Optimistic Pruning for Multiple Instance Learning

Amy McGovern [a,*], David Jensen [b]

[a] *School of Computer Science, University of Oklahoma, Norman, OK 73019, USA*

[b] *Computer Science Department, Knowledge Discovery Laboratory, University of Massachusetts Amherst, Amherst, MA 01003, USA*

Abstract

This paper introduces a simple evaluation function for multiple instance learning that admits an optimistic pruning strategy. We demonstrate comparable results to state of the art methods using significantly fewer computational resources.

Key words: Multiple-instance learning, optimistic pruning, chi-squared.
PACS: code, code

1 Introduction

This paper introduces a simple function for evaluating models in multiple instance learning (MIL). This function is the well-known chi-squared statistic, and it admits an optimistic pruning strategy. Current techniques in MIL primarily focus on creating new evaluation functions that are specifically designed for the MIL problem, adapting single-instance learning methods for MIL or a combination of these approaches. The primary contribution of the paper is the demonstration that a simple approach will either outperform or perform comparably to state of the art methods using far fewer computational resources. As Holte (1993) demonstrated for supervised learning, a simple approach provides an important baseline for the complex approaches. Our evaluation function is based on the χ^2 statistic with pruning performed similarly to Clearwater and Provost (1990); Smyth and Goodman (1992); Agrawal and Srikant (1994); Webb (1995); Oates and Cohen (1996). Although we use χ^2, other statistics such as G will also work. We empirically demonstrate the power of optimistic pruning on a variety of tasks introduced in Dietterich et al.

* Corresponding author. Phone: 405-325-5427 Fax: 405-325-4044
Email addresses: amcgovern@ou.edu (Amy McGovern), jensen@cs.umass.edu (David Jensen).

(1997), Maron (1998), Chen and Wang (2004), and Chen et al. (2006). As data grow in complexity, pruning becomes critical for MIL. For example, relational data introduces an explosion in search complexity and the optimistic pruning technique introduced here was the key to ensuring that our search terminated in a reasonable amount of time (McGovern and Jensen, 2003).

MIL is an important method for machine learning given that data for many real world learning tasks are more readily available as labeled *sets* of instances, or *bags*, rather than individually labeled instances (Dietterich et al., 1997; Maron and Ratan, 1998; Zhang et al., 2002). The MIL framework, as introduced by Dietterich et al. (1997), is a framework designed for learning with ambiguous data of this form. Each bag has a single label and it is unknown which instance(s) determined the bag's label. The goal of an MIL agent is to identify a concept that can be used to correctly label the training bags as well as unseen data.

For research reproducibility, we compare our evaluation function with other traditional functions using our own implementation. We compared our evaluation function to diverse density (Maron, 1998), one of the most successful specialized MIL evaluation functions. Diverse density has been used successfully for a number of tasks, including content based image retrieval and discovering abstract actions in reinforcement learning (McGovern, 2002; McGovern and Barto, 2001). We also compare the power of pruning to the more specialized search technique developed by Dietterich et al. (1997). Given the results of Bradford and Brodley (2001) [1] , it is not possible to statistically compare our results to published results such as Ray and Craven (2005) that use only a single run of 10-fold cross validation.

The novelty of our approach is the introduction of a simple evaluation function from supervised learning that admits a guaranteed pruning method. Other researchers have modified supervised learning techniques to work with MIL data although none have introduced a pruning search approach. These include Chevaleyre and Zucker (2001) who modified the RIPPER rule-learning algorithm to learn multiple instance concepts that can be expressed as rules. They use a similar extension to adapting decision trees for use with MIL data (Zucker and Chevaleyre, 2000). Likewise, Ramon and Raedt (2000) adapted the internal labeling methods for neural networks to learn a function that took in real-valued instances and output a label for each instance. Ray and Page (2001) modified regression for MIL by assuming that if an algorithm could identify the best instance in each bag, then straightforward regression could be used to learn a labeling method for each instance (and then for each bag). Gärtner et al. (2002) adapted kernel methods to work with MIL data by modifying the kernel distance measures to handle sets. Using a related approach, Chen and Wang (2004); Chen et al. (2006) adapted SVMs by modify-

[1] Bradford and Brodley (2001) show that comparing the performance of one algorithm to another using a single set of k-fold cross validation results can be problematic and we have observed considerable variance particularly with the standard MUSK data set.

ing the form of the data rather than change the underlying SVM algorithms while Andrews et al. (2003) adapted the SVM kernels directly to produce one of the current best MIL classification systems. Although Ray and Craven (2005) showed that straightforward supervised learning techniques can outperform some specialized MIL approaches, neither approach is a clear winner. In the case where the data are relational, several researchers have proposed modifications to supervised learning methods to accommodate flattened multiple-instance data (Chevaleyre and Zucker, 2001; Zucker and Chevaleyre, 2000). In previous work, we proposed that the data not be flattened and instead that the knowledge representation be modified for use with relational data (McGovern and Jensen, 2003). Because χ^2 admits an optimistic pruning strategy, we use straightforward exhaustive search for MIL.

