Learning Curves for the Analysis of Multiple Instance Classifiers

David M.J. Tax and Robert P.W. Duin

Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

Abstract. In Multiple Instance Learning (MIL) problems, objects are represented by a *set* of feature vectors, in contrast to the standard pattern recognition problems, where objects are represented by a single feature vector. Numerous classifiers have been proposed to solve this type of MIL classification problem. Unfortunately only two datasets are standard in this field (MUSK-1 and MUSK-2), and all classifiers are evaluated on these datasets using the standard classification error. In practice it is very informative to investigate their learning curves, i.e. the performance on train and test set for varying number of training objects. This paper offers an evaluation of several classifiers on the standard datasets MUSK-1 and MUSK-2 as a function of the training size. This suggests that for smaller datasets a Parzen density estimator may be preferrer over the other 'optimal' classifiers given in the literature.

1 Introduction

In many real-world classification problems objects cannot easily be represented by a single unique feature vector, because the objects are too rich and contain too many details and information. A typical example is the problem of image database retrieval. Each image can be of different size and can depict several physical objects in the same picture. In Multi(ple) Instance Learning (MIL) the standard pattern recognition assumption of having one feature vector per object is extended. The objects are represented by a collection (called a 'bag') of feature vectors (called 'instances'). In the training phase only bags are labeled (in a positive and negative class), but not the individual instances. A new bag has to be classified based on the collection of instances in that bag.

An example would be a medical classification problem: for the detection of abnormalities in the lung an X-ray image of the chest can be made. From this image small image patches can be extracted for which features can be derived. When each individual patch would have been labeled, an anomaly detector could have been trained. Although it is possible to classify a patient as healthy or ill, it is very hard to label each patch reliably. This therefore results in a multi-instance learning problem, where the collection (or bag) of patches (instances) is labeled in 'healthy' of 'ill', but where the individual instances are not. A patient is now classified as 'healthy' when none of the instances are classified as 'ill'.

Because objects are now represented by a collection of feature vectors, MIL datasets tend to be large and the classification problem difficult. Unfortunately,

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 724–733, 2008.

the effective sample size in MIL datasets is often limited. Not the instances have to be classified, but the bags. It is therefore not the number of instances, but the number of bags that is the determining factor for the performance of the classifier. Furthermore, these classifiers are often evaluated based on an error measure like classification error (or misclassification rate [11]). But for these low-sample-size problems, and problems where class priors and misclassification costs are unknown, the classification error may not be the most suitable. Performance measures like the Area under the ROC curve [2] have shown to be more reliable for small sample sizes.

Furthermore, the choice for the best classifier for a particular problem is often dependent on the sample size, i.e. the size of the training set. When more training samples are available, in general more complex classifiers can be trained. Reducing the training set size may result in a different (more simple) optimal classifier. It is therefore very informative to inspect the performance of classifiers for varying training set sizes.

In this paper we re-evaluate several MIL classifiers on the two standard datasets, MUSK-1 and MUSK-2. These two datasets from [6] are basically the only datasets that are consequently used in benchmarking of all MIL classification systems ([13,16,10,19,1,20,17,3,9,4] to name a few). To assess the relative performance of the methods over a wide range of sample sizes, we use the learning curve with the Area under the ROC curve. In section 2 we discuss a range of MIL classifiers, followed by a short introduction on learning curves and performance measures in section 3. In section 4 the results of the MIL classifiers is shown, and we summarize the conclusions in section 5.

