Submitted to ICML-97.

A Framework for Multiple-Instance Learning

Oded Maron
Artificial Intelligence Lab
NE43-755
M.I.T.
Cambridge, MA 02139
oded@ai.mit.edu

Tomás Lozano-Pérez
Artificial Intelligence Lab
NE43-836a
M.I.T.
Cambridge, MA 02139
tlp@ai.mit.edu

Abstract

Multiple-instance learning is a variation on supervised learning, where the task is to learn a concept given positive and negative *bags* of instances. Each bag may contain many instances, but a bag is labeled positive even if only one of the instances in it falls within the concept. A bag is labeled negative only if all the instances in it are negative. Multiple-instance learning has previously been illustrated in the context of drug activity prediction; in this paper we explore a variety of other applications where multiple-instance learning arises. We describe a general framework, called *diverse density*, which contains a family of algorithms for multiple-instance learning. We apply these algorithms to learn a simple description of a person from a series of images (bags) containing that person.

Keywords: multiple-instance learning, density estimation, recognition

1 Introduction

One of the drawbacks of applying the supervised learning model is that it is not always possible for a teacher to provide labeled examples for training. This limitation is overcome using other approaches such as unsupervised learning or reinforcement learning. A new model, called multiple-instance learning, was formalized in [Dietterich *et al.*, 1997] and provides a different way of modeling the teacher's weakness. Instead of receiving a set of instances which are labeled positive or negative, the learner receives a set of *bags* that are labeled positive or negative. Each bag contains many instances. A bag is labeled negative if all the instances in it are negative. On the other hand, a bag is labeled positive if there is at least one instance in it which is positive. From a collection of labeled bags, the learner tries to induce a concept which will label individual instances correctly. This problem is harder than even noisy supervised learning since the ratio of negative to positive instances in a positively-labeled bag (the noise ratio) can be arbitrarily high.

One example of multiple-instance learning occurs when trying to learn the concept of a chair from pictures of living rooms. Instead of getting cropped images of objects that are labeled as chairs or non-chairs, the learner is shown images of living rooms that are labeled positive only if there is a chair somewhere in the image. A bag is an entire image, and an instance can be a subsection of that image or a description of one of the objects in the image. While this application appears ambitious, we describe a multiple-instance learning system in Section 4 that learns a simple description of a person from a sequence of images that are labeled positive or negative depending on whether the person is in the image or not.

Another application of multiple-instance learning arises in drug activity prediction. In this application, one objective is to predict whether a candidate drug molecule will bind strongly to a target protein known to be involved in some disease state. Typically, one has examples of molecules that bind well to the target protein and also of molecules that do not bind well. Much as in a lock and key, shape is the most important factor in determining whether a drug molecule and the target protein will bind. However, drug molecules are flexible, so they can adopt a wide range of shapes. A positive example does not convey what shape the molecule took in order to bind – only that *one* of the shapes that the molecule can take was the right one. However, a negative example means that none of the shapes that the molecule can achieve was the right key. A bag contains a variety of conformations (a rigid transformation such as rotation or a non-rigid transformation such as a change in torsion angles, i.e. the orientation of atoms around the single bonds of the molecule), and given a sequence of positive and negative bags, a multiple-instance learner outputs a hypothesized shape for a molecule that would make a good fit with the target protein.

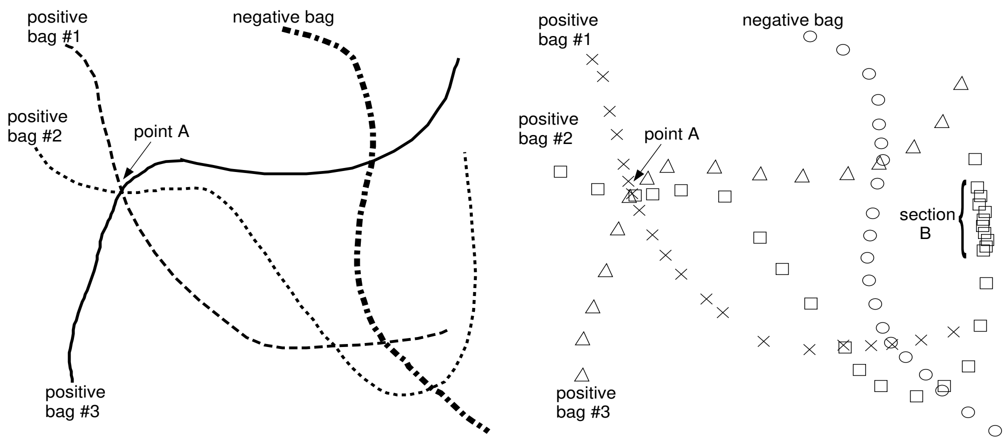
The multiple-instance learning model was only recently formalized by [Dietterich *et al.*, 1997]. They assume a hypothesis class of axis-parallel rectangles, and develop algorithms for dealing with the drug design problem described above. This work was followed by [Long and Tan, 1996], where a high-degree polynomial PAC bound was given for the number of examples needed to learn in the multiple-instance learning model. [Simard *et al.*, 1993] also model their learning problem using bags of instances and tangent distance. However, in their character recognition application, if a bag is labeled as the digit 2, then all instances in it are positive (though skewed) examples of the digit 2.

