Robust Multiple-Instance Learning with Superbags

Borislav Antić and Björn Ommer

Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany
{borislav.antic,bommer}@iwr.uni-heidelberg.de

Abstract. Multiple-instance learning consists of two alternating optimization steps: learning a classifier with missing labels and finding the missing labels with the classifier. These steps are iteratively performed on the same training data, thus imputing labels by evaluating the classifier on the data it is trained upon. Consequently this alternating optimization is prone to self-amplification and overfitting. To resolve this crucial issue of popular multiple-instance learning we propose to establish a random ensemble of sets of bags, i.e., superbags. Classifier training and label inference are then decoupled by performing them on different superbags. Label inference is performed on samples from separate superbags, and thus avoids label imputation on training samples in the same superbag. Experimental evaluations on standard datasets show consistent improvement over widely used approaches for multiple-instance learning.

1 Introduction

Over the last decade machine learning has come a long way in proposing novel training scenarios that are suited for new applications where the classical supervised learning is not feasible. Of particular importance is the framework of multiple-instance learning (MIL), both from the theoretical and practical point of view. In contrast to standard supervised learning where all training samples are provided with labels, pattern labels in MIL are unknown and need to be discovered during training. MIL accommodates the training patterns in bags, and the training labels are only provided at the bag level.

The MIL framework has been successfully applied in many practical problems because it provides a powerful mechanism to deal with label ambiguities that are common in weakly annotated datasets. After initial application of MIL to drug activity prediction [1], the concept of MIL has quickly spread to many other disciplines such as text-categorization [2] and computer vision. Many authors used MIL for image retrieval [3, 4], image categorization [5] or object detection [6]. Object tracking has also greatly benefited from the MIL setup [7, 8], which seamlessly picks among many ambiguous patches one that best represents the object and uses it for the update of the appearance model.

In order to learn a classifier in the presence of ambiguous labels, MIL proceeds by iterating between learning a classifier and finding the missing labels in an alternating fashion. However, the two steps are performed on the same training

samples, which renders classifier learning and label inference prone to overfitting and increases the variance of the estimates for the unknown labels.

So how can we resolve these issues and increase the robustness of MIL? First, we should avoid predicting the labels for the same instances that the classifier is trained upon. Second, we can decrease the uncertainty of label inference by averaging over multiple predictions from separate classifiers. These goals can be addressed by establishing a random ensemble of sets of bags, that we call *superbags*. Training a classifier on a superbag, predicting labels for elements from other superbags, and averaging all these predictions decouples classifier training and label inference and, thus, increases the robustness of MIL. We believe that the proposed approach is of general interest to many methods that employ the idea of MIL. The approach is easy to integrate to existing MIL methods, and its results show a consistent gain over baseline methods which train the classifier and predict the labels on the same patterns.

Training a classifier with missing labels also appears in semi-supervised learning, where it is addressed by co-training [9]. In co-training, two classifiers are trained on the same labeled set of points, but with different features. The classifiers then predict labels of a large unlabeled set of points. Patterns that are confidently labeled by either of the classifiers are appended to the training set of both classifiers. So the two classifiers are always updated with the same set of training points. Confident label predictions of one classifier are used during co-training to resolve ambiguities about unlabeled patterns of the other classifier. Different from the concept of co-training, our superbag approach trains the classifiers on the *same* features, but using *different* data points. Superbag classifiers are all trained on the same features, since in many applications finding new independent features is not feasible. Usually, it is not possible to change the feature representation of the given data. Training classifiers on the same data, as performed by co-training, makes the classifiers dependent on each other and their training less robust. Our superbag approach trains the classifiers on different data points which lowers the variance of inferred labels and increases the robustness of the classifiers. Superbag classifiers are trained only from weakly labeled points, whereas in co-training the classifiers are trained initially on the fully labeled set of points. Roth et al. [10] recently applied the MIL algorithm with co-training to the problem of multiple-camera object detection. They apply MIL algorithm alongside with co-training to detect image regions that most likely represent an object.

Classifying data patterns with multiple classifiers is also part of the bagging method [11]. Bagging makes several training sets by sampling them with replacement from the original set of points. These sets are then used for training the ensemble of classifiers, which later jointly classify new test points. In bagging, test labels are obtained by averaging the predictions from all classifiers in the ensemble. However, bagging can be applied only in the supervised setting, where the labels of training instances are all provided. In contrast, our superbag ensemble is trained on weakly labeled patterns, where finding the missing labels and training the classifiers are performed simultaneously. Besides, in our

