A Conditional Random Field for Multiple-Instance Learning

Thomas Deselaers DESELAERS@VISION.EE.ETHZ.CH
Vittorio Ferrari FERRARI@VISION.EE.ETHZ.CH
Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland

Abstract

We present MI-CRF, a conditional random field (CRF) model for multiple instance learning (MIL). MI-CRF models bags as nodes in a CRF with instances as their states. It combines discriminative unary instance classifiers and pairwise dissimilarity measures. We show that both forces improve the classification performance. Unlike other approaches, MI-CRF considers all bags jointly during training as well as during testing. This makes it possible to classify test bags in an imputation setup. The parameters of MI-CRF are learned using constraint generation. Furthermore, we show that MI-CRF can incorporate previous MIL algorithms to improve on their results. MI-CRF obtains competitive results on five standard MIL datasets.

1. Introduction

The aim of multiple instance learning (MIL) is to classify *bags* of *instances*. During training, the learner is given a set of *positive* and *negative training bags*. A bag is a collection of feature vectors called *instances*. During testing, a new *test bag* is classified to be either positive or negative.

In this paper, we adopt the definition of Dietterich et al. (1997) (and many others): a bag is positive if it contains at least one positive instance and negative if it contains only negative instances. In this scenario, MIL is a learning problem with incomplete data where the class labels of the instances are latent variables. In fact, as part of the learning process many methods also label the instances of the training bags (Andrews et al., 2002; Bunescu & Mooney, 2007; Dietterich et al., 1997; Fung et al., 2006; Gehler & Chapelle, 2007; Maron & Lozano-Perez, 1997; Viola et al., 2005; Zhou & Xu, 2007) although this is not required in the original problem definition.

A number of MIL applications have been identified in the literature, e.g. object detection in images (Viola et al., 2005), content-based image retrieval (Zhang et al., 2002), and protein identification (Tao et al., 2004).

In this paper we propose MI-CRF, which casts MIL into a conditional random field (CRF) framework (Lafferty et al., 2001). Bags correspond to the nodes of the CRF. The instances in a bag form the states a node can take. Classification corresponds to selecting one instance (positive bag) or selecting none (negative bag). We formulate this selection as inference in the CRF, which is naturally suited to select one state out of a discrete set for each node. It also enables us to consider all bags jointly and naturally combines different forces such as discriminative instance classifiers (sec. 2.2) and pairwise dissimilarity measures between instances from different bags (sec. 2.3). These forces are combined in a principled manner by learning the parameters of the CRF using constraint generation (sec. 3.2). MI-CRF obtains results competitive to the state of the art on five standard MIL datasets. Additionally, MI-CRF naturally lends itself to imputation, i.e. classifying multiple test bags jointly, and can incorporate other MIL algorithms as unary potentials.

Related Work

The mi-SVM and MI-SVM approaches by Andrews et al. (2002) extend support vector machines (SVMs) to MIL. They differ in the way they deal with the instances. MI-SVM considers only the most positive instance of a bag during training while mi-SVM considers all instances. Gehler & Chapelle (2007) show that deterministic annealing (DA) leads to better local optima in training and that varying the fraction of the instances considered positive in a positive bag can improve results. Similarly to MI-SVM, our model

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

maintains a latent selection of the most positive instance within a positive bag.

Bunescu & Mooney (2007) consider the special case where positive instances are sparse within the bags and modify the SVM training criterion such that instances are classified far away from the margin.

Maron & Lozano-Perez (1997) address MIL by finding regions in the instance space with instances from many different positive bags and few instances from negative bags. These are regions of diverse density (DD). Zhang & Goldman (2001) refined their learning algorithm using expectation maximization (EM).

Most of these approaches maintain a latent selection of positive instances in the training bags which inspired Zhou & Xu (2007) to show that MIL and semi-supervised learning are closely related. They consider the instances within positive bags as unlabeled with the constraint that at least one of them must be positive. They formulate a semi-supervised learning problem and use this to classify all instances from the positive bags. This reduces MIL to single instance classification.

Wang et al. (2008) do not start from the standard MIL definition but allow for other setups in order to give more flexibility to their model. They model the composition of instances into bags using mixtures and define a kernel mapping that combines the posteriors of the mixtures to determine bag classes.