In this paper, we describe a framework called *diverse density* for solving multiple-instance problems. Diverse density is a measure of how well a hypothesis performs with multiple-instance training examples. There are several ways to compute diverse density with a finite number of bags, each of which can give different results. We describe a family of algorithms for computing diverse density in Section 3. We also show that the algorithms developed in [Dietterich *et al.*, 1997] are members of this family.

We then show results of applying a member of this family to a difficult synthetic training set. In Section 4, we use one of the diverse density algorithms to learn a simple description of a person from a series of images that are labeled positive if the person is in the image and negative otherwise. The hypothesis learned by the system is then used to track the person. Finally, Section 5 concludes the paper with some discussion and open issues.

2 Diverse density

We motivate the idea of diverse density through a molecular example. Suppose that the shape of a candidate molecule can be adequately described by a feature vector. One such representation can be the distance from the center of the molecule to its edge along a uniformly distributed set (of size n) of vectors in 3-D. This ray representation was used in [Dietterich *et al.*, 1997] and [Jain *et al.*, 1994]. One instance of the molecule is therefore represented as a point in n-dimensional feature space. As the molecule changes its shape (through both rigid and non-rigid transformations), it will trace out a manifold through this



(a) The different shapes that a molecule can take on are represented as a path. The intersection point of positive paths is where they took on the same shape.

(b) Samples taken along the paths. Section B is a high density area, but point A is a high diverse density area.

Figure 1: A motivating example for diverse density

n-dimensional space[1]. Figure 1(a) shows the paths of four molecules through a 2-dimensional feature space.

If a candidate molecule is labeled positive, we know that in at least one place along the manifold, it took on the right shape for it to fit into the target protein. If the molecule is labeled negative, we know that none of the conformations along its manifold will allow binding with the target protein. Suppose that the correct set of rays was chosen to represent the molecule, or in other words, we only look at the section of the molecule that binds to the target protein. What do the positive and negative manifolds tell us about the location of the correct shape in feature space? The answer: it is where all positive feature-manifolds intersect without any negative feature-manifolds. For example, in Figure 1(a) it is point A. This is the point that all positive bags have in common, and that all negative bags do not contain.

The example that we have just gone through was extremely idealized and was meant to motivate further exploration, rather than to demonstrate a vi-

[1]In practice, one needs to restrict consideration to shapes of the molecule that have sufficiently low potential energy. But, we ignore this restriction in this simple illustration.

able technique for drug activity prediction. Specifically, it differs from typical multiple-instance applications in four ways:

- A bag does not give us complete distribution information, but only some arbitrary sample from that distribution. We do not have a description of the manifold for molecular shape; instead we have some samples along the manifold. Therefore, Figure 1(a) becomes Figure 1(b).
- Measurements can be noisy – the manifolds might never intersect because there might not be a single ideal shape, or there can be a noisy negative that strays into a positive intersection point.
- The positive bags intersect at more than one point – this implies that there are multiple active shapes..
- It is not known which attributes or features (in this particular example, rays) will result in the correct intersection point. If irrelevant attributes are included, it is possible that the manifolds will never intersect. If relevant attributes are not included, it is possible that negative points will overshadow any positive intersection.

This paper deals with the first three issues, but we also discuss how to tackle the feature selection problem in Section 5. When the representation of a bag changes from a continuous manifold to a set of discrete samples, the problem of trying to find an intersection changes to a problem of trying to find an area where there is both high density of positive points and low density of negative points. This is almost the correct measure of discrete intersection, but we need to refine it a little. The problem with positive density can be seen in Figure 1(b), in section B of bag #2. A bag can contain a group of points that are very close to each other. This is an area of high density, but it is not a good candidate for an intersection point unless *many* different positive bags have instances in that area as well. Therefore, we are not looking for high density, but high "diverse density". We define diverse density at a point to be a measure of how many different positive bags have instances near that point, and how far the negative instances are from that point.

