Deterministic Annealing for Multiple-Instance Learning

Peter V. Gehler
Max Planck Institute for Biological Cybernetics
Tübingen, 72070, Germany
peter.gehler@tuebingen.mpg.de

Olivier Chapelle
Max Planck Institute for Biological Cybernetics
Tübingen, 72070, Germany
olivier.chapelle@tuebingen.mpg.de

Abstract

In this paper we demonstrate how deterministic annealing can be applied to different SVM formulations of the multiple-instance learning (MIL) problem. Our results show that we find better local minima compared to the heuristic methods those problems are usually solved with. However this does not always translate into a better test error suggesting an inadequacy of the objective function. Based on this finding we propose a new objective function which together with the deterministic annealing algorithm finds better local minima and achieves better performance on a set of benchmark datasets. Furthermore the results also show how the structure of MIL datasets influence the performance of MIL algorithms and we discuss how future benchmark datasets for the MIL problem should be designed.

1 Introduction

In the multiple-instance learning (MIL) scenario training patterns are available only in bags for which a bag label is known. The pattern labels remain ambiguous in that although instances from the negative class are known, one has to infer which patterns belong to the positive class. It is only known that *at least* one pattern of a positive labeled bag belongs to the positive class. Since the MIL problem was introduced in [Dietterich et al., 1997] for the task of drug activity prediction, a number of different applications emerged in the literature. Up to now the span of applications cover a variety of problems such as identification of proteins [Tao et al., 2004], content based image retrieval [Zhang et al., 2002], object detection [Viola et al., 2005] and prediction of failures in hard drives [Murray et al., 2005].

Several special purpose algorithms for MIL have already been proposed. Those which try to infer the missing labels (or parts thereof) share the problem of not being convex. The pattern labels enter the objective functions as discrete variables creating combinatorial problems which are typically hard to solve. Most authors provide heuristic learning schemes to cope with this problem. In this paper we will apply deterministic annealing which is a standard tool from non-convex optimization to MIL versions of support vector machines (SVM). Our results show that this learning scheme finds better local minima which does not automatically translate into lower test error. This indicates an inadequacy of the objective function and we propose a refined version which overcomes its problems. This new SVM version also sheds light on the structure of the current benchmark datasets which might lead to the design of more appropriate MIL benchmark datasets in the future.

2 Multiple Instance Learning

In the classical supervised classification problem one is given a set of i.i.d. labeled patterns $(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ on which one tries to build a classifier $f : \mathbb{R}^d \to \{-1, 1\}$. The multiple-instance learning problem is a generalization of this setting where training patterns are given as bags $B_i \subset \mathbb{R}^d, i = 1, \ldots, N$ with labels Y_i provided only for the bag. Each bag consists of possibly many patterns $B_i = \{x_i^1, x_i^2, \ldots, x_i^{m_i}\}$. The bag label induces constraints on pattern labels in an asymmetric way. We want to emphasize that one has to distinguish between bag and pattern label and bear in mind that they have a different meaning (examples will be given in Section 2.3). A negative labeled bag contains only patterns to which a negative label can be assigned to. On the other hand a positive bag label only enforces that the bag contains *at least* one pattern in the bag which can be assigned to the positive class. We will refer to this pattern as the *witness* of the bag. There is no information about the other points, they might not even belong to either the positive or negative class. In the remainder a pattern label for x_i^j will be denoted by y_i^j.

One can roughly divide the different approaches that

have been proposed for MIL in three different categories. The first category consists of methods which ignore the MIL setting and treat the problem as a supervised one, but on the bag-level. Prominent members of this category are set kernel for SVMs [Tao et al., 2004, Gärtner et al., 2002, Chen et al., 2006] or extensions of the nearest neighbor algorithm using Hausdorff distances [Wang and Zucker, 2000]. In this paper we will consider only methods from the remaining two categories described in following two sections. We will review the proposed SVM formulations for the MIL problem and subsequently show how a deterministic annealing procedure can be used to solve them.

2.1 Identifying all labels

Models from the second category try to impute all the missing labels of the patterns in the positive labeled bags and subsequently treat the problem as a supervised one. These models implicitly assume that each ambiguous point can indeed be assigned to either the positive or negative (pattern) class, an assumption which clearly depends on the nature of the dataset. This approach motivated the construction of *mi-SVM* in [Andrews et al., 2002] which introduced SVMs to the MIL problem. Ambiguous labels enter the objective function as discrete variables over which one tries to optimize. Ignoring these additional variables the objective function is the same as in the standard supervised SVM case

$$\mathcal{L}(w, b, \xi_i, \{y_i^j\}) = \frac{1}{2}\|w\|_2^2 + C\|\xi\|_2, \tag{1}$$

where $w \in \mathbb{R}^d, b \in \mathbb{R}$ are the weight vector and offset of the SVM. The difference appears in the constraint set which is modified to ensure label consistency with the bag label

$$\text{(mi-SVM)} \qquad \begin{aligned} & y_i^j(\langle w, x_i^j\rangle + b) \geq 1 - \xi_i^j, \xi_i^j \geq 0, \\ & \forall i : Y_i = 1, \ \sum_{j=1}^{m_i} \frac{y_i^j + 1}{2} \geq 1, \\ & y_i^j \in \{-1, 1\} \\ & \forall i : Y_i = -1, \ y_i^j = -1. \end{aligned} \tag{2}$$