2 Multiple Instance Learning

At the core level, MIL is designed to solve the same problem as single-instance learning: identify a concept that can be used to correctly classify the training data as well as to generalize to unseen data. Although the two approaches differ in the class labels from which they learn, the actual learning process is quite similar. The algorithms have four components: the knowledge representation used to represent the concept space, the inference method, the search technique used to discover concepts, and the evaluation measure used to compare the different concepts identified by search. We give a brief overview of MIL in this formalism because it clarifies how our evaluation function fits into current MIL approaches.

The data available for an MIL problem are in the form of *bags* where each bag is composed of a set of instances with a single class label for the entire bag. Using the same notation as Maron (1998) and Maron and Lozano-Pérez (1998), a bag, B, is composed of a set of instances, I, and a single label, $L = [0, 1]$ for the bag. That is, $B = \{I_1, I_2, \ldots, I_k\}$ where $|B| = k$ and $L(B)$ is defined. Most MIL algorithms focus on the case where the label can be either positive or negative, $L = \{0, 1\}$ or $L = \{-, +\}$. It is also possible to make use of real-valued labels to indicate a degree of confidence that a positive instance appears in a bag as demonstrated by Amar et al. (2001). By definition, positive bags, $\mathbf{B}^+$, must contain at least one instance of the *target concept* while negative bags, $\mathbf{B}^-$, cannot contain any instances in the target concept. The ith positive bag is denoted as B_i^+ and the jth instance of the ith bag is denoted B_{ij}. The target concept, c, is the true concept that was used to label the bags. This is not generally known in advance nor is it known which instance contributes to a bag being labeled as positive. If each instance was individually labeled, the data would be in the form of a supervised learning problem. Overall, the data available for MIL are in the form $D = \{B_1^+, \ldots, B_m^+, B_1^-, \ldots, B_n^-\}$. The number of instances in each bag is not guaranteed to be homogeneous, that is, $|B_i|$ is not guaranteed to be equal to $|B_j|$.

The knowledge representation defines the concept space for the MIL algorithm. The goal is to identify a concept, c, that correctly labels the bags. The best form of the concept depends on the problem being addressed. In addition to defining the concept space, the knowledge representation component must define a distance metric for measuring the similarity between a concept and an instance. This is denoted $d(c, I)$.

The two knowledge representations that we use in this paper are the axis-parallel rectangle concept (APR), introduced by Dietterich et al. (1997), and the single-point and scaling concept (SPS), introduced by Maron and Lozano-Pérez (1998) and Maron (1998). An APR concept is defined by a hyper-rectangle that is parallel to each feature's axis. Each dimension of the rectangle is individually defined, with bounds indicating the necessary precision for distinguishing a feature. A SPS concept is represented by a point in the feature space and a weight, or scaling, for each feature that determines the relative importance of that dimension.

Given a knowledge representation, a learning algorithm needs an inference method. With MIL, the goal is to identify concepts that label bags. We denote the predicted label of a bag as $\hat{L}(B)$ and the actual label of a bag as $L(B)$. The standard approach, introduced by Dietterich et al. (1997), is to assume a bias towards positive bags. If any instance in the bag is within the concept, the bag is labeled as positive. Otherwise, the bag is labeled as negative. Since a single instance in the bag can bias the label to positive, we denote this as *positive-bias*. It has also been called *single-tuple bias* by Chevaleyre and Zucker (2001). If the labels of the bags can be real-valued, then the positive-bias label of bag B_i with k instances is defined to be: $L(B_i) = \max_k(1 - d(c, B_{ik}))$, where d is the distance function.

An MIL agent using an explicit concept learning approach needs a search method and an evaluation function to identify the highest scoring concept given the training data. Several MIL specific search methods have been proposed. For example, Dietterich et al. (1997) proposed several search methods, such as iterated discrim-APR and GFS-elim APR, that are specific to both the APR concepts and to the MIL problem. For SPS concepts and other concepts where the evaluation measure is differentiable, gradient descent techniques can be used (Maron and Lozano-Pérez, 1998; Maron, 1998). Likewise, Zhang and Goldman (2002) proposed using EM to search. We specifically compare to Dietterich's search approaches.

Although these approaches can work well for MIL, they are either tied to specific properties of the evaluation function (e.g. that it is differentiable) or to a specific knowledge representation (e.g. APR or SPS). With the knowledge that the search approach for MIL need not be specific to MIL itself, we can make use of many other search techniques from AI and single-instance learning. For example, a random search, such as that presented by Rosenstein and Barto (2001), is independent of the specific knowledge representation and has been shown to work successfully with MIL data (McGovern and Barto, 2001; McGovern, 2002; McGovern and Jensen, 2003). Several researchers have proposed heuristics to accelerate the search process. For example, as suggested by Maron (1998), the starting points for search can be limited to concepts near instances in the positive bags. This is a viable heuristic as MIL guarantees that at least one positive instance occurs in each positive bag. No starting points are selected from negative bags because no positive instances can occur in those bags. If search is a bottleneck, it can be further limited by sampling instances from the positive bags (similar to the approach taken by Ray and Page (2001) for use with multiple instance regression).