In Section 3 , we discuss in more detail various ways of computing diverse density. However, we can already describe the framework we will use for multiple-instance learning problems. We keep all instances from all bags in memory as points in an n-dimensional space (where n is the number of attributes in the representation of instances), and perform a search for areas in that space that have high diverse density. We output a hypothesis from our concept class that best covers the high diverse density areas. For example, if we were looking for exactly one ideal shape for a molecule, the algorithm would just output the point in feature space that has the highest diverse density.

Noisy instance measurements can be accommodated within this framework by allowing noise in the measurement of diverse density. For example, spreading out the contribution of positive and negative points beyond their actual values, and looking at the number and distance (and not just the existence or absence) of positive and negative instances.

Finally, there is the issue of what a learner should do when there is more than one area of high diverse density. This is not a new problem in the field of machine learning; it is yet another version of the essential problem of inductive inference – no amount of training data can completely distinguish the true concept from possible anomalies in the data. There are two common ways of dealing with this problem. One is to show that the more training the learner gets, the less the chance there is for it to output an incorrect hypothesis. With multiple-instance learning, we claim that the chance of an incorrect hypothesis having high diverse density decreases as either the number of positive bags increases or the number of total negative instances increases. However, increasing the number of instances in a positive bag does not necessarily help since the noise ratio in the bag could either increase or decrease. In Section 3, we give evidence supporting this claim. The second way is to allow a more complicated hypothesis class. In other words, let the algorithm output a disjunction of areas of high diverse density.

3 Algorithms for multiple-instance learning

In this section, we describe in detail one algorithm for computing diverse density. We apply the algorithm to a difficult artificial data set in order to illuminate the difference between density and diverse density. We also show how increasing the number of bags reduces the chance of false positives. We then generalize the algorithm to obtain a family of algorithms for computing diverse density.

3.1 MD: Multiplicative Density

We developed an algorithm called Multiplicative Density (MD) for computing diverse density. For every instance in every positive bag, place a Gaussian bump centered around it. Negative instances get a negative Gaussian bump centered around each[2]. We define "regular density" to be the density of positives vs. negatives. To calculate it at point p, we would simply calculate the sum of all the Gaussians at that point (a form of kernel density estimation, as in [Fukunaga, 1990]). However, that ignores the diversity of contribution at that point. We would like the diverse density at that point to be higher if there are many different bags that contribute heavily to it, and lower if the main contribution comes from just one bag. The solution is to add up each positive bag's contribution at p, and then *multiply* them together .

Formally, we have n positive bags and m negative bags, and suppose without loss of generality that each contains k instances. Let the i^{th} instance in positive bag j be $B^+_{j(i)}$, and likewise $B^-_{j(i)}$ for a negative bag. Let $G_x(p)$ be a symmetric Gaussian with mean x, evaluated at p. The contribution of the j^{th} positive bag at point p is $B^+_j(p) = \sum_{i=1}^{k} G_{B^+_{j(i)}}(p)$. Likewise, the contribution of the j^{th} negative bag at point p is $B^-_j(p) = -\sum_{i=1}^{k} G_{B^-_{j(i)}}(p)$. The entire negative contribution is $B^-(p) = \sum_{j=1}^{m} B^-_j(p)$.

MD calculates the diverse density at point p as $\prod_{j=1}^{m}(B^+_j(p) + B^-(p))$. To prevent problems with negative densities, if a positive bag's contribution is less than or equal to the total negative contribution at p, than their difference is set to some small ϵ.

Figure 2 shows the steps taken by MD on the following set of 1-dimensional data: positive bag #1 contains instances 1 and 5; positive bag #2 contains instances 2 and 3; positive bag #3 contains instances 1.5 and 4; negative bag #1 contains instances 2.5, 3.5, and 4.5. The underlying concept is the region $[1, 2]$. In (a) we see the Gaussian bump associated with every instance. In (b), every bag's contribution is added up. In (c), the negative contribution is added to every positive bag. Finally, in (d), the curves in (c) are multiplied (with $\epsilon = 0.0001$) and the peak indeed falls within $[1, 2]$.

The motivation behind MD is that we would like n close points from n different bags to weigh exponentially more than n points from the same bag.

[2]Despite the fact that we are using Gaussians and computing measures that are named "density", we are actually not computing probabilities and therefore normalization is not a factor.