2 Multi-instance Learning, Classifiers and Datasets

Assume that object i is represented by a bag B_i, containing a set of m_i instances $\mathbf{x}_j \in \mathbb{R}^p$: $B_i = \{\mathbf{x}_{ij}, j = 1, .., m_i\}$ (i.e. feature vectors of length p). Assume further that each bag is labeled with a positive or negative label: $Y_i \in \{\omega^+, \omega^-\}$. The label Y_i that will be assigned to a bag B_i is in principle determined by the number of instances $\mathbf{x}_{ij}$ that is positive. In the most extreme case a bag may be labeled positive when a single instance is classified as positive. When the classification is noisy, or a very large number of instances is present in each bag, a minimum number or a certain fraction of the total number of instances may be chosen. During training the label for each instance y_{ij} is unknown. A classifier has to determine which instances in each bag are informative for the bag label. Finally, a trained classifier has to estimate the class label from a bag of instances:

$$\widehat{Y} = f(B) = f(\{\mathbf{x}_j\}) \tag{1}$$

For the experiments in this paper we use a selected set of multi-instance learners. Due to space constraints not all classifiers can be discussed. The classifiers that are used to classify the multi-instance datasets are

Simple classifiers with bag combiner can be used when the multi-instance learning aspect of the learning problem is ignored. In training all instances are labeled according to their bag label. In the evaluation the classification outcomes of all the instances of a bag are combined to a single bag-outcome. One possible combining rule is the quantile combining. Assume that the (real) outcomes $o_j = f(\mathbf{x}_{ij})$ are sorted: $\boldsymbol{o}_i = \{o_1, o_2, ..., o_{N_i}\}$, where $o_1 \geq o_2 \geq ...o_{N_i}$. The quantile combining rule selects the q-th quantile value of this set:

$$\mathcal{Q}(\boldsymbol{o}_i) = o_n, \quad n = \lfloor qN_i \rfloor + 1 \tag{2}$$

This idea is similar to [3] where the bag-level classifier f_g is this quantile function.

In this paper we applied this combining rule to the outputs of the linear discriminant analysis (LDA) and the Parzen density estimator. The LDA is regularized by adding a small constant $\lambda = 10^{-6}$ to the diagonal. The Parzen density estimator fits a single Gaussian distribution with a fixed width parameter on each individual training sample. The width parameter in the Parzen density is optimized in a leave-one-out fashion on the training set [8]. In the quantile combination $q = 0.01$ is used, which means that for small bags only the presence of a single positive object is sufficient to classify the bag as positive.

Axis-parallel Rectangle [6] constructs a rectangular decision boundary, aligned with the feature axis, that is optimized such that at least one instance of each positive bag falls inside this box but such that none of the instances of the negative bags is inside. Three optimization schemes have been proposed, and in this paper the 'inside-out' scheme is used: it starts from a seed point and then grows a rectangle until it covers at least one instance per positive bag and no instances from negative bags. It also includes the possibility to select features such that in some feature dimensions no box face is defined. The implementation of the APR in this paper is based on [18], and the parameters are tuned to get a good average performance on the MUSK datasets.[1]

Citation k-NN [16] is a variant of the standard k nearest neighbor classifier. The standard k-NN is extended by considering not only the nearest bags on the training set, but also the training bags for which the test bag is the nearest neighbor. These are the so-called 'referees', or 'citers'. (See for a more complete explanation [16].) In this paper the distances between bags is computed using the Hausdorff distance, and $k = 5$ nearest neighbors and $r = 5$ citers are used.

Diverse Density [13] estimates the density of the co-occurrence of instances of positive bags, and corrects for the density of the negative instances. Areas of high diverse density contain at least one instance of all positive bags, but do not contain any negative instances. The diverse density method optimizes a location and a radius (basically, it models a Gaussian distribution) for which the diverse density is highest. When a new bag contains instances in this high diverse density area, it is classified as a positive bag.

Although the reported performances are very good, the optimization procedure of finding the location of a high diverse density area is computationally very intensive, and often local optima are reached. Therefore the procedure is

[1] For a full description of the algorithm, please see [6]. The chosen parameters are: the threshold distance in selecting significant features=1, $\tau = 0.995$, $\epsilon = 0.02$ and step size expanding the APR=0.1.

rerun several times with different initializations. In this paper we tried 10 different random initializations and used the solution which yielded the highest likelihood on the training set. Further parameter settings include the maximum number of epochs that the diverse density is optimized ($4 \times p$), and the tolerance in the change in position or change in likelihood during the optimization (here we used 10^{-5} and 10^{-7}).