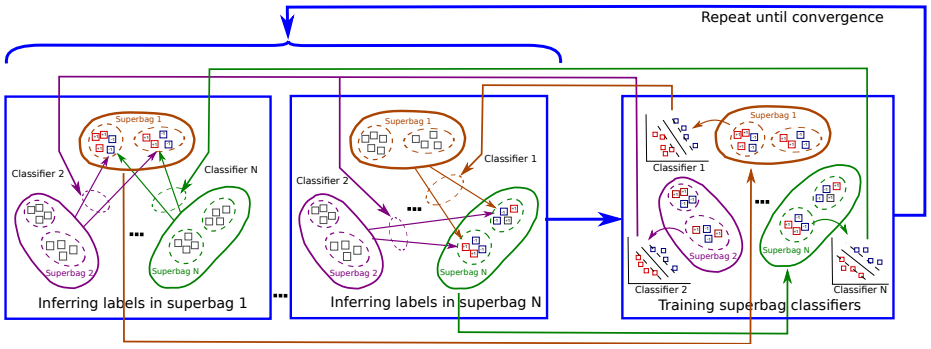


Fig. 1. Sketch of the processing pipeline for decoupling classifier training and label inference by performing them on different superbags.

superbag approach the ensemble of classifiers is used only during training to robustly resolve the missing labels, whereas in bagging the ensemble of classifiers is used only for testing. After inferring the missing labels of training instances, our superbag method trains the final classifier that predicts labels of test samples.

In Sect. 2 we review MIL before presenting our contribution in Sect. 3 that extends MIL algorithms with an ensemble of superbags. We continue in Sect. 4 to present instantiations for the various MIL methods extended by the concept of superbags. Sect. 5 analyses how ensemble of superbags reduces the uncertainty of label prediction in MIL and we perform an experimental evaluation on standard datasets in Sect. 5.2 before concluding in Sect. 6.

2 Multiple-Instance Learning

In a standard supervised setting, one is given a training set that consists of labeled patterns $(x_i, y_i) \in \mathbb{R}^d \times \mathcal{Y}$, and the goal is to learn a classifier $f : \mathbb{R}^d \to \mathcal{Y}$, i.e. a function that maps patterns to labels. Multiple-instance learning is dealing with pattern labels in a weakly supervised way. Labels are provided only for sets of instances that are called *bags*. Each bag B_I is specified by an index set $I \subseteq \{1, 2, ..., n\}$, i.e. $B_I = \{x_i : i \in I\}$. Multiple-instance learning is defined on the finite set of bags $\{B_I\}_{I \in \tilde{I}}$, where family of index sets $\tilde{I} \subseteq 2^{\{1,2,...,n\}}$ is a subset of the power set of set $\{1, ..., n\}$. There are in total m bags in the dataset, i.e. $|\tilde{I}| = m$. A label Y_I is associated with each bag B_I, and they are defined in the following way. Patterns in the negative bag *all* belong to the negative class, $Y_I = -1 \Rightarrow \forall i \in I : y_i = -1$. On the other hand, a bag with positive label requires that *at least* one of the patterns of the bag belongs to the positive class, $Y_I = 1 \Rightarrow \exists i \in I : y_i = 1$. The goal of MIL is to simultaneously find the missing pattern labels y_i and the instance classifier f. Finding the unknown pattern labels is also known as *imputation* step. Imputed labels have to satisfy MIL constraints that express the relation between bag labels Y_I and corresponding instance labels y_i, i.e. $Y_I = \max_{i \in I} y_i$.

Due to label ambiguity in positive bags, the learning problem is naturally defined as a mixed integer problem. In case of a linear discriminant function $f(x) = w^\top x$, one is looking for the weight vector[1] $w \in \mathbb{R}^d$, together with unobserved integer variables y_i that are missing instance labels in positive bags. Inferring the missing labels is a hard combinatorial problem that is usually solved with alternating optimization. It consists of the two steps: (i) *inferring the labels* - given the discriminant function, find the integer variables y_i that correspond to the unknown pattern labels in positive bags, (ii) *classifier learning* - given the inferred instance labels from the previous step, find the optimal parameter w of the discriminant function. These two steps are performed on the same patterns simultaneously, which means that the same instances are used for both training the discriminant function and imputing the missing labels. So for inferring the missing labels the classifier is evaluated on the same samples it was trained upon, which makes MIL susceptible to overfitting.

3 Multiple-Instance learning with Superbags

In the widely employed multiple-instance learning framework discriminative learning and label inference are always performed on the same training samples, which makes the learning algorithm susceptible to overfitting, and, as a consequence, increases the variance of instance label prediction. Our approach improves the robustness of multiple-instance learning by decoupling the training of discriminant function and the inference of pattern labels. Key steps of MIL algorithm are performed on separate bags with their results combined at the end in the final classifier. We show that by using separate sets of bags for classifier training and label inference, the final classifier can obtain a lower error in predicting unknown instance labels. With respect to the well-known bias-variance decomposition of the mean squared error, integration of superbags into MIL decreases the variance of label predictions without increasing the bias. Superbags slightly prolong the training time for MIL, but the time for testing stays the same. In the sequel we explain how label inference and classifier training steps become separated when the superbags are integrated into MIL.