Due to the discrete variables y_i^j this problem is no longer convex but a combinatorial one. To find the global minimum of $\mathcal{L}$ one would have to check all possible assignments of the labels. Therefore Andrews et al. [2002] use a heuristic method to optimize this objective function. Starting by assigning all pattern labels from positive labeled bags to be 1, the optimization for the parameters w, b and the assignment for y based on the resulting classification boundary is alternated. After each step the constraints (2) are checked and if necessary enforced by setting the label of the pattern whose function output is least negative to 1.

2.2 Identifying the witness

Finally the last category consists of methods which aim to identify the witness in the positive labeled bags which is responsible for the label. Successively a classifier is build on those witnesses only, while all other points drop out of the problem. SVM formulations of this versions are the *MI-SVM* by Andrews et al. [2002] and the *MICA* algorithm by Mangasarian and Wild [2005]. Both utilize the same objective function as in Eq.(1) but equip it with a different set of constraints.

$$\text{(MI-SVM)} \qquad \max_j(\langle w, x_i^j\rangle + b) \geq 1 - \xi_i, \xi_i \geq 0$$

$$\text{(MICA)} \qquad \begin{aligned} & \textstyle\sum_j \nu_i^j \langle w, x_i^j\rangle + b \geq 1 - \xi_i, \\ & \textstyle\sum_j \nu_i^j = 1, \ \ \nu_i^j, \xi_i \geq 0. \end{aligned}$$

Patterns from negative labeled bags are all used with slack variables ξ_i^j as in mi-SVM. The MI-SVM directly selects the "most positive" patterns from the bags and builds the classifier with them. MICA is not directly identifying a witness in the bag but a convex combination of all points in a bag which acts as a witness. This removes the integer representation involved in the MI-SVM at the expense of adding bilinear constraints to the program. It is only for positive bags with more than one pattern having an output larger than one that the MICA and the MI-SVM will differ in their choice of the witness. As discussed above both methods avoid assigning a label to all patterns. Both [Andrews et al., 2002] and [Mangasarian and Wild, 2005] proposed an algorithm which alternates between updates of the SVM parameters and identification of the witnesses. The main difference in both formulations is that for MICA l_1 penalization of the weights is used whereas the objective functions from [Andrews et al., 2002] use l_2 penalization. From a machine learning perspective it is a priori unclear which norm is better suited for a given problem and this is why we chose to use l_2 penalization to unify the presentation. All presented algorithms (and those which follow) can be kernelized and are easily implemented by extension of any SVM solver. In a kernelized version the convex combination is taken in the associated RKHS.

The EM-DD algorithm [Zhang et al., 2002] employs a probabilistic framework to find a witness t or multiple witnesses t_i of the positive class in feature space. Those points t_i should be close to at least one pattern from every positive labeled bag and as far away as possible from all points in the negative labeled bags. Again the optimization of this problem iterates between updates of t and assignments of the witnesses of the positive labeled bags.

As most algorithms which are used for MIL mi-SVM,MI-SVM,MICA and EM-DD share the problem of being combinatorial problems of the instance labels. All of

them require the minimization of a non-convex objective function and use heuristics to do so. In the remainder of the paper we will describe how deterministic annealing can be used to obtain better local minima of the objective functions for MI-SVM, MICA and mi-SVM.

2.3 Imputing all labels versus identifying witnesses

Which of the methods described so far should be applied to a given MIL problem? If one has the knowledge that for a given dataset patterns in the positive labeled bag can clearly be divided into a positive and a negative class, the mi-SVM algorithm is the natural choice. If however all of the patterns from positive labeled bags are believed to belong to the positive class a standard SVM can be employed where the label ambiguity is ignored altogether. The popular benchmark dataset for MIL MUSK1 falls under this category. A SVM decision function obtained by ignoring the ambiguity of the pattern labels already gives a classification performance of 85.6% which is already better than the reported results for EM-DD,MI-SVM and MICA. This finding is also reported in [Ray and Craven, 2005].