Recently, (Zhou et al., 2009) proposed two new methods for MIL that represent the bags as graphs and explicitly model the relationships between the instances within a bag. Instead, our MI-CRF models the relationships between instances from different bags.

In MI-CRF we consider all bags jointly and combine discriminative unary instances classifiers with pairwise dissimilarities between instances into the decision process. As demonstrated in the experiments (sec. 5), both components contribute to classification performance. The pairwise dissimilarities help classifying correctly (a) positive test bags because they contain some instances similar to instances in positive training bags; (b) negative test bags because most of their instances are dissimilar from instances in positive training bags.

Furthermore, we show that MI-CRF can incorporate as an additionally unary potential any previous MIL method capable of scoring individual instances. We demonstrate experimentally that incorporating MI-SVM improves on the results of either method alone.

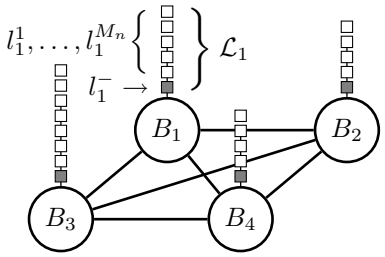

Figure 1. **MI-CRF:** Bags B_n are nodes in a fully connected CRF. The instances l_n (white squares) and the non-instance l_n^- (gray square) are the states a node can take.

2. The MI-CRF model

The goal is to classify *bags* of *instances* as either positive or negative. Since a bag is positive if at least one of its instances is positive, it is sufficient to identify one positive instance to classify a bag as positive. If a bag contains no positive instance, then it is negative.

In MI-CRF, classifying a bag corresponds to selecting an instance (for positive bags) or no instance (for negative bags) by minimizing an energy function defined over all bags. Ideally the energy is minimal when an instance is selected in every positive bag and no instance is selected in any negative bag.

Using CRFs for MIL is made possible by recent CRF inference algorithms that allow for efficiently finding good approximations of the optimal selection of states (Kolmogorov, 2006). Moreover, MI-CRF can be trained in a principled manner by constraint generation (Tsochantaridis et al., 2005).

2.1. Bag Classification as Instance Selection

The set of training bags $\mathcal{B} = (B_1, \ldots, B_N)$ is represented as a fully connected CRF (fig. 1). Each bag B_n is a node which can take on a state from a discrete set $\mathcal{L}_n = \{l_n^1, \ldots, l_n^{M_n}, l_n^-\}$ corresponding to its M_n instances $l_n^1, \ldots, l_n^{M_n}$ and a special *non-instance state* l_n^-.

A configuration $L = (l_1, \ldots, l_N)$ with $l_n \in \mathcal{L}_n$ selects either an instance or the non-instance state for each bag. This also induces a classification of bag B_n as positive if it selects an actual instance l_n, or as negative if it selects the non-instance l_n^-.

The posterior probability for configuration L is

$$p(L|\mathcal{B}, \Theta) \propto \exp\left(-E(L|\mathcal{B}, \Theta)\right), \quad \text{with} \tag{1}$$

$$E(L|\mathcal{B}, \Theta) = \sum_n \Phi(l_n|B_n, \Theta) + \sum_{n,m} \Psi(l_n, l_m|B_n, B_m, \Theta) \tag{2}$$

Algorithm 1 Training and testing with MI-CRF

(a) Training (given training bags $\mathcal{B}$): (sec. 2, 3)
1: Learn unary instance SVM classifiers Υ_f and derive unary non-instance energies Ω_f from them (sec. 3.1).
2: Learn non-instance prior Π and weights α using constraint generation (sec. 3.2).
3: Determine latent configuration $\hat{L}$ (sec. 3.2).

(b) Testing (for a single test bag B_T): (sec. 4.1)
1: Clamp training bags to $\hat{L}$.
2: Evaluate instance classifiers Υ_f for all instances in B_T.
3: Derive non-instance energy Ω_f for l_T^-.
4: Compute pairwise potentials Γ_g between $l_T \in B_T$ and clamped training instances.
5: Select the (non-)instance minimizing eq. (10) and classify B_T accordingly.

where Φ is a unary potential (sec. 2.2), Ψ is a pairwise potential (sec. 2.3), and Θ are the parameters of MI-CRF.