As stated above, the goal of multiple-instance learning is to simultaneously find the optimal pattern labeling y_i and the optimal discriminant function f. If we knew the correct classifier $f(x) = w^\top x$, unknown labels could be found by assigning all patterns to positive or negative class based on the sign of the discriminant function, $\hat{y}_i = \operatorname{sgn} f(x_i)$. However, the discriminant function is unknown and it needs to be learned from the training samples whose labels are unknown. In the course of multiple-instance learning, the unknown instance labels y_i, and the classifier hyperplane w are simultaneously updated. Labels $y_i \in \{-1, +1\}$ should ideally match the labels assigned to instances x_i by the classifier w. Optimal values of labels y_i are obtained by minimizing the loss function $\ell(x_i, y_i, w)$ that measures the discrepancy between the labels and classifier

[1] Offset w_0 of the discriminant function is included in the weight vector w.

predictions. Estimated labels also have to satisfy the MIL constraints defined on both positive and negative bags,

$$\sum_{i \in I} \frac{y_i + 1}{2} \geq 1, \text{ if } Y_I = 1, \tag{1}$$

$$\wedge \, \forall i \in I : y_i = -1, \text{ if } Y_I = -1. \tag{2}$$

In the baseline MIL setup, the classifier w is chosen from a set of well-behaved functions that is defined by a regularizer $\Omega(w)$. A good classifier needs to produce accurate predictions on the training points. The discrepancy between instance labels and predictions made by classifier is measured by the empirical loss function $R_{emp}(w)$. This function is the sum of loss functions at all training points, $R_{emp}(w) := \sum_{i=1}^{n} \ell(x_i, y_i, w)$. The classifier w is then found by minimizing the weighted sum of the regularizer $\Omega(w)$ and the empirical loss function $R_{emp}(w)$,

$$\hat{w} = \operatorname*{argmin}_{w} R_{emp}(w) + \lambda \Omega(w). \tag{3}$$

As both the pattern labels and the classifier hyperplane are updated on the same data, the two processes become strongly entangled, which increases the error of label predictions. The mean squared error (MSE) of label prediction in point x_i is the average squared difference between the prediction $\hat{y}_i$ and the ground truth label y_i, i.e. $MSE = \mathbb{E}\{(\hat{y}_i - y_i)^2\}$. MSE can be decomposed into the squared bias $b^2 = (\mathbb{E}\{\hat{y}_i\} - y_i)^2$ and the variance $\sigma^2 = \mathbb{E}\{(\hat{y}_i - \mathbb{E}\{\hat{y}_i\})^2\}$. The bias is a measure of systematic error in label prediction that gets larger as the model becomes less flexible. On the other hand, variance quantify how much the predicted value varies around the average prediction. Overly flexible models have high variance, because they easily overfit to the particular dataset. We measure MSE and variance of our superbag based MIL classifier in Sect. 5.1 by computing the average error of label predictions on a fixed set of test points.

A natural question is whether we can avoid overfitting by decreasing the variance of label predictions that will then lead to a more robust model that better generalizes to new samples. Our solution (illustrated in Figure 1) is based on two observations: (i) overfitting can be avoided if the labels are predicted by classifiers that are not trained on the same training samples, and, (ii) the variance of label predictions can be decreased by averaging over predictions of multiple classifiers. We show that both of these requirements can be satisfied if we randomly separate all training bags into multiple sets of bags - *superbags*. We train a classifier on each superbag and use it to predict labels of patterns that lie in other superbags. This way, we decouple the training of the classifier and the inference of missing labels. Moreover, for each unknown label we obtain a number of predictions from different classifiers in the ensemble, that allows us to average all predictions and obtain a lower variance for imputed labels. In contrast to the standard problem of model selection where the variance is traded off against the bias, in our case the bias is not increased, because we use the classifier of the same complexity throughout all experiments. We illustrate this in Sect. 5.1 that analyzes the uncertainty of label prediction in MIL.

As indicated above, superbags are generated by random sampling from the set of all training bags. We create an ensemble of superbags $\{S_{\mathcal{I}}\}_{\mathcal{I}\in\hat{\mathcal{I}}}$, where each superbag is defined as a set of bags, $S_{\mathcal{I}} = \{B_I : I \in \mathcal{I}\}$. The family of superbag indexes $\hat{\mathcal{I}} \subseteq 2^{\tilde{I}}$ is finite, and it is a subset of the power set of the index family $\tilde{I}$. Let k denote the total number of superbags, $|\hat{\mathcal{I}}| = k$. Superbags are created by sampling bags with replacement, otherwise classifiers would be trained on too few training bags in superbags. Consequently, superbags might overlap, i.e. superbag index sets $\mathcal{I} \in \hat{\mathcal{I}}$ are in general not disjoint. Size of a superbag is set to the fraction r of the total number of bags in the dataset, i.e. $\forall \mathcal{I} \in \hat{\mathcal{I}} : |S_{\mathcal{I}}| = r \cdot |\tilde{I}|$. A separate classifier $f_{\mathcal{I}}$ is trained on each superbag $S_{\mathcal{I}}$, i.e. a hyperplane $w_{\mathcal{I}}$ is learned only from patterns that belong to superbag $S_{\mathcal{I}}$. The hyperplane $w_{\mathcal{I}}$ is selected from a set of well-behaved functions that are determined by the regularizer $\Omega(w_{\mathcal{I}})$, and it also has to fit well to the training data in the superbag $S_{\mathcal{I}}$. This is quantified by the superbag empirical loss function,

$$R_{emp}(w_{\mathcal{I}}) := \sum_{I:I\in\mathcal{I}} \sum_{i:i\in I} \ell(x_i, y_i, w_{\mathcal{I}}). \tag{4}$$