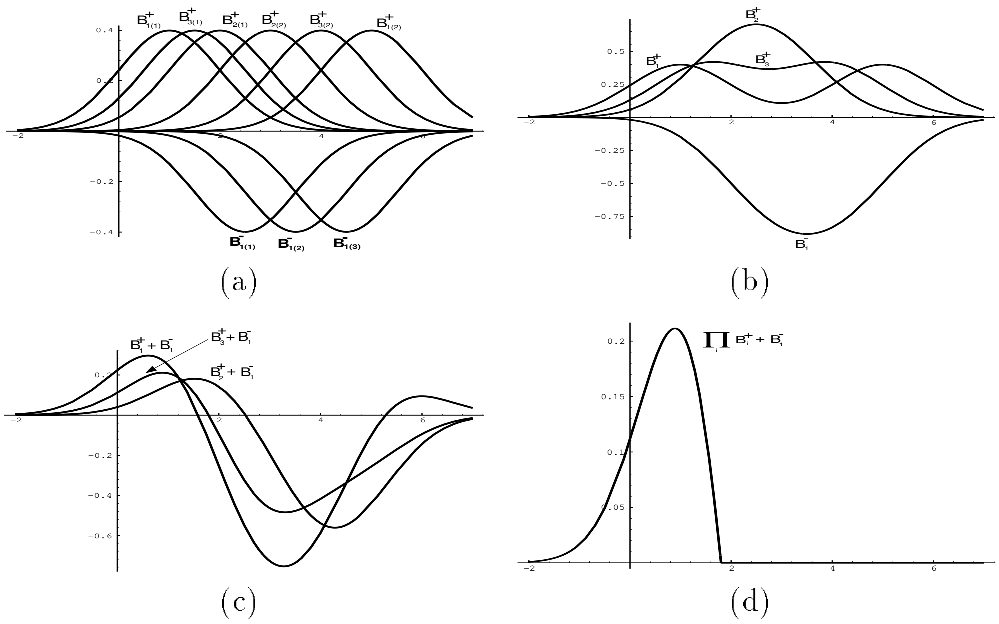


Figure 2: Snapshots of the MD algorithm in progress. (a) is the contribution of each instance. (b) is the contribution of each bag. (c) adds the negative contribution into every positive bag. (d) is the diverse density across the domain, with a peak in the underlying concept.

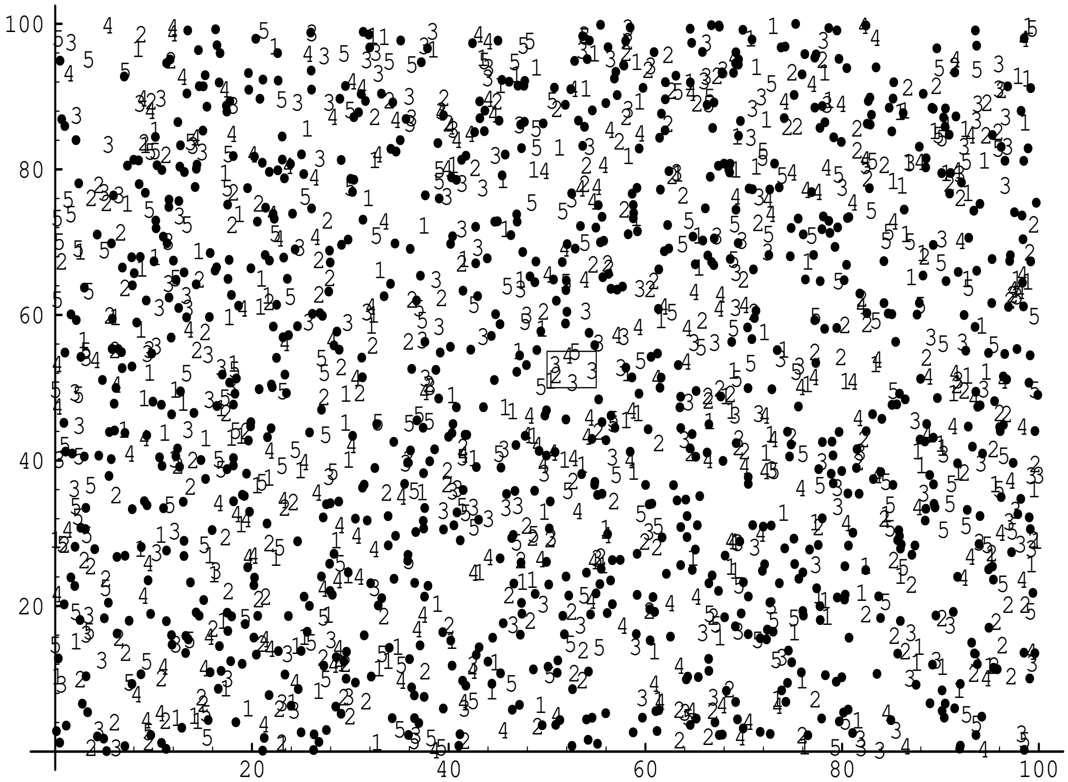
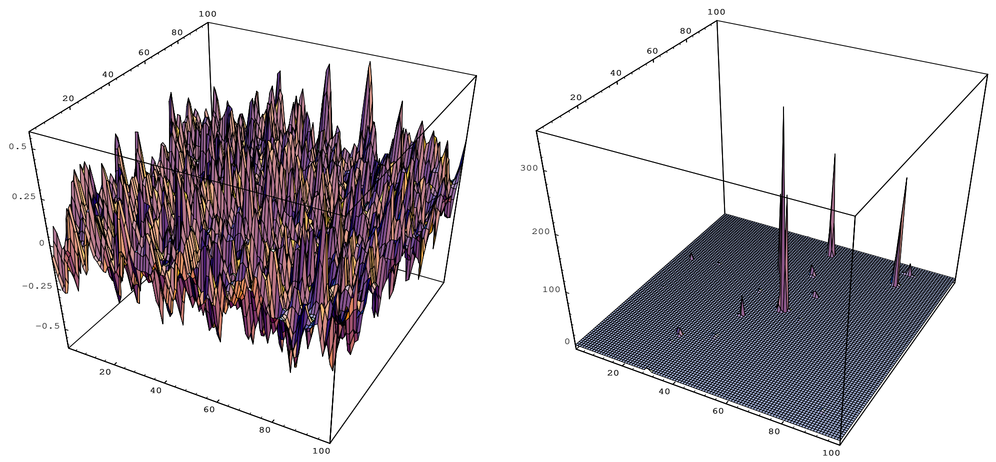