Inference on this CRF determines the configuration L^* minimizing eq. (2)

$$L^* = \arg\min_L \{E(L|\mathcal{B}, \Theta)\} \tag{3}$$

To determine L^* we use tree-reweighted message passing (Kolmogorov, 2006). Inference is the key operation during testing, where we are interested in the state of a test bag (sec. 4), and also plays an important role during training, as MI-CRF maintains a latent selection of the most positive instance in each positive training bag (sec. 3).

Algorithm 1 gives an overview of how MI-CRF is trained and how it is used for testing.

2.2. Unary Potential Φ

The unary potential Φ assigns an energy to each state in $\mathcal{L}$. For the states representing actual instances l_n it measures how likely these are to be positive. For the non-instance state it measures how likely the *bag* is to be negative (i.e. that *all* instances within it are negative).

In a positive bag, it should give low energy to positive instances and high energy to the non-instance. In a negative bag, it should give low energy to the non-instance and high energies to the actual instances (as they are all negative).

The unary potential is defined as

$$\begin{aligned}\Phi(l_n|B_n, \Theta) = \quad & \sum_f \quad \Big[\alpha_{\Upsilon_f} \Upsilon_f(l_n|B_n, \theta_{\Upsilon_f}) \\ & \qquad + \alpha_{\Omega_f} \Omega_f(l_n|B_n, \theta_{\Omega_f})\Big] \\ & + \Pi(l_n|\alpha_\Pi)\end{aligned} \tag{4}$$

It is a linear combination of (a) several instance classifiers Υ_f measuring how likely individual instances are to be positive[1]; (b) corresponding non-instance models Ω_f derived from the instance classifiers; they measure how likely the bag is to be negative; (c) a prior Π for the bag to be negative. The scalars α weight the terms.

Instance classifiers Υ_f encode the likelihood for an *instance* to be positive. $\Upsilon_f(l_n|B_n, \theta_{\Upsilon_f})$ is the signed distance of instance l_n from an SVM hyperplane θ_{Υ_f}. For the non-instance l_n^-, we set $\Upsilon(l_n^-|B_n, \theta_{\Upsilon_f}) = 0$. The different Υ_f use different kernels f ($f \in \{$RBF, linear, histogram intersection$\}$).

Non-instance models Ω_f encode the likelihood for a *bag* to be negative. They are derived from the instance classifiers Υ_f. Since a bag B_n is positive if it has at least one positive instance l_n, we define the likelihood for it to be negative to be inversely proportional to the likelihood of its highest scores instance to be positive (according to Υ_f).

More precisely, for each instance classifier Υ_f we create a corresponding non-instance model Ω_f:

$$\begin{aligned}&\Omega_f(l_n|B_n, \theta_{\Omega_f}) = \\ &\quad \begin{cases} -\min\limits_{l_n \in B_n} \{\Upsilon_f(l_n|B_n, \theta_{\Upsilon_f})\} & \text{for } l_n = l_n^- \\ 0 & \text{otherwise} \end{cases}\end{aligned} \tag{5}$$

The min-operation in the non-instance models Ω_f explicitly captures the asymmetry in the problem. Identifying a single positive instance dismisses the negative class label from a bag no matter how many negative instances it contains.

Non-instance prior Π. We define the prior for a bag to be negative as $p(l_n = l_n^-) \propto \exp(-\alpha_\Pi)$, which is incorporated into the model as an energy term of the form

$$\Pi(l_n|\alpha_\Pi) = \begin{cases} \alpha_\Pi & \text{if } l_n = l_n^- \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

with α_Π a parameter of MI-CRF.

2.3. Pairwise Potential Ψ

The pairwise potential Ψ measures the dissimilarity between two instances $l_n \in B_n$ and $l_m \in B_m$ capturing how likely they are to be from the same class

$$\Psi(l_n, l_m|B_n, B_m, \Theta) = \sum_g \alpha_{\Gamma_g} \Gamma_g(l_n, l_m|B_n, B_m) \tag{7}$$

[1] Note how for the simple case when instances from positive and negative bags are separable, the instance classifiers alone will already solve the MIL task.

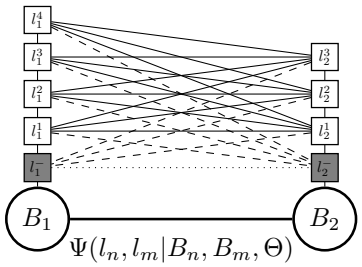

Figure 2. **The pairwise term between two bags** consists of dissimilarity between instances (solid lines). The connections between instances and non-instances (dashed lines) and the connection between the non-instances (dotted line) are set to 0.