New classifier is found by minimizing the weighted sum of the regularizer and the superbag empirical loss function,

$$\hat{w}_{\mathcal{I}} := \operatorname*{argmin}_{w_{\mathcal{I}}} R_{emp}(w_{\mathcal{I}}) + \lambda\Omega(w_{\mathcal{I}}). \tag{5}$$

Our goal is to decouple the inference of missing instance labels y_i from the training of the ensemble of classifiers $w_{\mathcal{I}}$. Therefore, ideally label y_i of pattern x_i is estimated only by classifiers that are trained on superbags which do not contain the point x_i. However, in order to provide numerical stability of the iterative procedure, we also include predictions made by classifiers that are trained on x_i, but these predictions are then given a smaller weight β, whose value is determined in cross-validation. Consequently, the labels y_i are inferred by the following optimization,

$$\hat{y}_i = \operatorname*{argmin}_{y_i} \sum_{\mathcal{I}:i\notin\mathcal{I}} \ell(x_i, y_i, w_{\mathcal{I}}) + \beta \sum_{\mathcal{I}:i\in\mathcal{I}} \ell(x_i, y_i, w_{\mathcal{I}}), \tag{6}$$

subject to the general MIL constraints defined earlier in Eq. 1 and 2. Note that standard MIL is a special case of superbag MIL when $k = 1$ and $r = 100\%$, i.e. when there is only one superbag which contains all the training bags. In that case, only the second term remains in Eq. 6.

4 Integrating Superbags into Common MIL Approaches

In this section we show how the concept of superbags can be easily integrated into some of the most popular instance-level classifiers for MIL that are based upon the standard soft-margin SVM formulation [2, 12]. In particular, we choose the methods AL-SVM, AW-SVM and ALP-SVM proposed by Gehler and Chapelle

[12], because they generalize the widely employed mi-SVM and MI-SVM methods [2] used in many different applications. In Sect. 4.1 - 4.3 we show how to integrate the concept of superbags into the methods of AL-SVM, AW-SVM and ALP-SVM, respectively.

4.1 AL-SVM with Superbags

AL-SVM [12] improves the very popular mi-SVM method [2] using the deterministic annealing (DA). DA is a general method for solving non-convex discrete optimization problems, such as $\hat{y} = \operatorname{argmin}_{y \in \{-1,1\}^n} J(y)$, that treats the unknown discrete variable y as binary *random* variable with unknown discrete distribution p. Gehler and Chapelle [12] assume that the probability distribution of missing instance labels y in the mi-SVM method can be factorized into a product of marginal probabilities $p_i = P(y_i = +1)$. In order to incorporate the MIL constraints, probabilities of all patterns in negative bags are set to zero, $p_i = 0$. For each positive bag B_I, there is at least one positive pattern, thus a constraint $\sum_{i:i \in I} p_i \geq 1$ is imposed.

The concept of superbags can be easily integrated into the AL-SVM algorithm. An ensemble of classifiers $w_{\mathcal{I}}$ is trained on all patterns from a superbag $S_{\mathcal{I}}$ by minimizing the regularized empirical loss,

$$\hat{w}_{\mathcal{I}} = \operatorname*{argmin}_{w_{\mathcal{I}}} R_{emp}(w_{\mathcal{I}}) + \lambda \Omega(w_{\mathcal{I}}), \tag{7}$$

with the empirical loss function

$$R_{emp}(w_{\mathcal{I}}) := \sum_{I: I \in \mathcal{I}} \sum_{i: i \in I} \big(p_i \ell(x_i, y_i = +1, w_{\mathcal{I}}) + (1 - p_i)\ell(x_i, y_i = -1, w_{\mathcal{I}})\big). \tag{8}$$

In contrast to Eq. 4 that operates on deterministic label assignments y_i, Eq. 8 contains the expectation of the empirical loss, because missing labels are now treated as binary random variables. In case of the quadratic regularizer $\Omega(w_{\mathcal{I}}) = \frac{1}{2}\|w_{\mathcal{I}}\|^2$ and the standard hinge-loss function $\ell(x_i, y_i, w_{\mathcal{I}}) = \max\big(0, 1 - y_i \cdot (w_{\mathcal{I}}^\top x_i)\big)$, the optimization problem reduces to a quadratic program (QP), that can be solved using standard solvers.