Figure 3: Negative and positive bags drawn from the same distribution, but labeled according to their intersection with the middle square. Negative instances are dots, positive are numbers associated with their bag. The square contains at least one instance from every positive bag and no negative instances.

In addition, we force the diverse density of an intersection of n bags to be significantly higher than an intersection of $n-1$ bags. We also take negative bags heavily into account: each positive bag is weighed against all the negative bags. This is to ensure that diverse density cannot be high in areas where there cannot be an intersection. We use Gaussians so that intersection does not have to be exact, but the closer different bags come to each other, the better.

We created the following artificial data set: n positive bags and m negative bags, each with k instances. Each instance was chosen randomly from a $[0,100] \times [0,100] \in \mathcal{R}^2$ domain, and the concept was a 5×5 square in the middle of the domain. A bag was labeled positive if at least one of its instances



(a) Surface using regular density (b) Surface using diverse density

Figure 4: Density surfaces over the example data of Figure 3

fell within the square, and negative if none did. An example (with $n = m = 5$, and $k = 200$) is shown in Figure 3 – the negative instances are dots, and each positive instance displays the bag it is from. The square in the middle contains at least one instance from every positive bag and no negative instances. This is a difficult data set because both positive and negative bags are drawn from the same distribution. They only differ in a small area of the domain.

Using regular density (adding up the contribution of every positive bag and subtracting negative bags), we can plot the density surface across the domain. Figure 4(a) shows this surface for the data set in Figure 3, and it is clear that finding the peak (a candidate hypothesis) is difficult because of the abundance of local maxima and because many points have similar maxima. However, when we plot the diverse density surface in Figure 4(b), it is easy to pick out the global maximum which is in fact within the desired concept.

The other major peaks in Figure 4(b) are the result of a chance concentration of instances from different bags in another part of the space. With a bit more bad luck, one of those peaks could have eclipsed the one in the middle. However, the chance of bad luck decreases as the number of bags increases. This can be seen in Figure 5, where the probability of the top diverse density

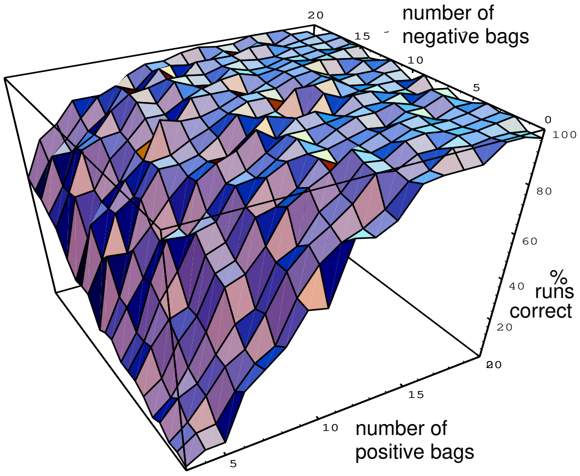


Figure 5: As the number of positive or negative bags increases, the number of times that the maximum diverse density is contained by the true concept also increases.

landing within the true concept increases as either the number of positive or negative bags increases. The number of points per bag was held at 200 and 100 randomly generated data sets were used to estimate the probability at every point in the graph.