Although random search can work for any knowledge representation and evaluation function, there are no guarantees that it will find globally maximal solutions. Given either sufficient time to search or an efficient approach to pruning, an exhaustive search approach can be used instead. This is the approach that we take in this paper. However, given the size of the search space, exhaustive search is only feasible with an ability to actively prune large portions of the search. The χ^2 statistic can be used as an evaluation function that admits an optimistic pruning approach. This is discussed in Section 3.

The final component of MIL is model evaluation. This is a critical component of any search process as it allows the agent to evaluate the current model based on the training data. For MIL, this function needs to evaluate a concept given the labeled set of training bags. Diverse density Maron and Lozano-Pérez (1998); Maron (1998) is perhaps the most intuitively appealing evaluation function. The most diversely dense concept is defined to be that which is at the intersection of the positive bags minus the union of the negative bags. The best concept covers as many positive bags as possible while also covering as few negative instances as possible. The next section introduces our proposed model evaluation and search technique.

3 Chi-squared model evaluation and optimistic pruning

This paper demonstrates that a simple evaluation function based on the chi-squared (χ^2) statistic can be quite effective for MIL. The primary advantage of using χ^2 as an evaluation function is that it admits an optimistic pruning approach, which is critical for effective exhaustive search in large spaces. Although many of the traditional MIL problems such as MUSK do not have a large search space, more complex problems such as those involving relational data do.

The χ^2 statistic is calculated based on a contingency table, such as that shown below.

		Actual bag labels	
		+	-
Predicted bag labels	+	True Positives (TP)	False Positives (FP)
	-	False Negatives (FN)	True Negatives (TN)

χ^2 is the sum of the normalized squared differences between the observed and expected frequencies in each entry of the contingency table. That is,

$$\chi^2 = \sum_i \frac{(c_o - c_e)^2}{c_e}$$

where c_o is the observed count in the table and c_e is the expected count. The expected count is calculated by multiplying the row and column margins and dividing by the overall number of records in the table.

For this example, the expected counts for each cell are shown below.

		Actual bag labels	
		+	-
Predicted bag labels	+	$E_{TP} = \frac{(TP+FP)*(TP+FN)}{N}$	$E_{FP} = \frac{(TP+FP)*(FP+TN)}{N}$
	-	$E_{FN} = \frac{(FN+TN)*(TP+FN)}{N}$	$E_{TN} = \frac{(FN+TN)*(FP+TN)}{N}$

Chi-squared is then $\frac{(TP-E_{TP})^2}{E_{TP}} + \frac{(FP-E_{FP})^2}{E_{FP}} + \frac{(FN-E_{FN})^2}{E_{FN}} + \frac{(TN-E_{TN})^2}{E_{TN}}$.

Although this example uses binary labels, real-valued labels can be binned and a larger table can be constructed. To fill in this table, MIL needs a concept c, the bags, $\mathbf{B}^+$ and $\mathbf{B}^-$, and an inference method to label bags. Assuming the positive-bias inference method described in the previous section, each entry in the table would be calculated as shown below.

		Actual bag labels	
		+	-
Predicted bag labels	+	$\sum_{B_i \in \mathbf{B}^+} \hat{L}(B_i)$	$\sum_{B_i \in \mathbf{B}^-} \hat{L}(B_i)$
	-	$\sum_{B_i \in \mathbf{B}^+}(1 - \hat{L}(B_i))$	$\sum_{B_i \in \mathbf{B}^-}(1 - \hat{L}(B_i))$

Under this evaluation function, the best model is the one with the highest chi-squared value. Chi-squared will be maximal in two cases: when the mass is concentrated along the main diagonal (e.g., in TP and TN) and when the mass is concentrated along the off-diagonal (e.g., in FP and FN). In the first case, the proposed concept is correctly predicting a maximum number of positive and negative bags, which is the overall goal. In the second case, the concept is predicting exactly the opposite of this goal. This is a well-known issue with the chi-squared statistic and the signed chi-squared statistic addresses this issue. We define the best concept to have a maximal signed chi-squared value. Signed chi-squared is equal to χ^2 if the mass is on the main diagonal and to $-\chi^2$ if it is on the off-diagonal. The concept with the maximal signed χ^2 value will predict the most positive and negative bags correctly. This differs from other evaluation functions, such as diverse density, that do not separate the effects of correctly predicting each negative bag and instead focus on predicting the positive bags.

3.1 Optimistic Pruning

The primary advantage of using χ^2 as an evaluation function is that it yields an efficient pruning approach by defining the maximum χ^2 value from any point in search. Both a general-to-specific and a specific-to-general search technique can make use of the properties of the χ^2 statistic to identify the best possible evaluation that can be reached from a particular point in a model search given the training bags. With this information, a search method can prune in an optimistic manner. The idea of pruning in this manner is very similar to that of Clearwater and Provost (1990), Smyth and Goodman (1992), Agrawal and Srikant (1994), Webb (1995), and Oates and Cohen (1996). The general ideas apply to other statistics that can be calculated on the contingency tables, such as the G statistic.