In the label inference step of our superbag-enhanced AL-SVM, label probabilities p_i are updated by minimizing the weighted sum of empirical losses incurred by superbag classifiers,

$$\begin{aligned} \hat{p}_i = \operatorname*{argmin}_{p_i} & \sum_{\mathcal{I}: i \notin \mathcal{I}} \big(p_i \ell(x_i, y_i = +1, w_{\mathcal{I}}) + (1 - p_i)\ell(x_i, y_i = -1, w_{\mathcal{I}})\big) + \\ \beta & \sum_{\mathcal{I}: i \in \mathcal{I}} \big(p_i \ell(x_i, y_i = +1, w_{\mathcal{I}}) + (1 - p_i)\ell(x_i, y_i = -1, w_{\mathcal{I}})\big) - T \cdot H(p_i). \end{aligned} \tag{9}$$

The entropy function H is used by DA to regularize the label inference. For a standard hinge-loss function, the solution can be obtained in a closed form, by transforming the MIL constraints into the Lagrangian dual and applying the Karush-Kuhn-Tucker theorem.

4.2 AW-SVM with Superbags

Superbags can be also successfully integrated into the AW-SVM method [12]. The AW-SVM algorithm uses DA to find solution to MI-SVM [2], which tries not infer all missing labels, but only those that are *witnesses* of the unknown target concept in positive bags. The AW-SVM method finds the distribution over patterns in positive bag, i.e. it calculates the probability p_i that an instance x_i is a witness in its bag. The sum of probabilities in each bag has to be one, $\sum_{i:i\in I} p_i = 1$. In negative bags, all patterns are treated as negative witnesses, since their labels are known to be negative.

AW-SVM can integrate easily the concept of superbags. A classifier $w_{\mathcal{I}}$ is trained only on samples from the superbag $S_{\mathcal{I}}$, which is achieved by minimizing the regularized expected empirical loss function,

$$\hat{w}_{\mathcal{I}} = \underset{w_{\mathcal{I}}}{\operatorname{argmin}}\, R_{emp}(w_{\mathcal{I}}) + \lambda\Omega(w_{\mathcal{I}}), \tag{10}$$

with the expected empirical loss function

$$R_{emp}(w_{\mathcal{I}}) := \sum_{I:I\in\mathcal{I}} \sum_{i:i\in I} p_i \ell(x_i, y_i = Y_I, w_{\mathcal{I}}). \tag{11}$$

For the quadratic regularizer and the hinge-loss function, the optimization in Eq. 11 is a standard QP that can be solved by standard solvers.

The inference step in our superbag extension of AW-SVM finds the probability p_i that pattern x_i is the witness in its bag. This amounts to minimizing the expected empirical loss given the general MIL constraints as before,

$$\hat{p}_i = \underset{p_i}{\operatorname{argmin}} \sum_{\mathcal{I}:i\notin\mathcal{I}} p_i \ell(x_i, y_i = Y_I, w_{\mathcal{I}}) + \beta \sum_{\mathcal{I}:i\in\mathcal{I}} p_i \ell(x_i, y_i = Y_I, w_{\mathcal{I}}) - T \cdot H(p_i) \tag{12}$$

The solution of the inference problem for the standard hinge-loss function can be easily obtained starting from the Lagrangian dual.

4.3 ALP-SVM with Superbags

The mi-SVM algorithm is initialized by labeling all patterns of a positive bag as positive. As a result, the mi-SVM method overestimates the number of positive patterns in a positive bag and the optimization gets easily trapped in a local optima. However, DA does not label too many patterns positively, but it suffers from relatively mild MIL constraints that ask for at least one positive pattern in a positive bag. Typically, only few patterns in a positive bags are labeled as positive. In order to alleviate the problem, Gehler and Chapelle [12] proposed to add a term to the MIL objective, that plays a similar role as the balancing constraint in the semi-supervised learning. Balancing term penalizes the large deviation from the expected number of positively labeled points in a positive

bag, that is assumed to be the α_I fraction of the total number of patterns in the bag I.

The balancing term $\sum_I (\sum_{i \in I} p_i - \alpha_I |I|)^2$ does not depend on the parameters $w_{\mathcal{I}}$ of the ensemble of superbag classifiers. Therefore, the classifiers can be trained in the same way as in Eq. 7. However, the label inference step is changed by the addition of the balancing term, and it has now the following form,

$$\begin{aligned}\hat{p}_i = \underset{p_i}{\operatorname{argmin}} & \sum_{\mathcal{I}: i \notin \mathcal{I}} \big(p_i \ell(x_i, y_i = +1, w_{\mathcal{I}}) + (1 - p_i)\ell(x_i, y_i = -1, w_{\mathcal{I}})\big) + \\ & \beta \sum_{\mathcal{I}: i \in \mathcal{I}} \big(p_i \ell(x_i, y_i = +1, w_{\mathcal{I}}) + (1 - p_i)\ell(x_i, y_i = -1, w_{\mathcal{I}})\big) \\ & + \gamma \sum_I \big(\sum_{i \in I} p_i - \alpha_I |I|\big)^2 - T \cdot H(p_i). \end{aligned} \tag{13}$$

5 Experimental Evaluation

In the experimental section, we evaluate how integrating superbags improves MIL by decreasing the variance of label predictions and avoiding overfitting. In Sect. 5.1 we first analyze the uncertainty of label predictions in a synthetic experiment, and in Sect. 5.2 standard MIL benchmark datasets are used to measure the performance gains after integrating the ensemble of superbags into some popular MIL methods that are used as baselines. Finally, in Sect. 5.3 we show how performance of the MIL based image re-ranking system can be improved if the concept of superbags is applied to it.

