labeled according to whether it failed during a window of observation. An interesting aspect to prediction in this setting is that it is done on-line, i.e. the algorithm learns a classifier for instances, which is applied to each instance as it becomes available in time. The authors learn a naïve Bayes model using an EM-based approach to solve this problem.

Discovering useful subgoals in reinforcement learning has been formulated as a multiple-instance problem [35]. Imagine a robot has to get from one room to another by passing through a connecting door. If the robot knew of the existence of the door, it could decompose the problem into two simpler subproblems, to be solved separately: getting from the initial location in the first room to the door, and then getting from the door to its destination. How could the robot discover such a "useful subgoal"? One approach formulates this as a multiple-instance problem. Each trajectory of the robot, where the robot starts at the source and then moves for some number of time steps, is considered to be a bag. An instance in a bag is a state of the world, that records observations such as, "is the robot's current location a door?" Trajectories that reach the destination are positive, while those that do not are negative. Given this data, we can learn a classifier that predicts which states are more likely to be seen on successful trajectories than on unsuccessful ones. These states are taken to be useful subgoals. In the previous example, the MI algorithm could learn that the state "location is a door" is a useful subgoal, since it appears on all successful trajectories, but infrequently on unsuccessful ones.

8 Future Directions

Multiple-instance learning remains an active research area. One direction that is being explored relaxes the "Constraints" in Figure 2 in different ways [36, 19]. For example, one could consider constraints where at least a certain number (or fraction) of instances have to be positive for a bag to be labeled positive. Similarly, it may be the case that a bag is labeled positive only if it does not contain a specific instance. Such relaxations are often studied as "generalized multiple-instance learning."

One such generalization of multiple-instance learning has been formally studied under the name "geometric patterns." In this setting, the target concept consists of a collection of axis-parallel rectangles, and a bag is labeled positive if and only if (1) each of its points lies in a target APR, and (2) every target APR contains a point. Noise-tolerant PAC algorithms [37] and on-line algorithms [38] have been presented for such concept classes. These algorithms make no assumptions on the distribution used to generate the bags (e.g. instances might not be generated by an iid process). This does not violate Auer et al.'s lower bound since these algorithms do not scale with the dimension of the input space.

Another recent direction explores the connections between multiple-instance and semi-supervised learning. Semi-supervised learning generally refers to learning from a setting where some instance labels are unknown. Multiple instance learning can be viewed as one example of this setting. Exploiting this connection between multi-instance learning and other methods for semi-supervised learning, recent work [39] proposes an approach where a multiple-instance problem is transformed into a semi-supervised learning problem. An advantage of the approach is that it automatically also takes into account unlabeled bags.

9 Cross References

Artificial Neural Network, Attribute, Classification, Data Set, Decision Trees, Expectation-Maximization, First-order Rule, Gaussian Distribution, Inductive Logic Programming, Kernel Methods, Linear Regression, Nearest Neighbor, Noise, On-line Learning, PAC Learning, Relational Learning, Supervised Learning

References

[1] Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence **89**(1-2) (1997) 31–71

[2] DeRaedt, L.: Attribute-value learning versus inductive logic programming: The missing links. In: Proceedings of the 8th International Conference on Inductive Logic Programming, Springer Verlag (1998) 1–8

[3] Valiant, L.G.: A theory of the learnable. Commun. ACM **27**(11) (1984) 1134–1142

[4] Long, P.M., Tan, L.: PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. Machine Learning **30**(1) (1998) 7–21

[5] Auer, P., Long, P.M., Srinivasan, A.: Approximating hyper-rectangles: learning and pseudorandom sets. Journal of Computer and System Sciences **57**(3) (1998) 376–388

[6] Auer, P.: On learning from multi-instance examples: Empirical evaluation of a theoretical approach. In: Proc. 14th International Conference on Machine Learning, Morgan Kaufmann (1997) 21–29

[7] Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In Jordan, M.I., Kearns, M.J., Solla, S.A., eds.: Advances in Neural Information Processing Systems. Volume 10., The MIT Press (1998)

[8] Maron, O.: Learning from Ambiguity. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA (1998)

[9] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA (1988)

[10] Zhang, Q., Goldman, S.: EM-DD: An improved multiple-instance learning technique. In: Advances in Neural Information Processing Systems. (2001) 1073–1080