EM-Diverse Density [19] is an extension of the Diverse Density approach. Here the highest diverse density position is optimized in an iterative Expectation-Maximization scheme. At each step of the EM scheme only the instance with the highest diverse density per bag is used. This simplifies the estimation of the diverse density, but because the estimated location of the maximum diverse density changes during the optimization, also the instance with the highest diverse density per bag changes each iteration. In practice it appears that this scheme is a bit more insensitive to local minima. The settings of this classifier is identical to these of the Diverse Density method.

MIL-Support vector classifier [10] is a standard support vector classifier that uses a multi-instance kernel to define a similarity between two bags of instances. In this paper very simple bag statistics are computed:

$$\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}(B_i) = [\min_j(\mathbf{x}_{ij}), \max_j(\mathbf{x}_{ij})] \tag{3}$$

This represents a single bag by one feature vector of length $2p$, where the minimum and maximum feature values per bag are stored. Notice that (3) does not compute a kernel, strictly speaking. On this feature vector standard kernels are computed, for instance the standard linear kernel $K(B_i, B_j) = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j$. In this paper the linear kernel, with a trade-off parameter $C = 1$ is used.

Table 1. The MUSK datasets and some characteristics

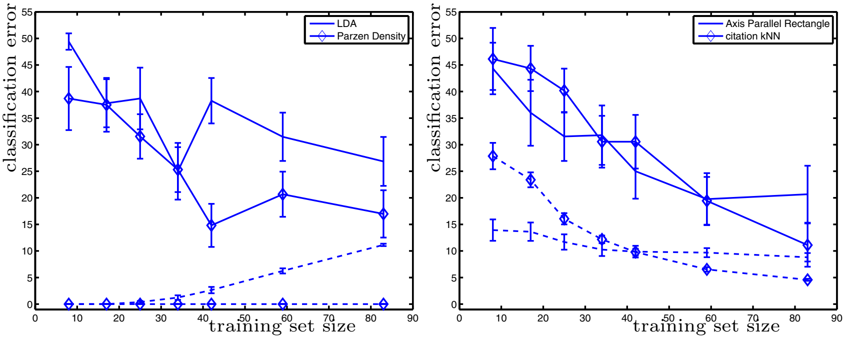
name	pos. bags	neg. bags	# instances	# features
MUSK-1	47	45	476	166
MUSK-2	39	63	698	166

The standard datasets in multi-instance learning are the MUSK-1 and MUSK-2 datasets. The task is to predict if a certain molecule can bind to a target molecule. The molecule is in fact a 3D object, and it can have multiple shapes (or conformations). When a molecule binds to a target, we know that at least one of the conformations has the ability to bind. When a molecule does not bind, none of the conformations have the correct shape. Each molecule can therefore described by a bag of conformations. In these datasets the 3D shape of each conformation is described by 166 features, and conformations of around 80-100 molecules are available. Dataset MUSK-1 contains on average 6 conformations per bag, while MUSK-2 has more than 60 conformations per bag. The task for these datasets is to predict on the basis of the collection of conformations if the molecule smells 'musky' or not. More specific information of the datasets are given in Table 1 and in [6].

3 Learning Curves and Performance Measures

Typically, a classification system is trained on a training set that is as large as possible, and evaluated on an independent test set [11]. It is expected that more training data results in a better performance of the classifier, because with more data the parameters of a classifier can be estimated more reliably. Often a cross-validation scheme is used where the data is split in M parts. $M-1$ parts are used for training, and the rest is used for testing. This is repeated M times and the performance is averaged. The classifier is thus trained M times on $M/(M-1)$ part of the data, and all data is used once for estimating the true error.