Optimistic pruning using χ^2 works as follows. Given a concept c to evaluate and a set of training bags, $\mathbf{B}^+$ and $\mathbf{B}^-$, the contingency table can be filled in as described above. Assume that the values in each cell are:

TP	FP
FN	TN

For general-to-specific search, pruning needs to calculate the maximum χ^2 value for the concept c by examining a concept c' that is based on c but is more specific. For example, a more specific APR concept would have a smaller rectangle along some dimension(s). For a relational concept, this may mean that more vertices and edges have been added to c'. A more specific concept is unable to match *more* bags than the original concept, so mass in the contingency table is restricted to move from the top row (i.e., positive predictions) to the bottom row. Also, since the columns of the table are the actual (and not the predicted) bag labels, mass can not move from one column to another. With these restrictions, a concept c' based on concept c would have a maximal signed χ^2 value when the table contains:

TP	0
FN	FP + TN

The χ^2 statistic on this table is the best possible χ^2 value for concept c. Concepts whose best possible evaluation is worse than the actual best evaluation seen so far can be immediately pruned.

Pruning is also possible for specific-to-general search, where a concept starts very specific (e.g. a rectangle containing a single instance) and is expanded to be more general (e.g. increasing the rectangle along a dimension). For this case, the maximum χ^2 evaluation would occur when the table contains:

TP + FN	FP
0	TN

In both cases, pruning is optimistic, meaning that concepts will never be pruned that should have been examined. This is guaranteed by the rules of both the general-to-specific and the specific-to-general search approaches. By choosing these approaches, the counts in the contingency table can only move from one row to the other which enables us to look ahead to the best possible χ^2 value.

4 Empirical Results: The Power of Pruning

Our empirical results demonstrate that our simple evaluation function and search approach are competitive with current MIL techniques and that they use much fewer computational resorces. We first examine the reduction in search time due to pruning. We then discuss performance on the standard MIL data sets, MUSK 1 and 2 and the standard COREL image classification task (for example, see Chen and Wang, 2004; Chen et al., 2006).

4.1 Ten dimensional environment

For the first set of experiments, we use a simulated ten dimensional environment based on the two-dimensional environment described by Maron (1998). The same results can be duplicated in two dimensions but we chose a ten dimensional environment to increase the difficulty of the task and the size of the search space. There are ten real-valued features $f_i \in [0, 100]$. The target concept is a square of width 5 centered at 50 in each dimension. Instances in each bag are generated by uniformly randomly sampling from $[0, 100]$ in each dimension. If any instance in a bag falls within the target region, the bag is labeled as positive and negative otherwise.

Figure 1 shows an example of this data in 2 dimensions. Instances in the positive bags are labeled with a plus and instances in the negative bags are shown with a dot. The target concept is shown with a rectangle. For this example, there are 20 positive and 20 negative bags and each bag has 25 instances. As the figure shows, this is a difficult concept to identify.

We hypothesized that an agent with the ability to extensively prune its search space would be able to outperform an agent that was not able to prune with equal numbers of search steps. We empirically evaluated this hypothesis using the ten-dimensional environment. All of the experiments used the same underlying code, so the effects are completely attributed to the power of pruning. We used the average area under the ROC curve (AUC) over 30 runs on a separate test set as a performance measure. As this increases to 1, the performance of the system is increasing.

The first experiment focuses on the question of how well each search method and evaluation function perform as a function of the number of search steps. We compared the general-to-specific search to Dietterich's iterated discrimination search. For this experiment, we varied the number of instances per bag but the overall number of bags remained fixed at 20 positive and 20 negative bags. For the general-to-specific search, we measured AUC on the test set every 5 search steps using the best APR concept found so far. We also examine the effects of pruning using the iterated discrimination search method described by Dietterich et al. (1997). Because this algorithm is $O(n^2)$ and the general-to-specific search is $O(n)$, the results at a specific number of search steps cannot be compared across search methods. For the iterated discrimination search, we varied the evaluation function from that proposed by Dietterich et al. (1997) to the χ^2 statistic with ties broken by the evaluation suggested by Dietterich.

Figure 2 compares the average AUC values over 30 runs for the general-to-specific search method with and without pruning. For the general-to-specific search, the advantage of pruning is clear from the beginning and grows as the problem becomes more difficult (as the number of instances in each bag is increased). Although both the pruning and non-pruning approaches converge to the same performance eventually, the ability to prune enables the answer to be found much more quickly.

Figure 3 compares the average AUC using Dietterich's iterated discrimination search method with and without pruning. The same clear difference in performance between pruning versus non-pruning shows up for the iterated discrimination search approach. In this case, the overall performance is lower than with general-to-specific search, likely because this technique is tuned to the MUSK problem. For the last problem, with 100 instances per bag, iterated discrimination search without pruning is not even able to perform above default (0.5 AUC) by 50,000 search steps.

The second experiment varied the knowledge representation to evaluate whether the effects of pruning were a function of the type of concept. For this set of experiments, the search method was a simple general-to-specific search that started from each instance in the positive bags. The evaluation function was the χ^2 statistic, and pruning was either enabled or disabled. We also varied the difficulty of the problem by varying the number of instances placed in each bag. As the size of each bag grows, the problem becomes more difficult because the signal-to-noise ratio decreases. The number of bags was fixed at 20 positive and 20 negative bags. We varied the number of search steps as well. More steps would presumably negate the advantage of pruning as both search approaches would have more time to complete.