3.2 A general family of diverse density algorithms

A curious reader might ask: why did you use Gaussians? Why did you choose to combine negative and positive bags in that manner? Why count ten instances from the same bag more than just one? The answer is that the MD algorithm seems to work well in practice, but there are many parts of it that can be modified without losing the basic spirit. However, any algorithm for computing diverse density at point p needs to balance three factors:

- Number of instances from different positive bags that are near p
- Number of negative instances that are near p
- Number of positive instances (regardless of bag) near p

In addition to balancing these, the algorithm also needs a distance metric so that "nearness" can be measured. We propose the following functions as building blocks for creating arbitrary diverse density measures at a point p.

$\boldsymbol{D}(\boldsymbol{a})$ – the distance between a and p.

$\boldsymbol{W}(\boldsymbol{x})$ – a weighting of the distance. In MD, we used a Gaussian. However, we can use other functions such as:

- Step – returns 1 if the points are within a certain distance Q, 0 otherwise. [Dietterich *et al.*, 1997], for example, describes algorithms for varying Q so that there are many positive bags represented within it and few negative bags.

- Other kernel functions with different widths – the smaller the width, the less influence far points have.

$\boldsymbol{C}^{+}(\boldsymbol{x}_1, ..., \boldsymbol{x}_k)$ – combines instances from one positive bag. MD simply sums their weighted contribution. Using Max will return the contribution of the closest point to p.

$\boldsymbol{C}^{-}(\boldsymbol{x}_1, ..., \boldsymbol{x}_k)$ – combines instances from one negative bag.

$\boldsymbol{F}(\boldsymbol{C}_1^{+}, ..., \boldsymbol{C}_n^{+}, \boldsymbol{C}_1^{-}, ..., \boldsymbol{C}_m^{-})$ – combines negative and positive bags. MD returns the product of the difference between each positive bag and the entire negative set. Regular density takes the sum of each bag's contribution. Another possibility for combining the contributions is to treat bags as a set of samples from different probability distributions. The diverse density of a point is the chance that it is in all the positive bags and not in any of the negative bags: $\prod_{j=1}^{m} B_j^{+}(p) \cdot (1 - B^{-}(p))$. However, this requires normalization of the contributions and some strong assumptions about the structure of the problem.

The functions D, W, and C essentially define a family of lazy learners as in [Atkeson *et al.*, 1997]. Various combinations of those functions can produce algorithms such as nearest-neighbor or kernel regression. Unlike lazy learners, though, each bag's estimate needs to be combined. In addition, evaluation of diverse density usually needs to be performed repeatedly on different points in the space until a maximum is found.

3.3 Computational issues

There are two critical factors involved in making these algorithms computationally efficient. One is to insure that the time to compute diverse density of an individual point does not grow as fast as the total number of training instances. The second is to insure that the time to find the maximum diverse density does not grow exponentially with the number of dimensions in the domain.

One of the reasons for the use of Gaussians as weighting functions is that they drop off fairly quickly. To compute the contribution of a bag at point p, we only need to look at instances that are close to p – all the other ones contribute nearly zero. By partitioning the space (much like [Omohundro, 1991]), we can achieve nearly constant time computation of diverse density at a point.

Finding the maximum diverse density is a more difficult issue. In general, we are searching an arbitrary density landscape and the number of local maxima and size of the search space could prohibit any efficient exploration. In this paper, we use gradient ascent with multiple starting points. This has worked succesfully in every test case because we know what starting points to use. The maximum diverse density point is made of contributions from some set of positive points. If we start an ascent from every positive point, one of them will be closest to the maximum, therefore contribute the most to it, and therefore have a climb directly to it.

4 Applying MD to a recognition task

We investigated a new application of multiple-instance learning in order to test the diverse density algorithms in a real-world setting. We would like to learn a description of a person so that we can recognize them when they enter a room or track them around. We would like to learn this description non-obtrusively, so that the person does not have to pose for the camera. To help us, we have a touch-sensitive board that returns coordinates of where the person touched it (when they were lecturing, for example). If we grab an image every time the board is touched during lecture, we have a set of positive training examples. However, the board's output is noisy and we don't know if the person was to the left or right of the touch-point. The solution is to consider each image as a bag, and take several subimages around the touch-point to be instances. If a bag is positive, we know that at least one of the subimages contains a cropped image of the person. We sampled 54 subimages for every bag, varying their location, height, and width.

