Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

SMILe: Shuffled Multiple-Instance Learning

Gary Doran and **Soumya Ray**
Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH 44106, USA
{*gary.doran,sray*}*@case.edu*

Abstract

Resampling techniques such as bagging are often used in supervised learning to produce more accurate classifiers. In this work, we show that multiple-instance learning admits a different form of resampling, which we call "shuffling." In shuffling, we resample *instances* in such a way that the resulting *bags* are likely to be correctly labeled. We show that resampling results in both a reduction of bag label noise and a propagation of additional informative constraints to a multiple-instance classifier. We empirically evaluate shuffling in the context of multiple-instance classification and multiple-instance active learning and show that the approach leads to significant improvements in accuracy.

Introduction

Consider a task such as content-based image retrieval (CBIR) (Maron and Ratan 1998), where we wish to automatically retrieve images from a database based on a small set of images labeled "interesting" or otherwise by a user. A typical assumption we might make in this case is that the images were interesting because they contained *at least one* "interesting" object. For example, if the user was interested in elephants, each image will contain an elephant somewhere in it. On the other hand, "uninteresting" images do not contain elephants. Of course we do not know precisely what the user was interested in, so one approach is to segment an image into its component objects, and assign the label "interesting" or "uninteresting" to the entire *set* of segments. From such data, a retrieval system needs to learn a classifier that can correctly retrieve new interesting images.

The CBIR task above can be naturally cast as a multiple-instance (MI) learning problem (Maron and Ratan 1998). The MI learning setting was originally introduced to represent data in the domain of drug activity prediction (Dietterich, Lathrop, and Lozano-Pérez 1997), but has since been applied to similar problems in which structured objects can be labeled at multiple levels of abstraction, with some rule relating labels at different levels. The MI representation of CBIR data contains labeled *bags* (images), which are sets of labeled *instances* (segments). A bag's label is a logical disjunction of its instances' binary labels; i.e., a bag is positive if any instance in the bag is positive, and negative if no instance is. However, training data contains only bag-level labels, which provide weak label information about instances. The task is to learn a classifier to predict the labels of new bags.

In supervised learning, resampling techniques are often useful in generating accurate classifiers. A good example is the bagging method (Breiman 1996), where an ensemble of classifiers is constructed using bootstrap resampling from the training set. Bagging is known to improve generalization by reducing sample variance. Such techniques have been directly extended to MIL as well by resampling bags (Zhou and Zhang 2003).

In our work, we describe a novel resampling technique for MIL. This technique differs from prior work by showing that it is possible to resample *instances* from positive bags and get new, accurately labeled positive bags. We show that this form of resampling, which we call *shuffling*, also has noise reduction properties similar to "standard" resampling. Furthermore, the sampled bags produced by our approach represent new constraints on instance labels, providing extra information to an MI classifier. This additional information can be particularly useful for settings such as active learning in which initial labeled datasets contain few bags. On real-world datasets, we show that shuffled bags improve classifier performance, increase the learning rate for MI active learning, and significantly outperform bagging MI classifiers.

Algorithm and Analysis

Consider an MI dataset with three positive bags, $\{x_1, x_2\}$, $\{x_3, x_4\}$, and $\{x_5, x_6\}$. Each positive bag must have some positive instance, so let us assume that x_1, x_3, and x_5 are the only "true" positive instances within these positive bags. Let us randomly generate two bags, *each* of which contains three instances sampled without replacement from the set of *all* instances in positive bags $\{x_1, \ldots, x_6\}$. We thus obtain two new bags, for example $\{x_1, x_4, x_6\}$ and $\{x_2, x_4, x_5\}$. Interestingly, we observe that the newly generated bags each contain at least one positive instance, and so they are also positive! Was this an accident? As it turns out, the only way we might have sampled a negative bag is by drawing the instances $\{x_2, x_4, x_6\}$ in some order, with probability only $\left(\frac{3}{6}\right)\left(\frac{2}{5}\right)\left(\frac{1}{4}\right) = 5\%$. Two independently sampled bags such as those above are positive $(95\%)^2 \approx 90\%$ of the time.