5.1 Analyzing the Uncertainty of Label Predictions in MIL

A synthetic experiment is created in order to analyze the uncertainty of label predictions in MIL. The dataset consists of $m = 100$ bags, where each bag has five points sampled from a unit square in the plane. The diagonal of the square separates the positive and the negative class. 30 bags are sampled strictly from the negative class (negative bags), while other bags are sampled from both classes. Baseline MIL algorithm in this experiment is mi-SVM, which infers the missing labels for all training instances. The mi-SVM algorithm can be obtained from AL-SVM by setting the temperature T to zero. We integrate the idea of superbags into the baseline mi-SVM method. Ten superbags are created by random sampling, with their size being changed from $r = 10\%$ to $r = 100\%$ in an increment of 10%. We note that superbags of the maximal size $r = 100\%$ both train the classifiers and find the missing labels on all the training data at once, which corresponds to the baseline mi-SVM method. We use linear SVM classifier and fix the hyperparameters to $C = 20$ and $\beta = 0.5$.

The averaged results over five hundred independent runs are given in Fig. 2. It shows the uncertainty of label predictions measured by the mean squared error (MSE), and its two components, variance and squared bias, that were discussed in Sect. 3. All three quantities are normalized with respect to the mean

Table 1. The classification error (%) on five different MIL benchmark datasets. We compare the performance of methods AL-SVM, AW-SVM and ALP-SVM when they use superbags to their baseline versions that are without superbags. In all cases, a consistent improvement in performance is achieved. The standard deviation for baseline is around 3.5% and 3.1% for superbags.

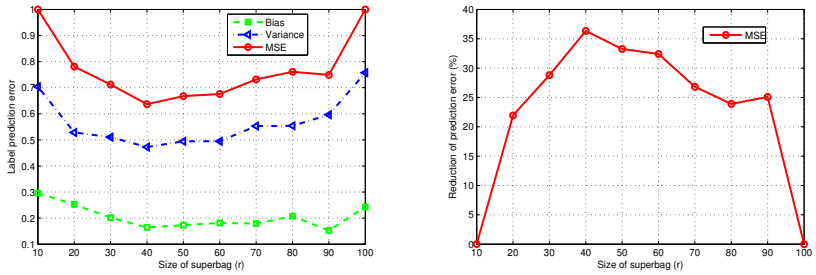
Dataset	EMDD [3]	mi-SVM [2]	MI-SVM [2]	MILIS [13]	AW - SVM			AL - SVM			ALP - SVM		
					B/L	Superbags	Gain	B/L	Superbags	Gain	B/L	Superbags	Gain
Musk1	15.2	12.6	22.1	11.4	14.3	14.2	+0.1	13.3	13.1	+0.2	13.7	12.1	+1.6
Musk2	15.1	16.4	15.7	8.9	16.2	13.8	+2.4	17.4	17.4	0	13.8	13.4	+0.4
Tiger	27.9	21.6	16	N/A	17	14.5	+2.5	21.5	16.5	+5	14	14	0
Elephant	21.7	17.8	18.6	N/A	18	17.5	+0.5	20.5	17.5	+3	16.5	16	+0.5
Fox	43.9	41.8	42.2	N/A	36.5	33	+3.5	36.5	33	+3.5	34	31	+3

squared error of the baseline mi-SVM method. The baseline method is obtained for $r = 100\%$. The variance quantifies how much the model is susceptible to overfitting. We see that by decreasing the size of superbags r, the variance of label predictions decreases, because overfitting becomes less prominent. This is a direct consequence of the decoupling of label inference and classifer training when the size of superbags is decreased. In this experiment, the changes of bias are smaller than the changes of variance. This is because the complexity of the original SVM classifier is not changed. The percent of reduction of MSE with respect to the baseline performance is shown on the right of Fig. 2. We see that the prediction error is always lower than in the baseline, and the maximal reduction is achieved for the superbag size of $r = 40\%$. The error rate in that case drops by approximately 35%.

5.2 Evaluation on Benchmark Datasets

In this section, we evaluate the proposed concept of superbags on the benchmark datasets for MIL. The ensemble of superbags is integrated with the popular AL-SVM, AW-SVM and ALP-SVM algorithms proposed by Gehler and Chapelle [12], and the results are compared to their baseline versions that do not use superbags. The benefit of these classifiers is that they can resolve ambiguous instance labels jointly with bag-level classification. Moreover, mi-SVM-like classifiers are quite popular in the MIL literature, and have found a wide use in computer vision. We compare our results also to the bag-level classifiers [13], which predict only bag labels using rich bag-level features, but do not infer missing instance labels, and are, thus, not applicable for many applications.