Consider on the other hand a face detection problem. Some image containing a face is split into all possible patches. The union of all patches forms a bag which we label positive as one of the patches shows the face completely and centered. However there is a continuum of patterns in this bag. Some show the face only partially others only parts like eyes. These patches should not belong to either of the classes. It is sufficient to identify the one patch with the complete face on it. On the other hand all patches from an image without a face can be labeled negative without any problem. In this case an algorithm like the MI-SVM or MICA is the one of choice.

There are also other problems which in the literature are considered to be MIL problems. Assume an image depicting a car is represented as a collection of small patches each single one not containing the entire car. The label "car" for the image, tells us that there is at least one car-part-patch amongst all those patches. But this conclusion can not be reversed. The existence of a car-patch in an image does not allow the conclusion that there is a car shown in the image. In this case it is clearly a combination of patches which matters. Thus in this example bag labels and pattern labels have different meanings. Nevertheless there are approaches which try to solve this problem using the MIL framework. In the setting presented in this paper the corresponding task would be to classify car-part-patches against other patches.

3 Deterministic Annealing

Deterministic annealing (DA) is a special case of an homotopy method and may be applied in a more general context than introduced here. Our outline mostly resembles Sindhwani et al. [2006] who applies DA to semi-supervised learning. For a more detailed review we refer to Rose [1998]. Suppose one is given a non-convex optimization problem of the form $y^* = \arg\min_{y\in\{0,1\}^n} F(y)$. DA finds a local minimum of this function as follows. Firstly the discrete variables are regarded as random binary variables defined over a space of probability distributions $\mathcal{P}$. Instead of solving the optimization problem directly one searches for a distribution $p \in \mathcal{P}$ which minimizes the expected value of F. By doing so, the optimization problem becomes continuous but is not easier to solve. For this reason, an additional convex term is added to the objective function: the entropy $\mathcal{S}$ of the distribution

$$p^* = \arg\min_{p\in\mathcal{P}} E_p(F(y)) - TS(p). \tag{3}$$

The parameter T which controls the trade off between the expectation and the entropy is called the *temperature* of the problem. As a first observation, note that for $T = 0$ and $\mathcal{P}$ including all point-mass distributions over $\{0,1\}^n$ the global minimizer p^* of the problem above will put all of its mass on the global minimizer of F. Thus the new formulation preserves the optimality of the original problem. If on the other hand $T \gg 0$ the entropy term in Eq.(3) dominates the objective function and the problem will be solved easily thanks to convexity. So we can find a solution by solving a sequence of problems for values of $T_0 > T_1 > \ldots > T_\infty = 0$ each of which is initialized at the solution obtained by the previous one. This sequence of temperatures is referred to as the annealing schedule. As T approaches zero the influence of the entropy term vanishes and the distribution will become more concentrated on the minimum of $E_p[F]$. In this case we can identify the discrete variables y by p. Of course there is no guarantee for global optimality because there might not be a path connecting the local minimizers for the chosen sequence of T to the global optimum of F.

4 DA applied to Multiple instance learning

We will now derive deterministic annealing algorithms for the formulations of the support vector machines described in Section 2. Recall that the objective function Eq.(1) is defined on both discrete and continuous variables. Therefore for a given temperature T we have to optimize p and the SVM parameters w and b. This can be done with an alternating scheme in a coordinate descent fashion which is guaranteed to decrease the objective function. Note that each alternating step itself is an easy to solve convex problem.

4.1 Deterministic Annealing for SVM inferring all patterns

The goal of the mi-SVM is to impute all missing labels of the instances in the positive labeled bags. Following the DA principle we will regard the label y_i^j of a pattern from a positive labeled bag $x_i^j \in B_i, Y_i = 1$ as an independent binary random variable. In principle our space of distributions $\mathcal{P}$ consists of all possible distributions over p_i^j. However since we know that there are no terms in the objective function which couple the pattern labels y_i^j we know that the optimal distribution has to factorize. Therefore we can restrict our search space $\mathcal{P}$ to the factorial distributions. The distribution for y_i^j is defined by $P(y_i^j = 1) = p_i^j$ which implies $P(y_i^j = -1) = 1 - p_i^j$. One can think of the value of p_i^j as the belief that the instance x_i^j belongs to the positive class. To simplify the notation we will fix $p_i^j = 0$ for all patterns from negative labeled bags. The constraint on the pattern labels from mi-SVM directly translates to $\sum_j p_i^j \geq 1$, namely the expectation of positive labeled patterns in a positive labeled bag is larger than one. Applying Equation (3) to the objective function from Eq.(1) we arrive at the following minimization problem which we now write as a unconstrained one with a loss function

$$\begin{aligned}\mathcal{L}_T(w,b,p) &= \|w\|_2^2 + C\sum_{i=1}^{N}\sum_{j=1}^{m_j}[p_i^j l(\langle w, x_i^j\rangle + b) \\ &\quad +(1-p_i^j)l(-\langle w, x_i^j\rangle - b)] \\ &\quad +T\sum_{i,j=1}^{N,m_j}(p_i^j \log p_i^j + \\ &\qquad (1-p_i^j)\log(1-p_i^j)). \quad (4)\end{aligned}$$