Therefore, if we pool together all instances from positive bags, and sample without replacement *enough times* to create new bags, then the generated bags will be (with high probability) positive as well. Note that we did *not* need to know or estimate the individual instance labels during the resampling process.

This seems to be an interesting fact, but what is the value added by these new bags? Let f be the function that assigns "true" or "false" labels to these instances, corresponding to positive and negative. Each positive bag is then labeled by a disjunction over instance labels, and serves as a *constraint* on f: we are looking for an f that satisfies all constraints. Viewing things from this angle, we notice that the new bags can *add new constraints*: in the above example, we knew that one of x_1 or x_2 must be positive; after we add the new bags, we know that one of x_1, x_4 or x_6 must be positive as well. If the classifier is reasonably confident that x_4 and x_6 are negative, this added constraint makes x_1 much more likely to be positive. In general, we can view these added bags as constraints that are (probabilistically) deduced from the initial set. While a rule-based MI algorithm might be able to infer such constraints from the given data, we conjecture (and demonstrate empirically) that certain statistical learning algorithms benefit greatly by having these made explicit, especially when the initial training set is small.

A second value addition that we demonstrate below is that, because we control the parameters of the resampling process, it is able to reduce bag-level label noise in the original data, which improves generalization and reduces overfitting. This benefit is similar to standard bootstrap resampling. As we demonstrate in the experiments, the combination of these two effects leads to very effective improvements in performance, especially when few labeled bags are available.

We note that, like other resampling procedures such as bagging, this approach "wraps around" any base classifier. However, unlike bagging, this is not inherently an ensemble method. It is perfectly feasible in our setting to simply add the new bags as constraints to the data, and run the base MI algorithm on the expanded sample. This is how we apply the technique in our experiments. In some respects, this is an advantage; for example, we do not need to worry about the additional storage costs of an ensemble, and the interpretability of the base classifier is not affected. We call our approach **S**huffled **M**ultiple-**I**nstance **Le**arning (SMILe).

We illustrate SMILe concretely with an example from the CBIR domain. Suppose we have two positive images from the "Coke Can" class, shown in Figure 1. In the MI representation, images are structureless "bags" of segments. Therefore, we can pool segments together and sample without replacement to generate synthetic images, such as those shown in the bottom of Figure 1. By sampling enough segments, the synthetic images are also likely to be positive, as they visibly are in this example.

To formalize SMILe, we define some notation. Suppose we have an MI dataset given by a list of bags B and associated labels Y, with $B_i = \{x_{ij} \in \mathbb{R}^n\}$ and $Y_i \in \{-1, +1\}$. The set of positive bags is denoted by $B_p = \{B_i \mid Y_i = +1\}$, and the set of positive bag instances

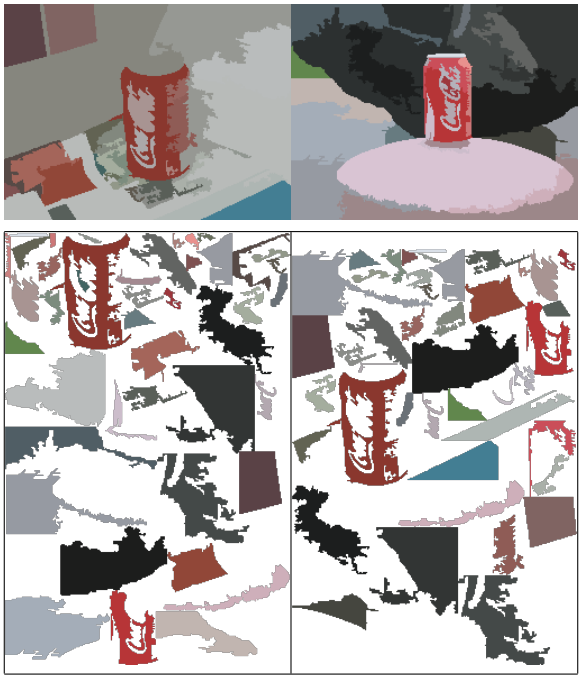

Figure 1: **Top:** Two positive images from the "Coke Can" class. **Bottom:** Two bags generated by sampling instances from the two images above.

by $X_p = \bigcup B_p$. The positive bag instances X_p are not the same as the set of positive instances, since some instances in positive bags might be negative. We can generate a set S of additional positive bags, each containing s instances sampled without replacement from the set of positive bag instances X_p (with replacement between each sampled bag). The set $B \cup S$ is then given to any MI classification algorithm $\mathcal{A}$, and the resulting classifier is used to predict new bag labels.