It is a linear combination of terms Γ_g measuring the dissimilarity between l_n and l_m according to various measures g. The scalars α_{Γ_g} weight the terms. Apart from α, the pairwise potential involve no other parameters.

Fig. 2 illustrates the computation of the pairwise potential for every pair of instances between two bags.

Roughly following Tolstoy (1873), we assume that positive instances are all similar whereas every negative instance is negative in its own way[2]. Thus, we expect the distance between two positive instances to be small. But we cannot assume anything about the negative instances (e.g. they might be uniformly distributed over the entire instance space), so we set $\Gamma(l_n^-, l_m | B_n, B_m, \theta_{\Lambda_g}) = \Gamma(l_n, l_m^- | B_n, B_m, \theta_{\Lambda_g}) = \Gamma(l_n^-, l_m^- | B_n, B_m, \theta_{\Lambda_g}) = 0$. Therefore, on its own, the pairwise potential is biased toward selecting the non-instance state. However, the other forces in the model counterbalance this, i.e. the instance classifiers Υ_f, the non-instance models Ω_f, and especially the non-instance prior Π. In sec. 5.1 we show that the resulting model is well balanced and chooses between instances and the non-instance state appropriately.

As dissimilarity measures g we use the L_1, L_2, and χ^2 distances.

Note that our models connects instances from different bags rather than instances within a bag (as in (Zhou et al., 2009)).

3. Training MI-CRF

To train MI-CRF we learn the parameters θ_{Υ_f} and the weights α from the training bags $\mathcal{B}$, such that running inference (eq. (3)) on $\mathcal{B}$ finds a configuration $\hat{L}$ that classifies all training bags correctly (sec. 3.2).

Training consists of two steps (algorithm 1): (1) setting up the unary energy terms (sec. 3.1), i.e. learning instance classifiers Υ_f and the corresponding non-instance models Ω_f (sec. 3.1); (2) learning the weights α to combine the terms and the non-instance prior α_Π. We learn the weights such that the lowest energy configuration $\hat{L}$ (eq. (2)) classifies all training bags correctly. We learn α using a constraint generation algorithm analogous to the structured-output SVM training (Tsochantaridis et al., 2005) (sec. 3.2).

3.1. Learning the Instance Classifiers Υ_f

To learn the instance classifiers Υ_f we train an SVM to separate all instances from all positive training bags from all instances from all negative training bags. To make it a suitable energy term in eq. (4), the SVM is trained to assign negative scores to instances from positive bags and positive scores to instances from negative bags. This corresponds to training an SVM with noisy labels: while we know that all instances in the negative bags are negative, among the instances from the positive bags there may be negative instances. However, as long as a significant portion of the instances in the positive bags are positive, the process will lead to a reasonable classifier performing better than chance (see row (a) in tab. 2). Indeed Bunescu & Mooney (2007) directly use Υ_f as a baseline called *single-instance learning SVM* and show that it obtains results comparable to early MIL approaches.

The non-instance energies Ω_f are directly derived from the Υ_f according to eq. (5).

3.2. Learning the Weights α and the Non-instance Prior Π

Given the unary and pairwise energy terms we learn weights α to balance their impacts[3]. The goal is to find a set of α such that the lowest energy configuration of eq. (2) classifies all training bags correctly. Note that there are exponentially many configurations L that classify all bags correctly (as well as exponentially many that classify all bags incorrectly). We want to find α so that there exists (at least) one configuration $\hat{L}$ that classifies all bags correctly with a lower energy than any configuration that does not.

We learn α w.r.t. a max-margin criterion, following the constraint-generation approach used to train struc-

[2] Note how the pairwise potential alone will already solve MIL in the simple case where all positive bags contains a similar positive instance and the negative bags only contain instances different from all positive ones.

[3] The form of Π in eq. (6) allows us to incorporate α_Π as an extra weight in eq. (4), and to learn it along with the other α.