A learning curve shows the change in (true) classification error for a varying training set size. Often not only the true error is estimated, but also the apparent error, i.e. the error on the training set. When the difference between the apparent error and the true error is large, the classifier is called overtrained, or overfitted. The performance on the training set gives a too optimistic estimate on what can be expected in practice.



(a) Learning curves LDA and Parzen. (b) Learning curves APR and k-NN.

Fig. 1. Learning curves for the MUSK-1 dataset using the classification error

Figure 1(a) presents some typical learning curves, showing the true and apparent error rate for the MUSK-1 dataset by the solid and dashed lines respectively. (This dataset and the classifiers are explained in section 2.) The training set size is a fraction of the total training size, running from 10% (around 10 bags, smaller numbers caused problems for some classifiers) to 99% (84 bags). In the left subfigure the LDA and Parzen classifiers are shown. For both classifiers a significant difference between the apparent and true error is visible. Both classifiers perfectly fit the training set, and only the training error of the LDA deteriorates slightly for sample sizes larger than 40. This suggest that this dataset is too small for these classifiers and that the classifiers overtrain.

One can also extrapolate if classifiers may gain significantly in performance when more training data is added. A flat learning curve (like the one from the Parzen classifier, although it is a bit noisy) suggests that the classifier is already trained well, and that more training data will not help the classifier much. A more steeply decreasing learning curve (like the one from the LDA) suggests that a bit more data may be very beneficial for this classifier.

This general picture does not accommodate some special cases. For instance, the nearest neighbor classifier always has a positive bias in the evaluation of the training set. Because it is using the training set as the classifier model, evaluating the k nearest neighbors in the training set will always include a copy of the test object. This is visible in Figure 1(b). The training performance of the Citation k-NN (which is a special version of the standard k-NN classifier) *increases* for larger training set sizes. Still a positive bias is present over all training set sizes. The slope in the learning curve also suggests that a bit more data may be very beneficial for the k-NN performance.

The learning curves for the Axis Parallel Rectangle classifier shows a similar characteristic. Here the apparent error also decreases (slightly) with increasing sample size. This is caused by the fact that the parameter settings in this method are fixed to values that are more suitable for larger training sets. In particular the parameter τ, that specifies how far the rectangle boundaries should be placed around the training instances, is a sensitive parameter. For smaller sample sizes the decision boundary has to be set a bit wider than for larger sample sizes. Optimizing this parameter by cross-validation appears to be a very costly and gives very noisy results.

For the evaluation of a classifier often the classification error is used. It just counts the number of misclassifications in a (test) set. Assume a classifier f is trained for a two-class classification problem, and evaluated on a test set $\mathcal{X} = \{(\mathbf{x}_i, y_i), i = 1, .., N\}$ where $\mathbf{x}_i \in \mathbb{R}^p$ are p-dimensional feature vectors, and $y_i \in \{\omega^+, \omega^-\}$ are class labels. The classification error is estimated by: $\hat{\epsilon} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{I}(f(\mathbf{x}) \neq y_i)$ where $\mathcal{I}(.)$ is the indicator function that outputs 1 when the statement is true and 0 otherwise.

A drawback of this measure is that it is sensitive to the class priors and that it does not take misclassification costs into account. Often the class priors and costs are unknown, and the empirical class priors are used. Especially when the data sampling is very skewed, deceiving performances are obtained. For these situations the area under the ROC curve (AUC) is more suitable (for MIL problems it is used in [14] for instance). It basically estimates the probability that a classifier gives a higher output for an object of the positive class than an object from the negative class [2]. It can be estimated by:

$$A\hat{U}C = \frac{1}{N^+N^-} \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \mathcal{I}(f(\mathbf{x}_i) > f(\mathbf{x}_j)) \tag{4}$$

where N^+ and N^- are the number of objects from the positive and negative class respectively. By this relative comparison this measure becomes independent of the class priors or class sampling. Furthermore, by the fact that it incorporates all possible pairs of positive and negative objects, the AUC tends to be a more stable performance estimator than the standard classification error [5] making it also easier to compare classifiers [12,15].