Figure 4 shows the average AUC for pruning with general-to-specific search minus the average AUC of no pruning. Positive numbers indicate that pruning is improving the performance of the system. The results in Figure 4 again validate the hypothesis that pruning yields a significant improvement in performance. As the difficulty of the problem increases, the ability to prune becomes more crucial. This is seen by the growing difference in AUC as the number of instances in each bag expands. Likewise, as the number of search steps increases, the need for pruning decreases. This is seen by the smaller differences in AUC as the number of search steps increases. Also note that pruning never contributes to a decrease in performance. This is demonstrated on the graph by the all positive differences in AUC.

Figure 5 shows the same differences in AUC with and without pruning for the SPS knowledge representation. These results also validate the hypothesis that pruning increases performance of the system. The main differences between the two figures is in the lack of improvement for the bags with more instances (400-600) and fewer search steps. In the case of the SPS knowledge representation, the problem is too difficult to be solved within the specified number of steps and performance is no different with or without pruning. This leads to the flat area for 10,000 search steps and 400-600 instances per bags. As the number of search steps increases, search with pruning is able to identify better concepts and performance improves compared to search without pruning.

4.2 MUSK *Results*

We also examined the results of using the χ^2 statistic as an evaluation function for use with the canonical MIL data set, MUSK. Although a simple general-to-specific search approach will also work here, empirical results demonstrated that Dietterich et al. (1997) iterated discrimination search yielded the best overall performance. This is likely because this search approach is specifically tuned for use with the MUSK data.

As before, our hypothesis is that the use of the χ^2 statistic for model evaluation will yield comparable or better performance in fewer search steps (using pruning) as the state-of-the-art search and evaluation approach for MUSK. For this experiment, we implemented the iterated discrimination search technique as presented by Dietterich et al. (1997). To ensure that we had implemented this search correctly, we first replicated the results reported by Dietterich. We examined a wider variety of values for the kernel density estimation step, $\tau = \{1.0, 2.0, 5.0, 10.0\}$, and ϵ from 0.9 to 0.0001. The kernels were standard gaussian kernels with width τ and at most $\epsilon/2$ probability outside the bounds of the APR. We found empirically that the variance was fairly high when considering only one run of 10-fold cross validation (as Dietterich did). Instead, we created 30 different 10-fold cross validation sets.

We compared the average AUC obtained using the size-based evaluation criteria proposed by Dietterich et al. (1997) with the results obtained using the χ^2 statistic as the evaluation function. As described in (Bradford and Brodley, 2001), we performed a paired t-test on each run of 10-fold cross validation and obtained a p-value by averaging the t-values for each run. Neither data set was statistically different from the other over any value of ϵ and τ. This was true for both MUSK1 (lowest average p value of 0.26) and MUSK2 (lowest average p value of 0.35). This is in line with our hypothesis that the use of the χ^2 statistic for model evaluation yields comparable or better performance in fewer search steps (using pruning) as more specialized MIL approaches.

4.3 *Image datasets*

Our final experiments compared the effect of the chi-squared evaluation function with the diverse density evaluation function using the COREL image data as described in Chen and Wang (2004) and Chen et al. (2006). As with the previous comparisons, the code is entirely the same except for the evaluation function which enables us to evaluate the effect of the evaluation function alone. Preliminary experiments indicated that the SPS knowledge representation was better suited to this task than the APR representation and we focus on SPS for these experiments.

The COREL image data set has 1000 examples of 20 different images. We used the blob image data publicly available from Chen et al.. As with Chen et al., we looked at each image category separately and labeled all images other than the target category as negative. For example, if image one was the target category, each image in category one became a positive bag and all other example image categories (2-20) were negative bags.

Our hypothesis was that χ^2 would perform as well as or better than diverse density using fewer computational resources. We varied the number of steps from 1000 to 1,000,000 for both evaluation functions and evaluated the AUC across 5 runs of 10-fold cross validation for the first ten image categories in corel. Using the inequality given in Bradford and Brodley (2001), we determined that we needed 5 different runs of 10-fold cross validation to statistically determine if the two evaluation functions were different.

Figure 6 shows the average AUC across the 5 runs of 10-fold cross validation for each of the first ten corel image categories for chi-squared and diverse density. The p-values obtained using the average student t value as described in Bradford and Brodley (2001) are shown on top of each subgraph. This test is performed after 1,000,000 steps of search. In all cases except for category 5, χ^2 either outperforms diverse density (categories 1, 2, 7, 9, 10) or is statistically indistinguishable (categories 3, 4, 6, 8, 10). This is consistent with our hypothesis that the χ^2 evaluation function performs as well as or better than current MIL techniques.