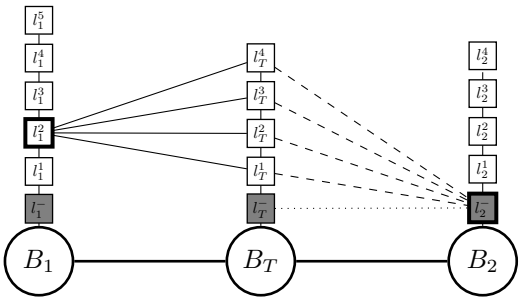

Figure 3. **Classifying a test bag B_T.** The positive training bag B_1 is clamped to its instance l_1^2 (bold), the negative training bag B_2 is clamped to its non-instance l_2^- (bold), as determined by $\hat{L}$ during training.

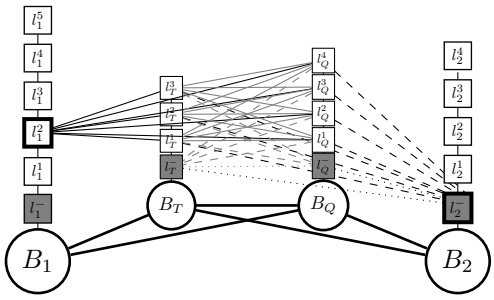

Figure 4. **Imputation: Jointly Classifying two test bags B_T and B_Q.** Analogously to fig. 3, the training bags are clamped according to the latent configuration $\hat{L} = (\hat{l}_1^2, \hat{l}_2^-)$. Gray lines are the connections between instances of the test bags.

tured output SVMs (Tsochantaridis et al., 2005) analogously to (Finley & Joachims, 2008; Szummer et al., 2008) who use a similar technique to learn the parameters of a CRF.

To learn the weights α we solve a generalized support vector machine training problem (Tsochantaridis et al., 2005):

$$\min_{\alpha,\xi} \frac{1}{2}||\alpha||^2 + C\xi \tag{8}$$

$$\text{s.t.} \quad E(\hat{L}|\mathcal{B},\Theta^\alpha) - E(L|\mathcal{B},\Theta^\alpha) \geq \Delta(\hat{L},L) - \xi, \forall L \neq \hat{L}$$
$$\xi \geq 0, \quad \alpha \geq 0$$

$C > 0$ is a constant controlling the trade-off between training error minimization and margin maximization, while ξ is a slack variable. Θ^α are the parameters of the CRF according to the current weight vector α, and $\Delta(\hat{L},L) = \sum_n \mathbf{1}(\hat{l}_n \neq l_n)$ is a 0/1 loss penalizing deviations from $\hat{L}$ (where $\mathbf{1}(c) = 1$ if c is true, and 0 otherwise). $\hat{L}$ is a latent configuration that classifies all training bags correctly and is updated in each training iteration.

In this formulation, every possible configuration L corresponds to a constraint so the number of constraints is exponential in the number of bags. It is infeasible to consider all constraints explicitly during optimization of (8).

Constraint generation only considers a small subset of the constraints explicitly. It starts with an empty set of constraints and in each iteration adds the configuration L^* that violates the constraints the most. This configuration can be found by solving a subproblem in the same form as eq. (3), but incorporating the loss $\Delta(\hat{L},L)$ as an additional unary term:

$$L^* = \arg\min_L \{E(L|\mathcal{B},\Theta) - \Delta(\hat{L},L)\} \tag{9}$$

After this subproblem is solved, we determine a new α by minimizing (8) including the new constraint L^*. We then update the CRF defined on the training bags using α and run inference (eq. (3)) on it to get the new latent configuration $\hat{L}$. Finally we correct $\hat{L}$ according to the class labels of the training bags (i.e. if $\hat{L}$ selects the non-instance in a positive bag, we correct it to pick the instance increasing the total energy the least).

$\hat{L}$ is initialized to select a random instance in each positive bag, and the non-instance in each negative bag. This procedure is iterated until no configuration exists that misclassifies a bag and which has a lower energy than the configuration $\hat{L}$ (which classifies all training bags correctly). In our experiments this typically took between 20 and 50 iterations. At this point training terminates, all constraints are fulfilled, and $\hat{L}$ contains a latent selection of an instance for each positive training bag.

We use tree-reweighted message passing (TRW-S) (Kolmogorov, 2006) to approximate the solution to the subproblem (eq. (9)). TRW-S decomposes an input problem into chains, solves them optimally, and then combines their solutions into a solution to the original problem. TRW-S maximizes a lower bound on the energy of the original problem and is guaranteed to converge. In our experiments we observed that the energy of the configurations found by TRW-S is on average only 5% higher than the lower bound. Thus our approximations are indeed close to the global optimum.