We use the well-established benchmark sets, MUSK [1] (Musk1 and Musk2) and COREL [2] (Tiger, Elephant and Fox), for the comparison of MIL algorithms. We follow the experimental setup of Gehler and Chapelle [12], and use SVM classifier with RBF kernel whose bandwidth σ and parameter C are selected from the sets $\sigma \in \{\ 0.5\sigma_0, \sigma_0, 2\sigma_0\}$ and $C \in \{1, 10\}$ by tenfold cross-validation. The value of σ_0 is computed as the median of pairwise distances of all training samples. The balance term α in ALP-SVM is selected by cross-validation from the set $\alpha \in \{0.1, 0.2, ..., 1.0\}$. AL-SVM and AW-SVM are performed without annealing, because, as noted by Gehler and Chapelle [12], annealing in their case



(a) Label prediction error (b) Reduction of the prediction error

Fig. 2. Analysis of the label prediction errors in the synthetic experiment. Fig. (a) shows the mean squared error (MSE) and its two components, variance and squared bias, when the size of superbags r is changed. All quantities are normalized with respect to MSE of the baseline mi-SVM method ($r = 100\%$). Fig. (b) shows in percents the reduction of the mean squared error with respect to the baseline mi-SVM method. The largest decrease of the prediction error (35%) is obtained for superbags of the size $r = 40\%$.

does not translate into a smaller test error. Consequently, annealing sequence is only applied to ALP-SVM, where it starts from the temperature $T = 10C$ and is decreased at the rate of 2/3 per round.

We integrate the idea of superbags into three standard MIL algorithms, AL-SVM, AW-SVM and ALP-SVM, and compare the results with their baseline versions that do not use superbags. We sample randomly $k = 10$ superbags. We select superbag size r from the set $r \in \{20\%, 50\%, 80\%\}$ and $\beta \in \{0.5, 1, 2\}$ by cross-validation. Superbag size r is the fraction of the total number of bags in the dataset. The best performance is typically obtained for $\beta = 0.5$, i.e. instance labels are strongly predicted by classifiers trained on different superbags.

The results of testing on MIL benchmark datasets are given in the Tab. 1. Classification error of superbag concept integrated into AL-SVM, AW-SVM and ALP-SVM methods is compared with their baseline version that do not use superbags. It is evident that the introduction of superbags to the standard MIL methods consistently improves the their baseline versions. The AL-SVM method achieves the gain of +5% on the Tiger dataset when superbags are used. The AW-SVM method show the gain of up to +3.5% on the Fox dataset. Lastly, the the ALP-SVM approach shows an improvement of +3% on the Fox dataset.

Tab. 1 also provides the results of other MIL methods, namely EMDD [3], MILIS[13], MI-SVM [2] and mi-SVM [2]. Our integration of the ensemble of superbags into ALP-SVM achieves the best score on all three COREL datasets, Tiger, Elephant and Fox. The results on the Tiger dataset are equal to the baseline ALP-SVM method. The superbags also significantly improve the performance of baseline methods on the MUSK datasets. On the Musk1 dataset, the performance of ALP-SVM after integration with superbags is *on par* ($< 1\%$ difference) with MILIS [13], the state-of-the-art method for the MUSK bench-

mark. Good performance of MILIS on Musk2 dataset is due to the rich bag-level features that MILIS as a bag-level classifier uses. As a consequence, bag-level classifiers are however limited in that they do not infer missing instance labels. By integrating superbags into the instance-level classifiers, their bag-level classification performance approaches the performance of the state-of-the-art bag-level classifiers, which are inferior in terms of the labels they can infer.

5.3 Image Re-Ranking Using Superbag MIL

As our last experiment, we apply our superbag enhanced MIL framework to the problem of web image re-ranking. Recently, several groups proposed MIL as a ranking framework [14, 4], that is particularly suitable for the re-ranking of web image search results.

The Google data set was proposed originally by [15] to enhance the learning of object categories from web image search results. The dataset consists of about 4000 images divided into 7 categories that have on average 600 images. Since images are taken from a text based search, only around 30% of images are with a "good" view of the desired class, 20% are "ok" views, whereas the remaining 50% of images are considered as "junk" images, as they are completely unrelated to the category. In order to apply MIL, the images need to be grouped into multiple bags beforehand. Positive bags are obtained by randomly sampling images that are returned as a search result for given category. If the group is large enough, it can be assumed that at least one image in a bag will be positive. Negative images are obtained by sampling only images from other categories. The seven categories used in the Google dataset are airplane, car (rear), face, leopard, motorbike, guitar and wrist watch.