The constraint set to this objective is

$$\text{(AL-SVM)} \qquad 0 \leq p_i^j \leq 1, \ \ \forall i,j \quad (5)$$

$$\sum_{j=1}^{m_i} p_i^j \geq 1 \ \ \forall i : Y_i = 1. \quad (6)$$

One possibility to solve this problem is to alternate between updating $\{w^*, b^*\} = \arg\min_{w,b} \mathcal{L}_T(w, b, p^*)$ and $p^* = \arg\min_p \mathcal{L}_T(w^*, b^*, p)$ until we converged to a new (local) minimum. For a fixed p the SVM parameters can be found using any quadratic program solver. For example one can simply duplicate the patterns from the positive labeled bags (one with a positive label and one with a negative) and use two different costs for each pattern, namely Cp_i^j and $C(1-p_i^j)$. To find the optimal value of p we write the dual function of the program

$$\mathcal{L'}_T(p,\lambda) = \mathcal{L}_T(w,b,p) - \sum_{i=1}^{N}\lambda_i(\sum_{j=1}^{m_i} p_i^j - 1), \ \text{s.t.} \ \lambda_i \geq 0. \quad (7)$$

Taking the derivative w.r.t. p and equating to zero yields the following expression for the optimal p while fulfilling the constraints

$$p_i^j(\lambda_i) = \sigma\left(\frac{-Cd_i^j + \lambda_i}{T}\right), \quad (8)$$

where d_i^j is the difference of positive and negative loss, i.e. $d_i^j = l(\langle w, x_i^j\rangle + b) - l(-\langle w, x_i^j\rangle - b)$ and $\sigma(t) = (1+\exp(-t))^{-1}$ denotes the sigmoid function. The solution will always satisfy $0 \leq p_i^j \leq 1$. The Lagrange multiplier λ_i couples only variables within a bag and therefore the optimization for p_i^j can be done in parallel for all the bags. To solve for p_i^j one can check if $\sum p_i^j(0) \geq 1$ in which case the constraint is satisfied and thus $\lambda_i = 0$. Otherwise we know that $\sum_j p_i^j(\lambda_i) = 1$ which implies $p_i^j = \sigma\left(\frac{-Cd_i^j}{T}\right) / \sum_j \sigma\left(\frac{-Cd_i^j}{T}\right)$. Therefore the calculation of the new assignments for p can be done very efficiently and does only incur marginal costs compared to the quadratic programs one has to solve at each iteration. The quadratic program can be initialized with the solution from the previous iteration to speed up convergence. In order to start with an easy convex program we have to choose T_0 to ensure that we start with high entropy distributions. We found that choosing $T_0 = 10C$ is sufficient to ensure $p_i^j \approx 0.5$ in all experiments we conducted. The resulting algorithm for AL-SVM is summarized in Algorithm 1.

We would like to emphasize that we are minimizing the same objective function with the same constraints as the program mi-SVM. However the initialization of the algorithm is different, and this fact influences the type of local minima which are found. The optimization procedure proposed in [Andrews et al., 2002] initializes pattern labels to be identical to the bag label. We observed that for each iteration of their algorithm only some labels are changed and the whole algorithm is biased toward solutions with a large number of positive labeled points. Their algorithm is equivalent to the DA algorithm if the initialization $p_i^j = (y_i + 1)/2$ is used and one starts with $T_0 \approx 0$. In the experiments we use $T_0 = 10^{-8}$ to emulate this case.

4.2 Deterministic Annealing for SVMs identifying the witness

The two other methods build classifiers on the believed witnesses of the bags (the following derivation easily extends to the case when it is known that more than one positive instance resides in a bag). In the MI-SVM the most positive pattern is chosen to be this witness while in the MICA this integer representation is replaced by a convex combination of the points. We use a distribution over the patterns which leads to a convex combination of the costs. Each instance has a belief of being the witness of the bag which

Algorithm 1 Deterministic Annealing for identifying all Labels (AL-SVM)