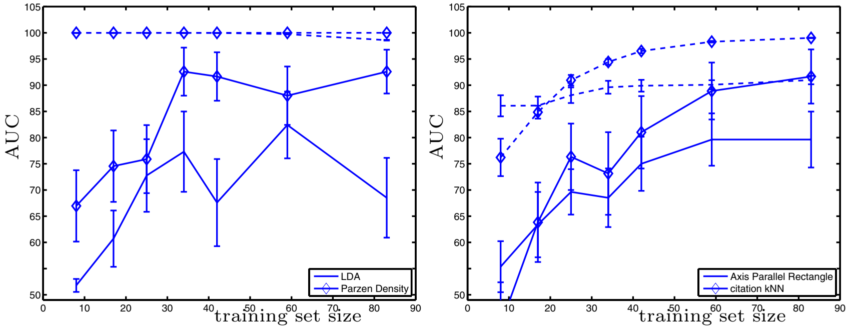
4 Experiments

In this section we evaluate all the classifiers on some datasets. We applied 10-fold stratified cross-validation on the bags, and varied the training set size from 10% to 99% of the total design set. All classifiers are implemented in the pattern recognition toolbox Prtools [7]. The implementations of the APR, the maximum Diverse Density and the EM-Diverse density are based on [18].

In figure 2 the learning curves for the same classifiers on *exactly* the same train and test sets as in figure 1 are shown, but instead of the classification error the AUC (×100) is used. First notice that a well performing classifier has a *low* classification error but a *high* AUC. The general trends are the same, but some subtle changes can be observed. Figure 1 using the classification error suggests that the performance for the Parzen classifier is slowly, but uniformly increasing. Figure 2 shows that the performance in terms of AUC is not improving significantly after $N = 30$. This suggests that around $N = 30$ the density of the target concept is estimated relatively reliably, but that the operating point is still uncertain. Because the AUC only estimates the ranking of the objects, it is not influenced by a poor operating point (or threshold on the density). Furthermore, the left graph in figure 1 suggests that for $N = 30$ bags per class the LDA and Parzen perform similarly. Figure 2 on the other hand suggests that averaged over all operating points the Parzen may still be preferred.

When we compare the final performances of the APR and the Citation k-NN in Figure 1, we observe another typical behavior: for smaller training sets the more simple APR classifier has smaller error. When the training set size is increased to over $N = 60$, the more complex Citation k-NN starts to win. This is even more prominent in figure 2 where the Citation k-NN is already better than the APR for $N = 25$.

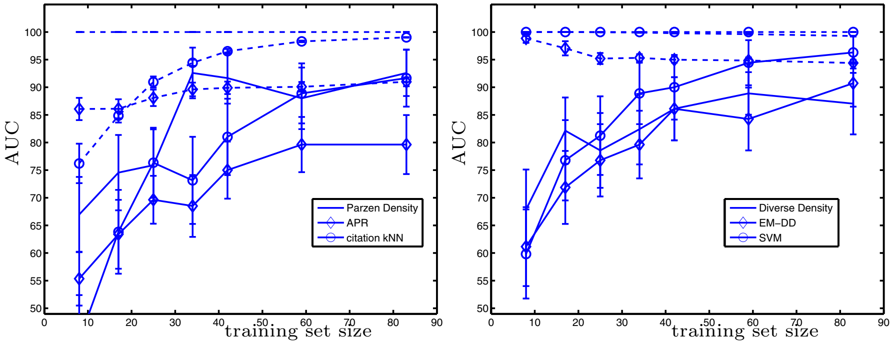
In figures 3 and 4 all the learning curves for all classifiers are shown on datasets MUSK-1 and MUSK-2, respectively. From the results shown in figure 3 it can be concluded that for small sample sizes (upto $N = 50$) the Parzen density performs



(a) Learning curves for LDA and Parzen density classifiers.

(b) Learning curves for the APR and k-NN classifiers.