We first show that the process we have outlined can always generate positive bags with high probability. In the worst case, only one positive instance in each of the positive bags B_p will be positive, for a total of $|B_p|$ true positive instances. The remaining $|X_p| - |B_p|$ positive bag instances will be negative. Therefore, the probability of generating a bag with all negative instances is at most the chance of sampling a negative instance s times, or $((|X_p| - |B_p|) / |X_p|)^s$. In fact, this bound could be improved, since sampling without replacement makes each additional negative instance less likely to be drawn. Nonetheless, we see that the probability of sampling a negative bag, which translates to label noise in S, decreases exponentially with shuffled bag sizes s. Therefore, we can generate positive bags with high probability by adjusting s appropriately:

Proposition 1. *Suppose a set of shuffled bags is generated from an MI dataset without label noise on the original bags. Then if each shuffled bag is of size* $s \geq \frac{1}{C} \log \frac{1}{\epsilon_s}$, *where* $C = \log |X_p| - \log (|X_p| - |B_p|)$, $1 - \epsilon_s$ *of the resulting bags will be positive.*

Proof. To bound the true label noise on S, it is sufficient to bound $\left(\left(|X_p| - |B_p|\right) / |X_p|\right)^s$ by the desired noise level, ϵ_s. Thus, we have:

$$\begin{aligned}
\epsilon_s &\geq \left(\left(|X_p| - |B_p|\right) / |X_p|\right)^s \\
\log \epsilon_s &\geq s \left[\log\left(|X_p| - |B_p|\right) - \log |X_p|\right] \\
\log \tfrac{1}{\epsilon_s} &\leq s \left[\log |X_p| - \log\left(|X_p| - |B_p|\right)\right] \\
s &\geq \frac{\log \frac{1}{\epsilon_s}}{\log |X_p| - \log\left(|X_p| - |B_p|\right)}.
\end{aligned}$$

□

Thus, even in the worst case when there is only one positive instance per positive bag, we can achieve low noise ϵ_s in bags with only $O(\log \frac{1}{\epsilon_s})$ instances. As an example, for the CBIR datasets described in our experiments, positive bags contain approximately 30 instances. For a worst-case noise level of 10%, this requires sampling roughly 70 instances per shuffled bag. Of course, the process of sampling without replacement cannot continue indefinitely, and if s exceeds $|X_p| - |B_p|$, then the resulting sampled bag is guaranteed to be positive, since all instances from *some* positive bag will have been sampled.

We next turn to the question of the value added by the new bags. First, the analysis above suggests that SMILe can reduce *existing* positive bag label noise in an MI dataset:

Proposition 2. *Suppose an MI dataset B contains bags with label error $\epsilon > 0$. By choosing ϵ_s sufficiently small, the noise on the resulting dataset $B \cup S$ will be $\widehat{\epsilon} < \epsilon$, where S is the set of shuffled bags added to the dataset.*

Proof. Suppose bag label error ϵ is decomposed such that $\epsilon_+ > 0$ of positively labeled bags in an MI dataset are negative, and $\epsilon_- > 0$ of the negatively labeled bags are positive. Now, the worst case number of positive instances in positive bags is reduced from $|B_p|$ to $(1 - \epsilon_+)\,|B_p|$. For an abitrary desired ϵ_s error rate on the labels of shuffled bags, we choose s, the size of a shuffled bag, to satisfy:

$$\epsilon_s \geq \left(\left(|X_p| - (1 - \epsilon_+)\,|B_p|\right) / |X_p|\right)^s,$$