Initialize $p_i^j = \frac{1}{2}$ if $Y_i = 1$, 0 otherwise.
Initialize $T = 10C$ (relatively high temperature)
while $S(p) > \epsilon$ **do**
  repeat
    compute w, b using quadratic problem solver
    set $q = p$
    compute p by Eq.(8) (and solve in λ, cf text)
  until $KL(p, q) < \epsilon$
  set $T = T/1.5$
end while

is encoded in p_i^j. To ensure that we have a probability distribution over each bag we have to add the constraint $\sum_{j=1}^{m_i} p_i^j = 1, \ \forall i : Y_i = 1$. The probability space $\mathcal{P}$ includes all distributions over the patterns in each bag. We set $p_i^j = 1$ for instances from negative labeled bags, effectively treating them as bags with a single pattern, and keep these values fixed (In fact we are sure that those patterns are witnesses of their bag label). In this case DA translates the objective function Eq.(1) into

$$\begin{aligned}\mathcal{L}_T(w, b, p) &= \|w\|_2^2 + C \sum_{i=1}^{N} \sum_{j=1}^{m_i} p_i^j l(Y_i(\langle w, x_i^j \rangle + b)) \\ &\quad + T \sum_{i=1}^{N} \sum_{j=1}^{m_i} p_i^j \log p_i^j \qquad (9)\end{aligned}$$

$$\text{(AW-SVM)} \quad \text{s.t.} \quad \sum_{i=1}^{m_j} p_i^j = 1 \ \ \forall i : Y_i = 1. \qquad (10)$$

We iterate between updating the search for optimal parameters w and p at a temperature T in the same way we did for the AL-SVM. Each instance enters the objective function with an individual weight Cp_i^j. Again taking the dual function to Eq.(9) and equating its derivative to zero we obtain an analytic solution for p

$$p_i^j = \exp\left(-\frac{Cl_i^j}{T}\right) / \sum_k^{m_i} \exp\left(-\frac{Cl_i^k}{T}\right), \qquad (11)$$

where we abbreviated $l_i^j = l(Y_i(\langle w, x_i^j \rangle + b))$. Thus similar to the AL-SVM updates of p_i^j are of only marginal computational cost. A high value of T favors high entropy distributions, in this case $p_i^j \approx 1/m_i$. The lower the value of T the more will p_i^j be concentrated on patterns which occur a low cost. In the extreme $T \to 0$ only points with $l_i^j = 0$ will have a $p_i^j > 0$, or if all points in a bag have a positive cost the one with minimum loss will be picked as the witness of the bag label. The latter case is exactly the same solution found by MI-SVM and MICA. Again $T_0 = 10C$ was used as a starting value for the temperature. The complete algorithm for annealing based on witnesses is summarized in Algorithm 2.

The formulation we obtained here is a deterministic annealing version of MICA. This is seen by taking $T = 0$ and identifying ν_i^j with p_i^j. We are optimizing the same objective function using a more sophisticated algorithm as originally proposed in [Mangasarian and Wild, 2005]. Additionally the entropy term determines the choice of the convex combinations in cases when there are more possibilities for the MICA. In contrast to the MI-SVM and MICA the DA algorithm might lead to more than one witness.

Algorithm 2 Deterministic Annealing for identifying the Witness (AW-SVM)

Initialize $p_i^j = \frac{1}{|X_i|}$,if $Y_i = 1$ and $p_i^j = 1$,if $Y_i = -1$
Initialize $T = 10C$
while p changed in the inner loop **do**
  repeat
    compute w, b using quadratic problem solver
    set $q = p$
    Set p according to Eq.(11)
  until $KL(q, p) < \epsilon$
  set $T = T/1.5$
end while
set $y_i^j = 1$ for all $p_i^j > \epsilon$ and update w, b using this assignment

5 Experiment: 2D Toy dataset

To compare the differences between the algorithm from Andrews et al. [2002] and the annealing algorithm we conducted an experiment using synthetic 2D data. This way we can control the number of instances with a positive label in the positive labeled bags and therefore test the results for pattern and bag accuracy. We created ten different type of datasets by varying the fraction of positive labeled points per bag over $f = 0.1, 0.2, \ldots, 1$. A bag was generated in the following way. The label Y_i and the size m_i are uniformly sampled from $\{-1, 1\}$ and $\{1, 2, \ldots, 10\}$. For a negative labeled bag we sample m_i patterns uniformly from the black region (negative class) in the leftmost picture in Figure 1. A positive labeled bag consists of $\lceil fm_i \rceil$ points sampled uniformly from the white region (positive class) and the remaining $\lfloor (1-f)m_i \rfloor$ points from the negative class. For each fraction f we sampled 30 training and 100 test bags. The hyperparameters were fixed to $C = 100$ and $\sigma = 1$ in the radial basis function kernels. Using this data AL-SVM and AW-SVM were trained with $T = 10^{-8}$ and $T = 10C$. The averaged results over 50 independent runs are shown in the two plots on the right in Figure 1. Matlab code used for the experiments is available online at `http://www.kyb.mpg.de/bs/people/pgehler/mil/mil.html`