4. Classifying Test Bags

Given the fully trained MI-CRF model, we can classify new test bags individually (sec. 4.1, algorithm 1) or jointly in an imputation setup (sec. 4.2).

4.1. Classification of a Single Test Bag

In order to classify a test bag B_T we incorporate it as an additional node into MI-CRF and determine its class by selecting the state l_T that minimizes eq. (2). We clamp all training bags B_n to the state $\hat{l}_n$ from the latent configuration $\hat{L}$ determined during training (sec. 3.2). Testing is computationally cheap, because (a) only a small fraction of the pairwise potentials need to be computed, and (b) inference can be performed efficiently and exactly as

$$l_T^* = \arg\min_{l_T} \left\{ \Phi(l_T|B_T,\Theta) + \sum_n \Psi(l_T,\hat{l}_n|B_T,B_n,\Theta) \right\} \tag{10}$$

which only requires $O(N|\mathcal{L}_T|)$ operations (with N the number of training bags and $|\mathcal{L}_T|$ the number of instances in the test bag).

Fig. 3 shows the classification of a test bag with two training bags.

4.2. Joint Classification of Multiple Test Bags

The above procedure for classification naturally lends itself to imputation. Instead of classifying a single test bag, multiple test bags are classified jointly. In this case, additionally to the connections to the clamped training bags, we build full pairwise connections between the test bags. This encourages test bags with similar instances to be classified as positive (Tolstoy, 1873).

In this imputation setup the number of possible configurations is exponential in the number of test bags and thus inference cannot be performed exactly. Analogously as in training, we use TRW-S (Kolmogorov, 2006) to obtain a good approximation.

Fig. 4 shows the imputation setup with two test and two training bags.

5. Experiments

We present experiments on five standard MIL datasets (Andrews et al., 2002): musk (1 and 2) and the Corel datasets (Tiger, Elephant, and Fox) used in (Andrews et al., 2002; Bunescu & Mooney, 2007; Fung et al., 2006; Gehler & Chapelle, 2007; Maron & Lozano-Perez, 1997; Zhou & Xu, 2007; Zhou et al., 2009). Table 1 gives an overview of the datasets. Following the standard protocol, experiments are performed in 10-fold cross-validation and the per-fold average test classification performance is reported in table 2.

Table 1. **Datasets.** The columns mean: *bags pos/neg*: positive/negative bags in the dataset; *avg inst/bag*: average number of instances per bag; *dim*: dimensionality of the instance space.

dataset	bags pos/neg	avg inst/bag	dim
Musk 1	47/45	5.17	166
Musk 2	39/63	64.69	166
Elephant	100/100	6.96	230
Fox	100/100	6.60	230
Tiger	100/100	6.10	230

5.1. Qualitative Analysis

Fig. 5 shows a visualization of the energy terms of a trained MI-CRF model. For visualization, the model was reduced to five positive and five negative bags after training. Both unary and pairwise potentials show the desired structure: the unaries of the actual instances have low energies in the positive bags and high energies in the negative bags. Conversely, the non-instances have high and low energies respectively. The pairwise potentials are low between several pairs of instances from positive bags (corresponding to the positive instances). Instead they are mostly high between instances from positive and negative bags. Between pairs of instances from negative bags the energies are sometimes high and sometimes low (salt and pepper patterns). This confirms our conjecture from section 2.3 about the distribution of positive and negative instances.

As the figure also shows, the latent configuration $\hat{L}$ selects low energy instances in positive bags with rather low energy connections between each other.

5.2. Quantitative Analysis

Table 2 shows results for several variants of MI-CRF compared to (Zhang & Goldman, 2001; Andrews et al., 2002; Fung et al., 2006; Gehler & Chapelle, 2007; Wang et al., 2008; Zhou et al., 2009). To the best of our knowledge, miGraph (Zhou et al., 2009) obtains the best results published on these datasets.