5 Conclusions

The contribution of this paper is of a simple MIL evaluation function that performs as well as or better than current MIL techniques and uses considerably fewer computational resources. MIL is often considered to be a special form of supervised learning that requires special methods. Although Ray and Craven (2005) examined directly using supervised learning techniques for MIL versus the specialized MIL approaches, neither approach was a clear winner. Our technique brings a simple evaluation function to MIL that performs well and enables pruning. Pruning is critical for large search spaces, such as those introduced by relational data. We empirically demonstrated the power of pruning and the performance of the χ^2 evaluation function on a simulated 10-dimensional environment, on the canonical MUSK data set, and on the Corel image classification test.

Acknowledgments

This effort was supported by DARPA and AFRL under contracts F30602-00-2-0597 and F30602-01-2-0566. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, AFRL, or the U.S. Government.

References

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In Bocca, J. B., Jarke, M., and Zaniolo, C., editors, *Proceedings of the 20th Inter-*

national Conference on Very Large Data Bases, VLDB, pages 487–499. Morgan Kaufmann.

Amar, R. A., Dooly, D. R., Goldman, S. A., and Zhang, Q. (2001). Multiple-instance learning of real-valued data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 3–10. Morgan Kaufmann, San Francisco, CA.

Andrews, S., Tsochantaridis, I., and Hofmann, T. (2003). Support vector machines for multiple-instance learning. In S. Becker, S. T. and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 561–568, Cambridge, MA. MIT Press.

Bradford, J. P. and Brodley, C. E. (2001). The effect of instance-space partition on significance. *Machine Learning*, 42:269–286.

Chen, Y., Bi, J., and Wang, J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12).

Chen, Y. and Wang, J. Z. (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939.

Chevaleyre, Y. and Zucker, J.-D. (2001). A framework for learning rules from multiple instance data. In *12th European Conference on Machine Learning*, volume 2167 of *LNCS*, pages 49–60. Springer.

Clearwater, S. and Provost, F. (1990). RL4: A tool for knowledge-based induction. In *Proceedings of the Second International IEEE Conference on Tools for Artificial Intelligence (TAI-90)*, pages 24–30.

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

Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. (2002). Multi-instance kernels. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002)*, pages 179–186. Morgan Kaufmann.

Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–90.

Maron, O. (1998). *Learning from Ambiguity*. PhD thesis, Massachusetts Institute of Technology.

Maron, O. and Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems 10*, pages 570–576, Cambridge, Massachusetts. MIT Press.

Maron, O. and Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 341–349. Morgan Kaufmann, San Francisco, CA.

McGovern, A. (2002). *Autonomous Discovery of Temporal Abstractions from Interaction with an Environment*. PhD thesis, University of Massachusetts Amherst.

McGovern, A. and Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. In Brodley, C. and Danyluk, A., editors, *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 361–368, San Francisco, CA. Morgan Kaufmann Publishers.

McGovern, A. and Jensen, D. (2003). Identifying predictive structures in relational data using multiple instance learning. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 528–535. AAAI Press.

Oates, T. and Cohen, P. R. (1996). Searching for structure in multiple streams of data. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 346–354. Morgan Kauffman.

Ramon, J. and Raedt, L. D. (2000). Multi instance neural networks. In *Proceedings of the IMCL-2000 Workshop on Attribute-Value and Relational Learning: Crossing the Boundaries*.

Ray, S. and Craven, M. (2005). Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 697–704.

Ray, S. and Page, D. (2001). Multiple instance regression. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 425–432. Morgan Kaufmann, San Francisco, CA.

Rosenstein, M. T. and Barto, A. G. (2001). Robot weightlifting by direct policy search. In Nebel, B., editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, volume 2, pages 839–844, San Francisco. Morgan Kaufmann.

Smyth, P. and Goodman, R. M. (1992). An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):301–316.

Webb, G. I. (1995). OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3:431–465.

Zhang, Q. and Goldman, S. A. (2002). EM-DD: An improved multiple-instance learning technique. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Zhang, Q., Goldman, S. A., Yu, W., and Fritts, J. E. (2002). Content-based image retrieval using multiple-instance learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 682–689. Morgan Kaufmann, San Francisco, CA.

Zucker, J.-D. and Chevaleyre, Y. (2000). Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. application to the mutagenesis problem. Technical Report 6, University of Paris.