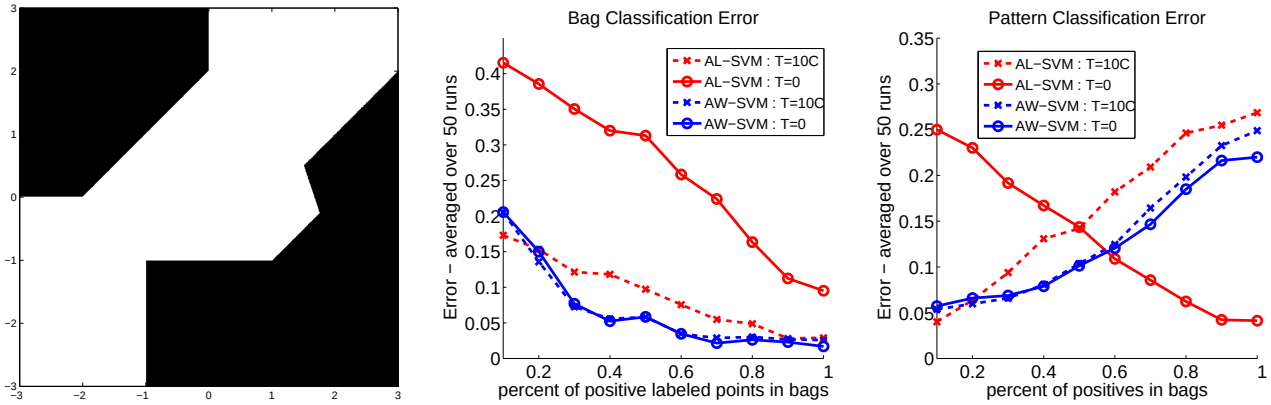

Figure 1: A 2D toy dataset. From left to right: Regions of positive (white) and negative (black) patterns (ground truth), Bag classification error averaged over 50 runs. Pattern classification error averaged over 50 runs.

These simple experiments reveal an important property of the algorithms. In the case of little ambiguity the non-annealed versions of the AL-SVM which are equivalent to the mi-SVM formulation give low error rates in pattern label accuracy (right). On the other side if only few points per positive labeled bag are of the positive class the error rates are very high. This result stands in contrast to the annealed versions where the reversed behavior is observed. Using the DA algorithm with $T = 10C$ the classification rates on the bag label are better than their non-annealed counterparts and also yield lower values of the objective function. This behavior can be explained by observing that the mi-SVM initializes all pattern labels to be positive and therefore tends to local minima close to this initialization. Hereby this algorithm overestimates the number of positive labeled points in a bag. The annealed version however has a problem as well, it underestimates this number.

The results for the annealed and non-annealed AW-SVM do not differ in this toy example. For this data set both methods seem not so prone to local minima and it is more or less irrelevant which patterns are identified as the witness.

6 A new objective function - ALP-SVM

The findings in the previous section raises the question of whether the objective function Eq.(1) is suited for the MIL problem. The alternating algorithm from Andrews et al. [2002] overestimates the number of positive labeled points in a bag which is a result of the initialization and the algorithm it is solved with. The annealing on the other hand is not biased towards a low or a high number of positive points. However the previous experiments show that it suffers from the problem of underestimation which is an indication that it is the objective function which is inadequate. This motivates the following extension of the objective function. We replace Eq.(1) by

$$\begin{aligned} \mathcal{L}'(w, b, \xi_i, \{y_i^j\}) &= \mathcal{L}(w, b, \xi_i, \{y_i^j\}) \qquad (12) \\ \text{(ALP-SVM)} \qquad &+ C_2 \sum_i \left(\sum_j^{m_i} \frac{y_i^j - 1}{2} - m_i p_i^* \right)^2 , \end{aligned}$$

with the constraint set of (mi-SVM). The new hyperparameter p_i^* can be used to control the expected number of positive labeled points per bag. Assignments $\{y_i^1, \ldots, y_i^{m_i}\}$ which deviate from this fraction are penalized. For bags with a negative bag label we set $p_i^* = 0$ because we do not expect any positive labeled points in these bags. This way an over- and underestimation of the fraction of positive labeled points per bag can be avoided, similar to a balancing constraint in semi-supervised learning. A balancing constraint ensures that the fraction of positive to negative labeled points estimated on the unlabeled point set is the same as that from the labeled training examples. This quantity can therefore be estimated from the training set where in the MIL setting there is ambiguity of the data and therefore no obvious way of how to choose this value. The value for p_i^* can either be prefixed due to prior knowledge or be left open as a hyperparameter estimated via cross validation. As the number of parameters to be estimated scales with the number of positive bags we will simplify by setting $p_i^* = p_j^* \; \forall i, j : Y_i = Y_j$.