First we show the impact of the individual components of our model. Setup (a) is a simple baseline only using one single instance classifier, i.e. an SVM trained to separate instances from positive and negative bags (sec. 3.1). If at least one instance from a test bag is classified as positive the entire bag is deemed positive, and negative otherwise. In setup (b), MI-CRF uses three different instance classifiers Υ_f, the corresponding non-instance models Ω_f, and the non-instance prior Π, but no pairwise term. All SVMs in

Table 2. **Results.** Bag classification accuracies [%] of MI-CRF on five standard MIL datasets compared to the state of the art. The results in the upper half are taken from the respective papers. See main text for discussion.

method	Musk 1	Musk 2	Elephant	Fox	Tiger	Average
EM-DD (Zhang & Goldman, 2001)	84.8	84.9	78.3	56.1	72.1	75.2
mi-SVM (Andrews et al., 2002)	87.4	83.6	80.0	57.9	78.9	77.6
MI-SVM (Andrews et al., 2002)[2]	77.9	84.3	73.1	58.8	66.6	72.1
MICA (Fung et al., 2006)	84.4	**90.5**	82.5	62.0	82.0	80.3
MI-SVM + DA (Gehler & Chapelle, 2007)	85.7	83.8	82.0	63.5	83.0	79.6
mi-SVM+DA+p (Gehler & Chapelle, 2007)	86.3	86.2	83.5	**66.0**	**86.0**	81.6
PPMM Kernel (Wang et al., 2008)	**95.6**	81.2	82.4	60.3	80.2	79.9
MIGraph (Zhou et al., 2009)	90.0	90.0	85.1	61.2	81.9	81.6
miGraph (Zhou et al., 2009)	88.9	90.3	**86.6**	61.6	**86.0**	**82.7**
(a) single instance SVM	78.3	71.6	73.0	58.0	75.5	71.3
(b) only $\Upsilon_f, \Omega_f, \Pi$	85.9	75.5	72.5	57.5	70.5	72.4
(c) full MI-CRF	87.0	78.4	**85.0**	**65.0**	79.5	79.0
(d) + imputation	87.0	78.4	**85.0**	**65.0**	80.0	79.1
(e) full MI-CRF + cross-val. C	**88.0**	**84.3**	**85.0**	63.5	**82.5**	**80.7**
(f) MI-SVM baseline[2]	81.5	**86.3**	82.0	61.5	82.0	78.7
(g) MI-CRF incorporating MI-SVM (cross-val. C)	**88.0**	85.3	**85.0**	**67.5**	**83.0**	**81.8**

[2] We observe that our reimplementation of MI-SVM performs better than what reported by (Andrews et al., 2002).

the experiments were trained using libSVM and its default parameters for $C = 1$ and kernel parameters (for the RBF kernel $\gamma = 1/D$, with D the dimensionality of the instance space) (Chang & Lin, 2001). Setup (c) is the full MI-CRF, adding three pairwise dissimilarities to setup (b).

As the table shows, MI-CRF substantially outperforms the baseline (a) and all components of the model contribute to the result. In particular notice the large improvement brought by the pairwise dissimilarity terms (from (b) to (c)). This confirms that pairwise dissimilarity complements well discriminative instance classifiers. Bringing the two components together is the main strength of MI-CRF.

In setup (d), the classification is performed using imputation, which MI-CRF supports naturally (sec. 4.2). This bring a minor improvement.

In these experiments the parameter C of the weight learning (8) was always set to 1. If we adapt C using cross validation within each of the training folds we obtain another improvement (from setup (c) to (e)). Notice how this fold-specific C is estimated using only the training bags within the fold.

In a second series of experiments we use MI-SVM (Andrews et al., 2002) as a baseline (setup (f)). Since MI-SVM is able to score each instance within a bag, it can be incorporated as an additional unary potential into MI-CRF (setup (g)). As the table shows, (g) improves over both MI-SVM alone (f) and MI-CRF (e). Our setup (g) outperforms all previous works but the very recent miGraph (Zhou et al., 2009). While we present experiments with MI-SVM, any other MIL approach that can score individual instances could easily be incorporated.

Runtime. On the musk1 dataset, inference using TRW-S takes approximately 0.03s. The total runtime of a cross-validation experiment (setup (c)) is about 200s, of which 120s are taken to compute the pairwise dissimilarities.

6. Discussion and Conclusion

We presented MI-CRF, a novel approach to MIL which represents bags as nodes in a CRF and instances as their states. It combines discriminative unary instance classifiers and pairwise dissimilarity measures. We experimentally demonstrated that both aid classification. MI-CRF can easily incorporate as additional unary potentials other MIL approaches that score individual instances and improves on their classification performance. Furthermore, MI-CRF naturally lends itself to imputation. Unlike other approaches, MI-CRF considers all bags jointly during training as well as during testing. MI-CRF obtains results competitive with the state of the art on five standard MIL datasets.