Fig. 2. Learning curves for the MUSK-1 dataset using the AUC perf. measure (×100). The errorbars indicate one standard deviation of variation.



(a) Learning curves for Parzen density, APR and Citation k-NN

(b) Learning curves for Diverse Density, EM-DD and the support vector classifier

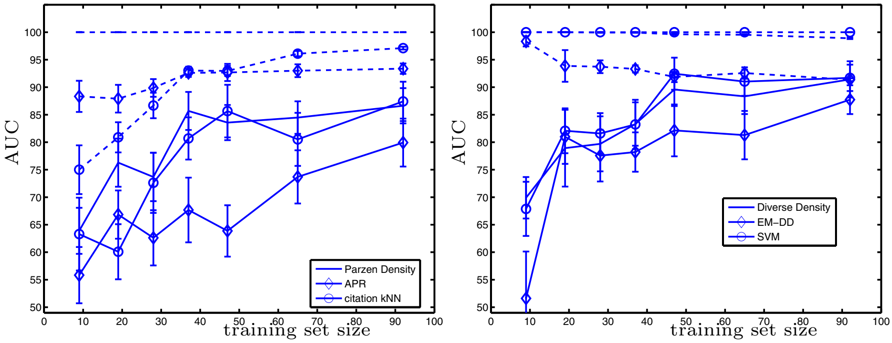
Fig. 3. Learning curves for MUSK-1 using the AUC performance measure

very well. Even for just 8 training bags per class, the AUC is around 0.70. For training sizes around $N = 30$ the Parzen reaches an AUC of more than 0.90, but it does not improve performance significantly for larger training sizes.

The Axis Parallel Rectangle performs relatively poorly in these experiments. The AUC for this method barely reaches 0.75 for the MUSK-1 dataset. The performance for the MUSK-2 dataset is slightly better, around 0.80.

The Citation k-NN and the Support vector classifier perform about equal. Both show promising performance, in that their learning curves are still increasing. This suggest that more training data may benefit the k-NN and SVM most. The very best performance on the MUSK-1 dataset is obtained by the SVM, with an AUC of more than 0.95. This suggest that the simple min, max-feature representation, given in equation (3), characterizes the problem very well.

In figure 4 the same classifiers are applied to the MUSK-2 dataset. This dataset has the same number of bags, but has far more instances per bag. Overall



(a) Learning curves for Parzen density, APR and Citation k-NN

(b) Learning curves for Diverse Density, EM-DD and the support vector classifier

Fig. 4. Learning curves for MUSK-2 using the AUC performance ×100. Errorbars indicate one standard deviation over a 10-fold stratified cross-validation over the bags.

the characteristics are similar, although there are telling differences. The Parzen density estimator requires both in the MUSK-1 and MUSK-2 datasets about 30 bags to obtain a good performance, but Parzen finally obtains a slightly worse performance for larger training set sizes. Most other models, in particular the SVM and the Diverse Density, can exploit the larger number of instances per bag for the smaller sample sizes (around $N = 20$). For the SVM the minimum and maximum feature values can be estimated better, and for the Diverse Density a better estimate for the diverse density per bag can be obtained. For the higher sample sizes this advantage is lost, and the final best performance on the MUSK-2 dataset is just above 0.90, obtained again by the SVM. The EM-DD does not seem to gain too much by increasing the number of instances per bag, its performance increases just slightly after $N = 20$.

5 Conclusions

In real classification problems it is often interesting to see how classifiers behave for varying training set sizes. The resulting curve, the learning curve, shows which classifier is more suitable for small training set sizes, and which classifier has the most promising performance improvement when more data is available. For multi-instance learning problems this is very relevant, because the number of (in particular positive) bags is often limited. By comparing the learning curves of different classifiers for the MUSK-1 and MUSK-2 datasets, suitable models for different training set sizes can be obtained.