The objective function Eq.(12) can easily be optimized using deterministic annealing. Replacing the integer values y_i^j by introducing probabilities for their assignment the new term in the objective function of ALP-SVM translates to

$$C_2 \sum_i \left(\sum_j^{m_i} p_i^j - m_i p_i^* \right)^2 \qquad (13)$$

The only difference to the DA algorithm for AL-SVM is the update of the probabilities p. Again the parameters can be optimized for each bag independently. For a fixed set of SVM parameters α, b we solve Eq.(12) including an entropy term for p_i using conjugate gradient ignoring the constraint $\sum_j p_i^j \geq 1$ (Eq.(6)). If a solution does not satisfy Eq.(6), i.e. is outside the feasible region of ALP-SVM we know that a solution of the constraint problem will lie on the simplex $\sum_j p_i^j = 1$. In this case Eq.(13) is simply a constant and thus the solution is the same as for the AL-SVM.

7 Experiment: Benchmark datasets

For a comparison of the proposed algorithms to those published in the literature and especially the SVM programs described above we ran experiments on some benchmark datasets for the MIL problem. We used the MUSK and the COREL datasets (Tiger,Elephant,Fox) used in [Andrews et al., 2002][1].

Again a first set of experiments was run to compare deterministic annealing to the alternating heuristic algorithm. As already noted the AL-SVM is equivalent to the mi-SVM if the temperature is set to a very low value. We used again $T_0 = 10^{-8}$ and initialized all pattern labels to be the same as the bag label to obtain the results for this optimization technique. Note that the published results of the mi-SVM and MI-SVM are obtained using l_2 penalization and the hinge loss function. For MICA the l_1 norm of the weights was used as a regularizer together with the hinge loss and therefore those published results can not be compared if one wants to judge the quality of the algorithm. To unify the presentation we ran all experiments using quadratic loss function and l_2 penalization of the weights.

We used an RBF kernel and set the bandwidth to the the median of the pairwise pattern distances denoted by σ_{emp}. The remaining hyperparameters were optimized using 10 fold cross validation where we searched over the grid $C \in \{1, 10\}$, $C_2 \in \{1, 10\}$ and $p^* \in \{0.1, .0.2, \ldots, 1\}$. The results are shown in Table 1. In addition to the cross validation error we also report the average fraction of estimated positive instances in a positive bag $\hat{p}$.

On all dataset except Fox we observe the same behavior as in the 2D toy example, that setting $T = 0$ leads to high values of $\hat{p}$ while setting $T = 10C$ yields a low value of $\hat{p}$. The ALP-SVM penalizes deviation from the prespecified fraction p^* and therefore overcomes this problem by finding solutions which lie "in between". Using the new objective function we obtain better results on the COREL datasets. For the MUSK1 dataset there was no better solution found than setting all pattern labels positive, a solution also found by the ALP-SVM.

[1] `www.cs.columbia.edu/~andrews/mil/datasets.html`

	T=0		T=10C		T=10C,p^*	
	err	$\hat{p}$	err	$\hat{p}$	err	$\hat{p}$
Tiger	25.0	79%	30.5	19%	**14**	60%
Fox	43.5	60%	38.5	16%	**35**	72%
Elephant	24.0	91%	30.5	14%	**16.5**	58%
Musk1	**14.3**	100%	20.6	38%	**14.3**	99%

Table 1: Results on several benchmark datasets. Standard deviation of the 10x fold cross validation error is usually around 3.5%

A final set of experiments was done on all the datasets described above as well as on MUSK2. We used the same grid of hyperparameters as in the initial experiments but now also varied the width of the kernel bandwidth in the interval $\sigma \in \{\sigma_{\text{emp}}, 2\sigma_{\text{emp}}, 0.5\sigma_{\text{emp}}\}$. However best performance was almost always obtained using using the initial bandwidth σ_{emp}. The final results together with those reported in Zhang et al. [2002], Mangasarian and Wild [2005], Andrews et al. [2002] are shown in Table 2. Note that the results obtained using MICA,MI-SVM and AW-SVM on the one and mi-SVM and AL-SVM on the other hand despite their similarity vary quite a bit. We therefore suspect that the datasets are very sensitive to model selection. The fractions of positive points per positive labeled bag for the best solution of the ALP-SVM for all datasets are also shown in Table 2.

The experiments show that DA does not help for the formulations identifying the witness and for the MUSK datasets even worsen performance. However we have to emphasize that using DA one always achieves a lower value of the objective function (numbers not reported here). There are two possible explanations to this phenomenon. Either the objective function is not suited for these particular datasets or it is more or less irrelevant which witnesses are identified. The objective functions could be easier to optimize in this case.

The results of the ALP-SVM are promising. Using this formulation the under/overestimation of $\hat{p}$ is overcome and a better local minima of the objective function directly translates into better classification performance. In the direct comparison with an annealed and non-annealed AL-SVM the cross validation error is lower on all datasets. As the results using the ALP-SVM are better than those from AW-SVM,MI-SVM and MICA (except MUSK2) it seems that the latter methods waste information by using only one point per bag for building the decision function. They could in principle benefit if they are able to identify witnesses in a positive labeled bag more reliably.