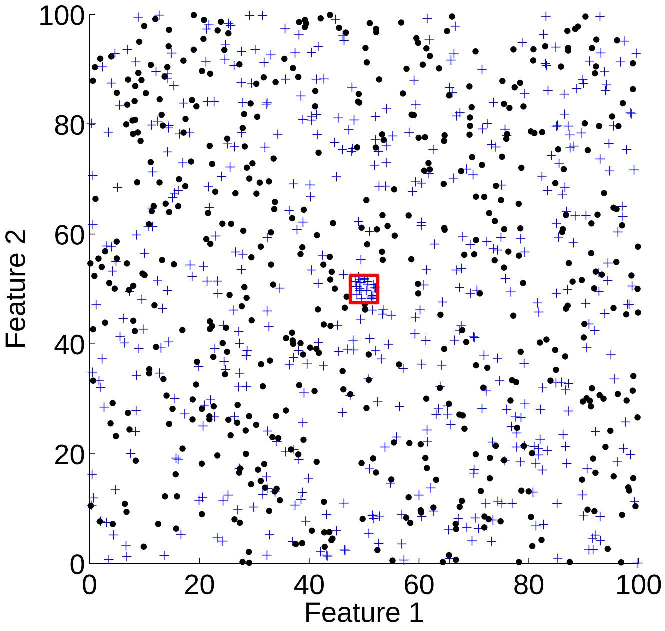
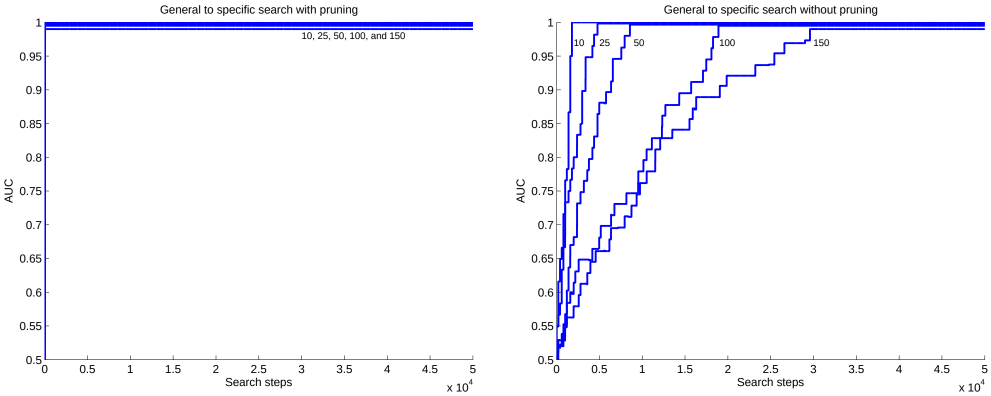


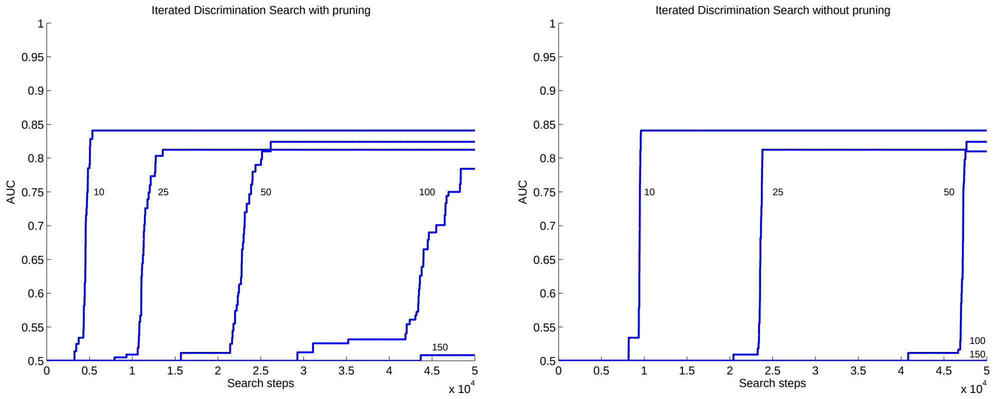
Fig. 1. Two dimensional example of the synthetic data set. We used a 10 dimensional version of this data in our experiments and show the two dimensional version for readability. Positive instances are shown with a + and negative instances are shown with a dot. The target concept is outlined with a square.



A: With pruning

B: Without pruning

Fig. 2. Comparison of average AUC values for general to specific search. Graph A shows the results with pruning enabled and graph B shows the results without pruning. The text indicates the number of instances per bag.



A: With pruning

B: Without pruning

Fig. 3. Comparison of average AUC values for Dietterich's Iterated Discrimination search. Graph A shows the results with pruning enabled and graph B shows the results without pruning. The text indicates the number of instances per bag.

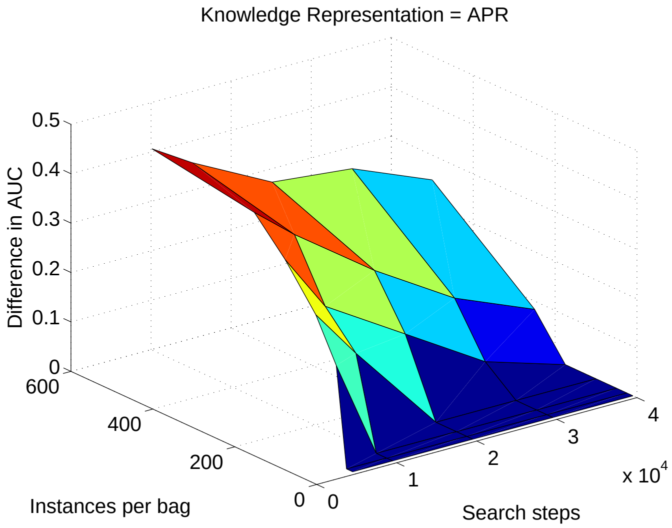


Fig. 4. Average difference in AUC for pruning versus no-pruning in a 10-dimensional MIL problem as a function of the number of instances in each bag and the number of search steps. The evaluation function is the χ^2 statistic in both cases.