We chose a very simple scheme for representing a person. Divide the image into three parts that roughly correspond to head, torso, and legs. Compute a color histogram of each of these three parts and take the most common color of each. We use hue and saturation as color measurements (to alleviate problems with different lighting conditions), resulting in a six-dimensional vector that describes a subimage instance. Figure 6 shows the process of going from a picture to the bag that represents it.

This representational scheme means that the most we can hope to learn is the person's pants color, shirt color, and hair color. While limiting, it turns out to be powerful enough to distinguish people in many situations. In addition, this representation is easily extendible by dividing the person into more sections, and by looking at the entire histogram rather than just the dominating color in it.

We trained an MD algorithm on various people, using about 10 positive and 10 negative bags for each person. The negative bags were images of an empty room or of other people at the board. In every case, the system succeeded in learning the color of the pants and shirt of the person. In cases where a person faced the camera and also had their back to the camera, the algorithm had difficulty in identifying a dominating color for their head. The most impressive demonstration of the system occured when the set of positive training images included two people – only one of which was in all the images (see Figure 7). The algorithm successfully learned a description of the common person.

Eventually, we would like to scale the application so that given a sequence of images of the person anywhere (not just touching the board), we can learn a description of that person. Each bag will need to contain man more instances because we will no longer know approximately where the person, nor will we know if they are standing, sitting, or lying down. The experiments shown thus far make us confident that this is possible with a less coarse representation and many more training images.

5 Discussion

Multiple-instance learning falls in the gap between supervised and unsupervised learning. Like supervised learning, there is a teacher who can label the bags. However, the teacher cannot give a specific label to any instance in a positively labeled bag. Like unsupervised learning, there is no test set with which to measure how well the algorithm did; instead we have created an artificial metric – diverse density. This is done in the spirit of artificial metrics that