	EMDD	MI-SVM	MICA	AW-SVM		mi-SVM	AL-SVM		ALP-SVM	
MUSK1	15.2	22.1	15.6	14.3	20.6	**12.6**	14.3	20.6	13.7	$\hat{p} = 1$
MUSK2	15.1	15.7	**9.5**	16.2	20.8	16.4	17.4	13.8	13.8	$\hat{p} = 0.28$
Tiger	27.9	16	18	17	17	21.6	21.5	28	**14**	$\hat{p} = 0.6$
Elephant	21.7	18.6	17.5	18	19	17.8	20.5	29	**16.5**	$\hat{p} = 0.58$
Fox	43.9	42.2	38	36.5	37	41.8	36.5	37	**34**	$\hat{p} = 0.71$

Table 2: Results on several benchmark datasets. Left column in AW-SVM and AL-SVM are results obtained with $T_0 = 10^{-8} \approx 0$, whereas the right column states the result for $T = 10C$. The standard deviation of the 10x fold error is usually around 3.5% for our experiments.

8 Discussion and Conclusion

We presented the deterministic annealing algorithm for SVM formulations of the MIL problem. This method consistently finds better local minima of the objective functions. Furthermore we reported results which led to the conclusion that the algorithm of the mi-SVM as presented in [Andrews et al., 2002] is heavily biased towards problems with very little ambiguity. The deterministic annealing algorithm can solve this problem but comes with the problem of underestimating the number of positive points. This behavior renders both algorithms inapplicable for a general class of datasets.

We proposed the ALP-SVM, an extension of the mi-SVM objective function which opens up the possibility to encode prior knowledge about the dataset in a principled way. The deterministic annealing algorithm allows to optimize the new objective function at about the same computational cost as the AL-SVM. For the COREL datasets we obtain the best results reported so far in the literature. It is our belief that a new set of benchmark datasets are needed to compare all proposed MIL algorithms thoroughly. These dataset should have different levels of ambiguity in order to identify algorithms which are too sensible to this property.

The AW-SVM can be extended to the case of more witnesses per positive labeled bag. This can be helpful in object recognition when multiple instances of an object are present in an image. In this case the heuristic learning method for MI-SVM and MICA might be more prone to local minima than the deterministic annealing scheme. For the latter it is also straightforward to derive a formulation which penalizes deviation from some prespecified number of witnesses, similar to the ALP-SVM.

Acknowledgments

This work was supported in part by the research project CLASS (IST project 027978) and the IST Programme of the European Community under the PASCAL Network of Excellence IST-2002-506778. This publication only reflects the authors' views.

References

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Neural Information Processing Systems*, pages 561–568, 2002.

Yixin Chen, Jinbo Bi, and James Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12), 2006.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. Multi-instance kernels. In *Proc. 19th International Conf. on Machine Learning*, pages 179–186, 2002.

Olvi L. Mangasarian and Edward W. Wild. Multiple instance classification via successive linear programming. Technical Report 05-02, Data Mining Institute, University of Wisconsin, 2005.

Joseph F. Murray, Gordon F. Hughes, and Kenneth Kreutz-Delgado. Machine learning methods for predicting failures in hard drives: A multiple-instance application. *Journal of Machine Learning Research*, 6:783–816, 2005.

Soumya Ray and Mark Craven. Supervised versus multiple instance learning: an empirical comparison. In *Proc. 22nd International Conf. on Machine Learning*, pages 697–704, 2005.

Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of IEEE*, volume 86, pages 2210–2239, 1998.

Vikas Sindhwani, S. Sathiya Keerthi, and Olivier Chapelle. Deterministic annealing for semi-supervised kernel machines. In *Proc. 23rd International Conf. on Machine Learning*, 2006.

Qingping Tao, Stephen Scott, N. V. Vinodchandran, and Thomas Takeo Osugi. Svm-based generalized multiple-instance learning via approximate box counting. In *Proc. 21st International Conf. on Machine Learning*, page 101, 2004.

Paul A. Viola, John Platt, and Cha Zhang. Multiple instance boosting for object detection. In *Neural Information Processing Systems*, 2005.

Jun Wang and Jean-Daniel Zucker. Solving the multiple-instance problem: A lazy learning approach. In *Proc. 17th International Conf. on Machine Learning*, pages 1119–1126, 2000.

Qi Zhang, Sally A. Goldman, Wei Yu, and Jason Fritts. Content-based image retrieval using multiple-instance learning. In *Proc. 19th International Conf. on Machine Learning*, pages 682–689, 2002.