In order to build a feature representation for images in the Google dataset, we densely sample features around edges at multiple scales. Extracted features are represented by the SIFT descriptor. Our method for feature sampling is simpler than that described in [14, 15], where four interest point detectors are used, i.e. Kadir&Brady operator, Harris-Hessian detector, difference of Gaussians and Edge-Laplace detector. SIFT descriptors are quantized into a codebook of 500 visual words. Each image is represented as a bag of words (BoW), i.e. a histogram of visual words in that image. We use mi-SVM as a baseline for re-ranking of the Google images and compare it to our superbag approach that is integrated with the mi-SVM algorithm. In both cases an SVM with RBF kernel is employed and the kernel bandwidth is set to $4/A$, where A is the mean squared distance between images. All bags are of the same size of 15 images. We use 5 superbags where each superbag consists of 40 bags. Following [14, 4] the per-category precision at 15% recall is measured for performance evaluation. Both "good" and "ok" images are treated as positive samples in the experimental evaluation.

The results of the per-category and mean precision at 15% recall are provided in Tab. 2. Both, baseline mi-SVM and superbag enhanced mi-SVM consistently improve the results of Google's original web image search over all categories. We also see that superbag enhanced mi-SVM achieves significant improvement over the baseline mi-SVM on four out of seven categories. For airplane category this

Table 2. Per-category precision and the mean precision (%) at 15% recall over 7 categories of the Google dataset. The abbreviations are for the category names: Airplane (A), Cars-rear (C), Face (F), Guitar (G), Leopard (L), Motorbike (M) and Wrist-watch (W).

	A	C	F	G	L	M	W	Mean
Google [15]	70.00	69.49	43.82	56.58	66.07	72.53	88.89	66.77
mi-SVM [2]	50.72	64.06	86.67	84.31	64.91	83.54	93.02	75.32
Superbags	76.09	71.93	82.98	82.69	78.72	80.49	97.56	81.50
WsMIL [4]	100	81	57	52	66	79	95	75.71
Schroff [16]	58.5	N/A	N/A	70.0	49.6	74.8	98.1	70.20
PMIL-CPB [14]	100	75.34	89.91	82.74	86.15	76.63	95.72	86.64

improvement is highest and equals approximately 25%. The gain of superbag enhanced mi-SVM in the mean precision over the baseline mi-SVM is 6.2%. The rest of the table shows the results of other state-of-the-art methods, i.e. WsMIL [4], Schroff et al. [16] and PMIL-CPB [14]. The superbag enhanced mi-SVM has the second best result of all compared methods in terms of mean precision, and the gain over Schroff et al.'s method and WsMIL is 11.3% and 5.8%, respectively. Only PMIL-CPB scores better by 5.1% than our superbag based approach. This is a consequence of a stronger constraint for positive bags, which requires that at least a portion of instances in a positive bag is positive, whereas we use a standard MIL constraint with at least one positive instance per positive bag. Besides, the concept of superbags can be also integrated with the PMIL-CPB method to increase its performance.

6 Conclusion

In this paper we have addressed a fundamental issue of widely used multiple-instance learning. In the underlying optimization algorithm, classifier learning and inference of missing labels are iteratively performed on the same training samples. This leads to overfitting and increases the variance of the label estimates. We have tackled these issues by introducing superbags, which effectively decouple both processes. Experiments on standard datasets have shown that this method consistently improves several widely used approaches to multiple-instance learning when being integrated into the optimization routine.

Acknowledgement. This work has been supported by the German Research Foundation (DFG) within the program "Spatio-/Temporal Graphical Models and Applications in Image Analysis", grant GRK 1653, and by the Excellence Initiative of the German Federal Government, DFG project number ZUK 49/1.

References

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

2. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems 15, MIT Press (2003) 561–568
3. Zhang, Q., Goldman, S.A.: Em-dd: An improved multiple-instance learning technique. In: In Advances in Neural Information Processing Systems, MIT Press (2001) 1073–1080
4. Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In: CVPR. (2008)
5. Chen, Y., Bi, J., Wang, J.Z.: Miles: Multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Mach. Intell. **28** (2006) 1931–1947
6. Monroy, A., Ommer, B.: Beyond bounding-boxes: Learning object shape by model-driven grouping. In: ECCV (3). (2012) 580–593
7. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2011)
8. Leistner, C., Saffari, A., Bischof, H.: Miforests: multiple-instance learning with randomized trees. In: Proceedings of the 11th European conference on Computer vision: Part VI. ECCV'10, Berlin, Heidelberg, Springer-Verlag (2010) 29–42
9. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. COLT 98, New York, NY, USA, ACM (1998) 92–100
10. Roth, P., Leistner, C., Berger, A., Bischof, H.: Multiple instance learning from multiple cameras. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. (2010) 17 –24
11. Breiman, L.: Bagging predictors. Mach. Learn. **24** (1996) 123–140
12. Gehler, P.V., Chapelle, O.: Deterministic annealing for multiple-instance learning. Journal of Machine Learning Research - Proceedings Track **2** (2007) 123–130
13. Fu, Z., Robles-Kelly, A., Zhou, J.: Milis: Multiple instance learning with instance selection. IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 958–977
14. Li, W., Duan, L., Xu, D., Tsang, I.W.H.: Text-based image retrieval using progressive multi-instance learning. In: ICCV. (2011) 2049–2055
15. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: Proceedings of the 10th International Conference on Computer Vision, Beijing, China. Volume 2. (2005) 1816–1823
16. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 754–766