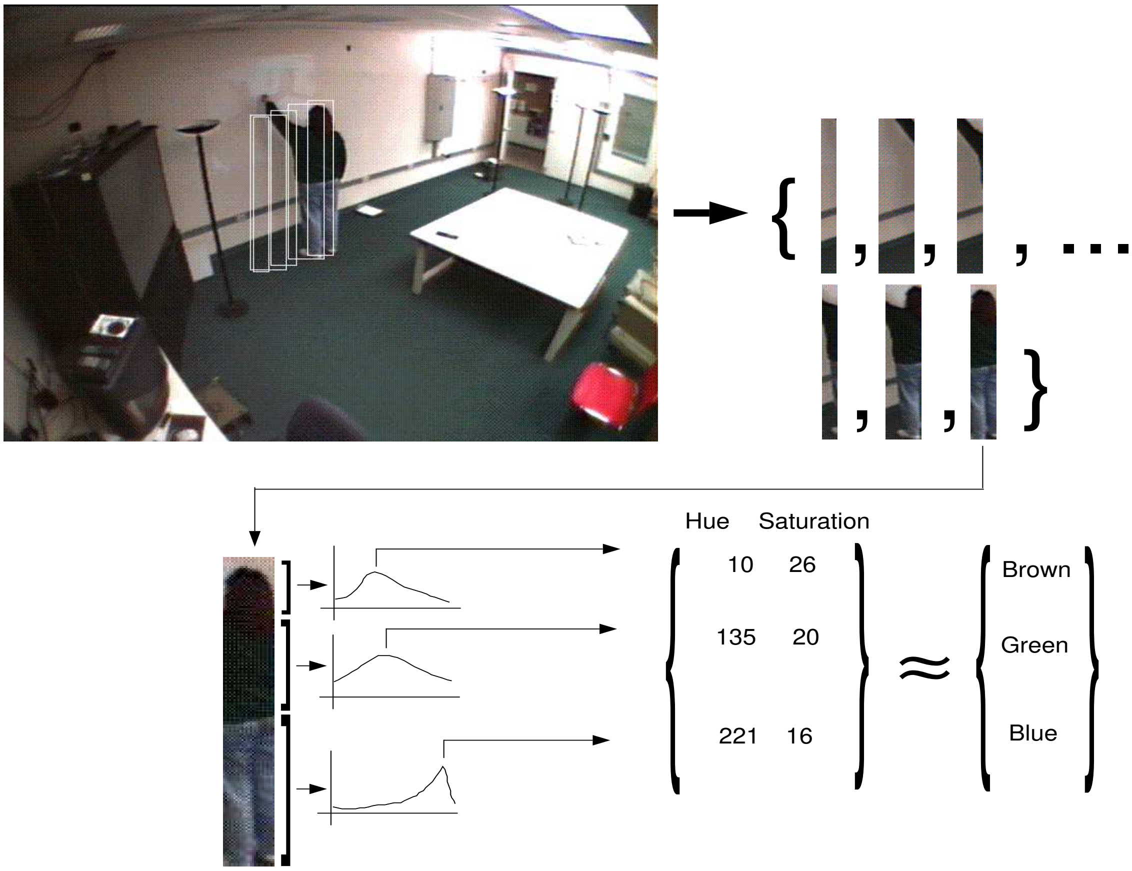


Figure 6: Extracting instances from one image bag: the white rectangles are some of the 54 subimages sampled around the touch-point. For each subimage, the modes of the hue-saturation histograms are computed for the head, torso, and leg areas (actual histograms not shown). This results in a six-dimensional point representing one instance. The process is repeated for every subimage.

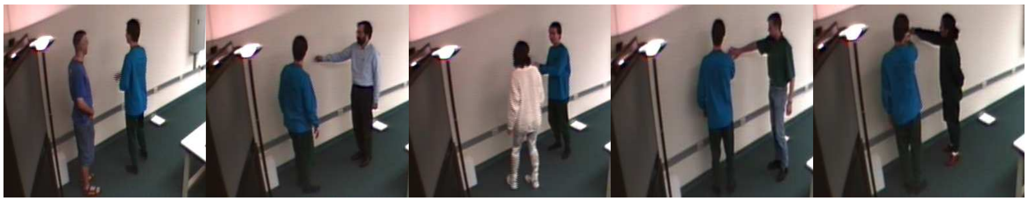


Figure 7: A training set of images with one person in common

clustering algorithms use; for instance, maximizing the inter-cluster distance and minimizing the intra-cluster distance [Duda and Hart, 1973].

Feature selection, one of the central issues in any learning algorithm, has not been discussed much in this paper. We have assumed that the dimensions of our data are all independent, all equally important for prediction, and all normalized to be in the same range. While this was a strong assumption, it helped us concentrate on the essential aspects of diverse density. Just as a search for maximal diverse density is used to find the correct region in the domain, it can also be used to find the correct subset of features to use or more generally, the correct weighting of attributes. We are working on such algorithms in the drug design domain.

In this paper, we have only discussed learning algorithms for binary classification. In situations where the number of classes is more than two, the framework can easily accommodate them. For each class, take the bags labeled with that class to be positive, and all the rest to be negative. Learn the concept for that class, and repeat for each class. When asked to predict the class of point x, return the class with the highest diverse density at x.

This framework can also accommodate a certain type of regression which comes up in molecular binding prediction. Normally, chemists can not only say whether two molecules bind or not, they can give an estimate of the bind strength. Instead of learning the shape of a ligand that binds, we want to learn the shape of a ligand that has the highest binding strength. With the MD algorithm, for each instance in a bag, we can place a bump whose height is proportional to the strength of the binding associated with that bag.

References

[Atkeson *et al.*, 1997] C. G. Atkeson, A. W. Moore, and S. A. Schaal. Locally Weighted Learning. *AI Review (to appear)*, 1997.

[Dietterich *et al.*, 1997] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the Multiple-Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence Journal (to appear)*, 1997.

[Duda and Hart, 1973] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[Fukunaga, 1990] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 1990.

[Jain *et al.*, 1994] A. N. Jain, T. G. Dietterich, R. H. Lathrop, D. Chapman, R. E. Critchlow Jr., B. E. Bauer, T. A. Webster, and T. Lozano-Pérez. Compass: A shape-based machine learning tool for drug design. *Journal of Computer-Aided Molecular Design*, 8:635–652, 1994.

[Long and Tan, 1996] P. M. Long and L. Tan. PAC-learning axis alligned rectangles with respect to product distributions from multiple-instance examples. In *Proceedings of the 1996 Conference on Computational Learning Theory*, 1996.

[Omohundro, 1991] Stephen M. Omohundro. Bumptrees for Efficient Function, Constraint, and Classification Learning. In Moody Lippmann and Touretzky, editors, *Advaces in Neural Information Processing Systems 3*, San Mateo, CA, 1991. Morgan Kaufmann.

[Simard *et al.*, 1993] P. Simard, Y. Le Cun, and J. Denker. Efficient Pattern Recognition Using a New Transformation Distance. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Neural Information Systems 5*. Morgan Kaufmann, 1993.