Acknowledgement. We thank Peter Gehler for providing his implementation of MI-SVM.

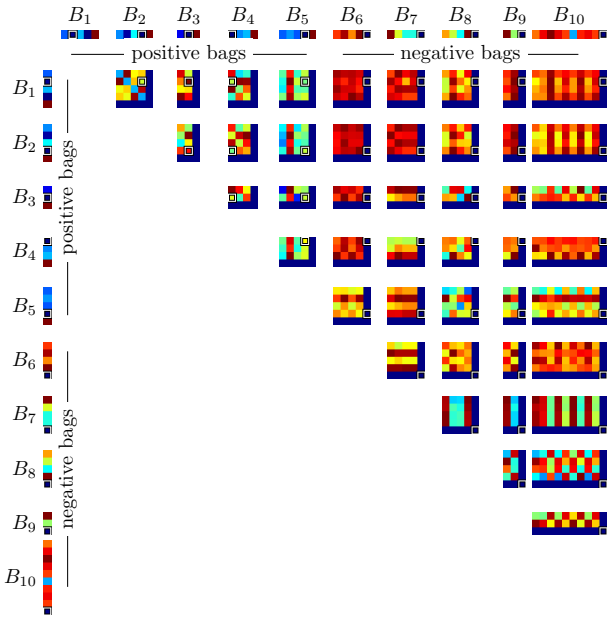

Figure 5. **MI-CRF on 10 training bags from Musk 1.** The first row and column show the unary potentials $\Phi(l_n|B_n, \Theta)$ (rightmost/bottom correspond to non-instances l_n^-). The matrix in column B_n, row B_m denotes the pairwise potential $\Psi(l_n, l_m|B_n, B_m, \Theta)$. Both potentials include several composing terms summed according to the learned α (eq. (4), eq. (7)). The latent selection $\hat{L} = (l_1^2, l_2^4, l_3^2, l_4^1, l_5^4, l_6^-, l_7^-, l_8^-, l_9^-, l_{10}^-)$ is denoted by black/white rectangles around the corresponding state (unary) or pair of states (pairwise). (Best viewed in color.)

References

Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

Bunescu, R. C. and Mooney, R. J. Multiple instance learning for sparse positive bags. In *ICML*, 2007.

Chang, C.-C. and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001.

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

Finley, T. and Joachims, T. Training structural svms when exact inference is intractable. In *ICML*, 2008.

Fung, G., Dundar, M., Krishnapuram, B., and Rao, R. B. Multiple instance learning for computer aided diagnosis. In *NIPS*, 2006.

Gehler, P. V. and Chapelle, O. Deterministic annealing for multiple-instance learning. In *AISTATS*, 2007.

Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568 – 1583, 2006.

Lafferty, J., McCallum, A., and Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, 2001.

Maron, O. and Lozano-Perez, T. A framework for multiple-instance learning. In *NIPS*, 1997.

Szummer, M., Kohli, P., and Hoiem, D. Learning CRFs using graph cuts. In *ECCV*, 2008.

Tao, Q., Scott, S., Vinodchandran, N., and Osugi, T. T. SVM-based generalized multiple instance learning via approximate box counting. In *ICML*, 2004.

Tolstoy, L. *Anna Karenina*. The Russian Messenger, 1873.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

Viola, P. A., Platt, J., and Zhang, C. Multiple instance boosting for object detection. In *NIPS*, 2005.

Wang, H.-Y., Yang, Q., and Zha, H. Adaptive p-posterior mixture-model kernels for multiple instance learning. In *ICML*, 2008.

Zhang, Q. and Goldman, S. A. EM-DD: An improved multiple-instance learning technique. In *NIPS*, 2001.

Zhang, Q., Goldman, S. A., Yu, W., and Fritts, J. Content-based image retrieval using multiple instance learning. In *ICML*, 2002.

Zhou, Z.-H. and Xu, J.-M. On the relation between multi-instance learning and semi-supervised learning. In *ICML*, 2007.

Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. Multi-instance learning by treating instances as non-I.I.D. samples. In *ICML*, 2009.