[11] Blockeel, H., Page, D., Srinivasan, A.: Multi-instance tree learning. In: Proceedings of 22nd International Conference on Machine Learning, Bonn, Germany. (2005) 57–64

[12] Ramon, J., DeRaedt, L.: Multi instance neural networks. In: Proceedings of ICML-2000 workshop on Attribute-Value and Relational Learning. (2000)

[13] Zhou, Z.H., Zhang, M.L.: Neural networks for multi-instance learning. In: Proceedings of the International Conference on Intelligent Information Technology. (2002)

[14] Ray, S., Craven, M.: Supervised versus multiple-instance learning: An empirical comparison. In: Proceedings of the 22nd International Conference on Machine Learning, ACM Press (2005) 697–704

[15] Xu, X., Frank, E.: Logistic regression and boosting for labeled bags of instances. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Sydney, Springer-Verlag (2004) 272–281

[16] Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (2000) 1119–1125

[17] Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In Becker, S., Thrun, S., Obermayer, K., eds.: Advances in Neural Information Processing Systems. Volume 15. MIT Press (2003)

[18] Gartner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In Sammut, C., Hoffmann, A., eds.: Proceedings of the 19th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (2002) 179–186

[19] Tao, Q., Scott, S.D., Vinodchandran, N.V.: SVM-based generalized multiple-instance learning via approximate box counting. In: Proceedings of the Twenty-First International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (2004) 779–806

[20] Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann (1995)

[21] Quinlan, J.R.: Learning logical definitions from relations. Machine Learning **5** (1990) 239–2666

[22] Blockeel, H., De Raedt, L.: Top-down induction of first order logical decision trees. Artificial Intelligence **101**(1-2) (1998) 285–297

[23] Alphonse, E., Matwin, S.: Feature subset selection and inductive logic programming. In: Proceedings of the 19th International Conference on Machine Learning. (2002) 11–18

[24] Blum, A., Kalai, A.: A note on learning from multiple-instance examples. Machine Learning Journal **30**(1) (1998) 23–29

[25] Kearns, M.: Efficient noise-tolerant learning from statistical queries. Journal of the ACM **45**(6) (1998) 983–1006

[26] Papadimitriou, C.: Computational Complexity. Addison-Wesley (1994)

[27] McGovern, A., Jensen, D.: Identifying predictive structures in relational data using multiple instance learning. In: Proceedings of the 20th International Conference on Machine Learning. (2003) 528–535

[28] Ray, S., Page, D.: Multiple instance regression. In: Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, Morgan Kaufmann (2001)

[29] Dooly, D.R., Zhang, Q., Goldman, S.A., Amar, R.A.: Multiple-instance learning of real-valued data. Journal of Machine Learning Research **3** (2002) 651–678

[30] Angluin, D.: Queries and concept learning. Machine Learning **2**(4) (1988) 319–342

[31] Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning **2**(4) (1988) 285–318

[32] Dooly, D.R., Goldman, S.A., Kwek, S.S.: Real-valued multiple-instance learning with queries. Journal of Computer and System Sciences **72**(1) (2006) 1–15

[33] Zhang, Q., Yu, W., Goldman, S., Fritts, J.: Content-based image retrieval using multiple-instance learning. In: Proceedings of the Nineteenth International Conference on Machine Learning, Morgan Kaufmann (2002) 682–689

[34] Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Machine learning methods for predicting failures in hard drives: A multiple-instance application. Journal of Machine Learning Research **6** (2005) 783–816

[35] McGovern, A., Barto, A.G.: Automatic discovery of subgoals in reinforcement learning using diverse density. In: Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 361–368

[36] Weidmann, N., Frank, E., Pfahringer, B.: A two-level learning method for generalized multi-instance problems. In: Proceedings of the European Conference on Machine Learning. (2003) 468–479

[37] Goldman, S.A., Scott, S.D.: A theoretical and empirical study of a noise-tolerant algorithm to learn geometric patterns. Machine Learning **37**(1) (1999) 5–49

[38] Goldman, S.A., Kwek, S.K., Scott, S.D.: Agnostic learning of geometric patterns. Journal of Computer and System Sciences **6**(1) (2001) 123–151

[39] Rahmani, R., Goldman, S.A.: Missl: Multiple-instance semi-supervised learning. In: Proceedings of the 23rd International Conference on Machine Learning. (2006) 705–712