The experimental results in this paper suggest that simple classifiers like the Parzen density classifier, or the Support Vector Machine with a simple (but fitting) bag representation work well for both the MUSK datasets. The overall best performance for larger training set sizes are obtained using the SVM (over 0.95 AUC). Its learning curve suggest that even better performances can be obtained when more training data may be available.

The results also show that more instances per bag may not be advantageous. The best classification performances are obtained with the MUSK-1 dataset, that has on average 6 instances per bag, instead of the around 60 instances per bag in MUSK-2. When more instances per bag are present, the search problem to find the informative instance in each positive bag becomes harder. The classifiers therefore have to be trained better to find that single instance that distinguishes a positive from a negative bag. A topic for further research is if the number of instances per class is still important when not a single positive instance is required for labeling a bag positive, but a certain minimum *fraction* of instances. In that case it may be expected that the search problem does not become much harder and that more instances does not directly deteriorate the results.

References

1. Andrews, S., Hofmann, T., Tsochantaridis, I.: Multiple instance learning with generalized support vector machines. In: Proceedings of the AAAI National Conference on Artificial Intelligence (2002)

2. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145–1159 (1997)
3. Cannon, A., Hush, D.: Multiple instance learning using simple classifiers. In: Proceedings of the international conference on machine learning and applications, pp. 123–128 (2004)
4. Chen, Y., Bi, J., Wang, J.Z.: MILES: Multiple-instance learning via embedded instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12), 1931–1947 (2006)
5. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. In: Advances in Neural Information Processing Systems (NIPS 2003) (2004)
6. Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(1-2), 31–71 (1997)
7. Duin, P., Juszcak, R.P.W., Paclik, P., Pekalska, E., de Ridder, D., Tax, D.M.J.: Prtools, a Matlab toolbox for pattern recognition, version 4.0 (January 2004)
8. Duin, R.P.W.: On the choice of the smoothing parameters for Parzen estimators of probability density functions. IEEE Transactions on Computers C-25(11), 1175–1179 (1976)
9. Gao, S., Sun, Q.: A generalized discriminative multiple instance learning for multimedia semantic concept detection. In: ICIP 2006, pp. 2901–2904 (2006)
10. Gärtner, T., Flach, P.A., Kowwalczyk, A., Smola, A.J.: Multi-instance kernels. In: Sammut, C., Hoffmann, A. (eds.) Proceedings of the 19th International Conference on Machine Learning, pp. 179–186. Morgan Kaufmann, San Francisco (2002)
11. Hand, D.J.: Construction and assessment of classification rules. Wiley, New York (1997)
12. Ling, C.X., Huang, J., Zhang, H.: AUC, a better measure than accuracy in comparing learning algorithms. In: Proceedings of the 2003 Canadian artificial intelligence conference (2003)
13. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: Advances in Neural Information Processing Systems, vol. 10, pp. 570–576. MIT Press, Cambridge (1998)
14. Ray, S., Craven, M.: Supervised versus multiple instance learning: an empirical comparison. In: ICML 2005: Proceedings of the 22nd international conference on Machine learning, pp. 697–704. ACM, New York (2005)
15. Rosset, S.: Model selection via the AUC. In: ICML 2004, pp. 703–710 (2004)
16. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Proc. 17th International Conf. on Machine Learning, pp. 1119–1125. Morgan Kaufmann, San Francisco (2000)
17. Xu, X., Frank, E.: Logistic regression and boosting for labeled bags of instances. In: Proc. of the Pacific-Asia conference on knowledge discovery and data mining. Springer, Heidelberg (2004)
18. Zhang, M.-L.: Matlab toolbox: Mil learners and their ensemble versions
19. Zhang, Q., Goldman, S.: EM-DD: An improved multiple-instance learning technique. In: Advances in Neural Information Processing Systems, vol. 14. MIT Press, Cambridge (2002)
20. Zhou, Z.-H., Zhang, M.-L.: Ensembles of multi-instance learners. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS, vol. 2837, pp. 492–502. Springer, Heidelberg (2003)