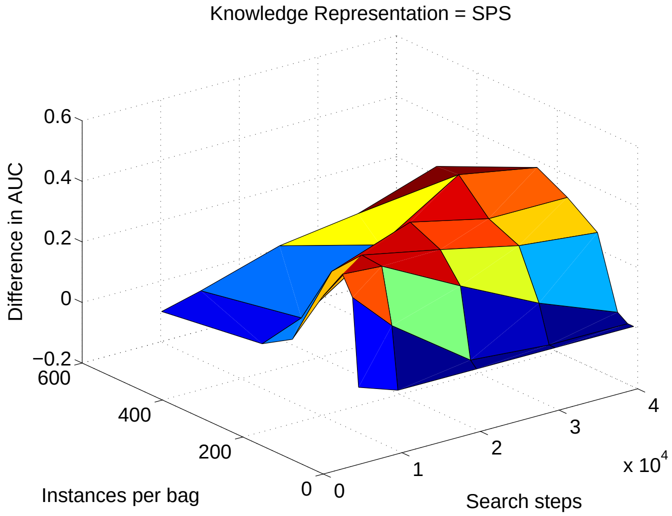


Fig. 5. Average difference in AUC for pruning versus no-pruning in a 10-dimensional MIL problem as a function of the number of instances in each bag and the number of search steps. The evaluation function is the χ^2 statistic in both cases.

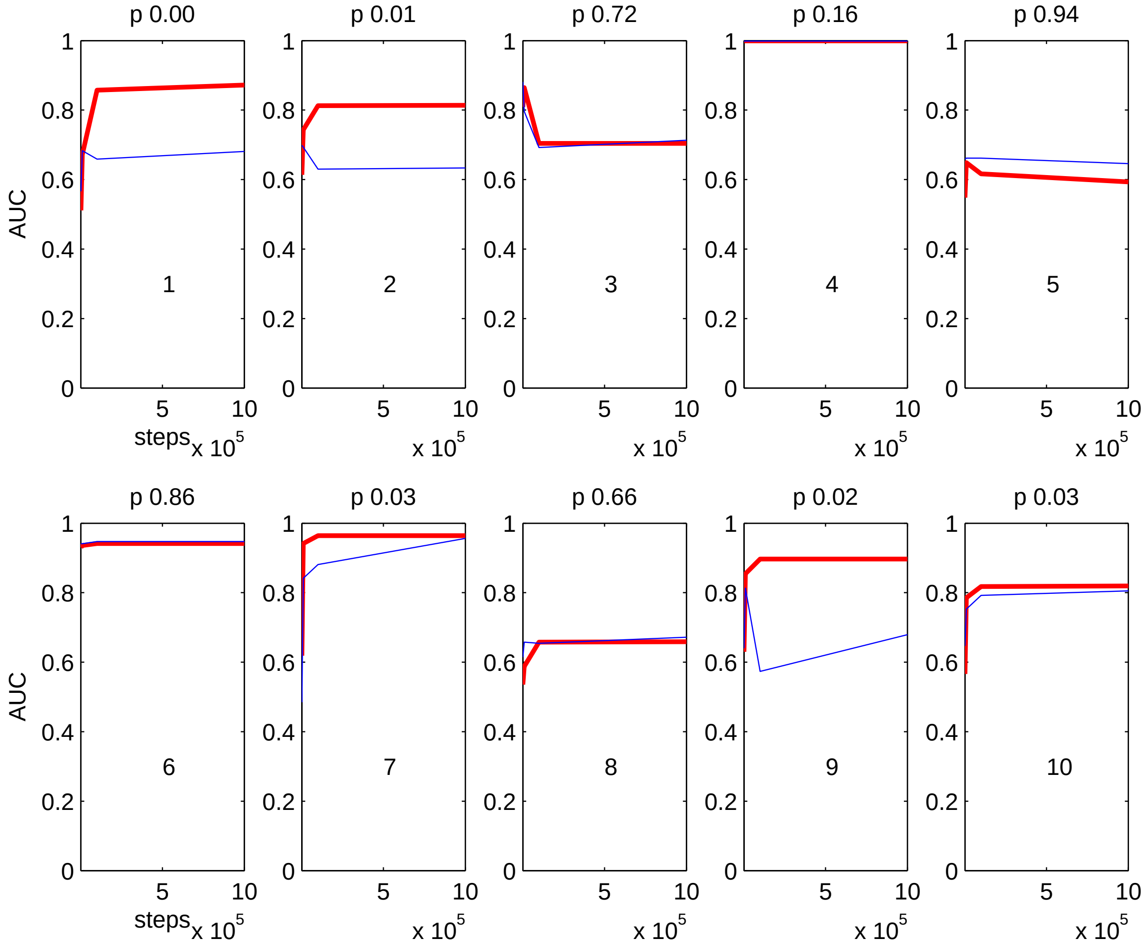


Fig. 6. Average AUC of the chi-squared evaluation function (blue) and diverse density (red) using general to specific search over 5 runs of 10-fold cross validation on the COREL image data set. For each image number (1-10), the images in that data are considered to be the positive bags and all other images are negative.