then as in the proof above, we must pick shuffled bags of size $s \geq \frac{1}{C} \log \frac{1}{\epsilon_s}$, where $C = \log |X_p| - \log\left(|X_p| - (1 - \epsilon_+)\,|B_p|\right)$. The resulting dataset with shuffled bags labeled positive has $|B_n|$ bags with noise ϵ_-, $|B_p|$ bags with noise ϵ_+, and $|S|$ bags with noise ϵ_s, for a total noise rate of:

$$\widehat{\epsilon} = \frac{\epsilon_-\,|B_n| + \epsilon_+\,|B_p| + \epsilon_s\,|S|}{|B| + |S|}.$$

If we choose ϵ_s small enough so that $\epsilon_s < \min\{\epsilon_-, \epsilon_+\}$, this is sufficient to have $\widehat{\epsilon} < \epsilon$ when shuffled bags are added. □

This shows that like other resampling methods such as bagging, SMILe can reduce variance by reducing noise in an MI dataset.

We next discuss a second benefit of shuffling: its potential to improve performance by introducing additional constraints on an MI classifier. Because the nature of the constraints added by SMILe depends on the particular MI classifier used, we sketch a "template" algorithm below intended to approximate the behavior of an instance-based MI classifier. First, the algorithm arbitrarily assigns labels to instances within bags such that their labels respect the MI condition. That is, all instances in negative bags are labeled negative, and at least one instance in each positive bag is labeled positive. Then, a supervised classifier is trained on the resulting instance-labeled dataset. This is similar to the behavior of algorithms such as mi-SVM (Andrews, Tsochantaridis, and Hofmann 2003), which perform these two steps simultaneously.

Now, assume that instances are separable, and the supervised learning step above returns a consistent classifier. Therefore, all instances in negative bags are labeled correctly by the classifier. However, consider the following types of instances from X_p, the instances in positive bags: (i) the set of instances P selected to be labeled positive by the classifier, (ii) the set of instances in $X_p - P$ that are labeled negative, but whose *true* labels are positive, and (iii) the set of instances in $X_p - P$ that are correctly labeled negative (from the set the negative instances in positive bags). Let δ_- be the false negative rate in positive bags, i.e. the fraction of the instances in $X_p - P$ in category (ii).

Suppose now that we add a single shuffled bag to our dataset and observe the marginal change in the classifier. There are two major cases: (1) some instance in the shuffled bag is in P, or (2) all instances in the shuffled bag are in $X_p - P$. In the first case, the added shuffled bag is already consistent with the classifier (it is a positive bag containing at least one positively labeled instance), so it does not induce a change in the current classifier. The second case has two sub-cases: (2a) the shuffled bag contains at least one *true* positive instance from category (ii) above, or (2b) the shuffled bag only contains negative instances from category (iii) above. In case (2a), we expect the added bag to improve the classifier, since it adds useful, correct constraint information that requires a change in the current classifier. On the other hand, case (2b) is the introduction of a noisy bag, which would also induce a change in the classifier, but based on incorrect information. However the chance of this happening is bounded by the propositions above and *the number of shuffled bags already introduced*, as we discuss below.

How many shuffled bags should be used? For a shuffled bag to have a marginal effect, it must fall under case (2), so each of its s instances must be drawn from $X_p - P$, which occurs with probability at least $\left(\left(|X_p| - |P|\right) / |X_p|\right)^s$. Given that a shuffled bag falls under case (2), the probability that it is harmful as in (2b) is $(1 - \delta_-)^s$, since every instance must be drawn from category (iii) of the instances in X_p. Therefore, as more shuffled bags are added, those of type (2) that have an effect on the classifier will cause more instances to be added to P to satisfy constraints. This decreases the probability that subsequent sampled bags will be of type (2). Furthermore, as bags of type (2a) are sampled, the classifier will use the added constraints to reduce the false negative rate δ_- on positive bag instances. This makes it less likely that subsequent shuffled bags will be of type (2a) and more likely they will be (2b). The exact quantitative nature of the tradeoff between the extra information contained in the constraints of additional shuffled bags and the detrimental ef-

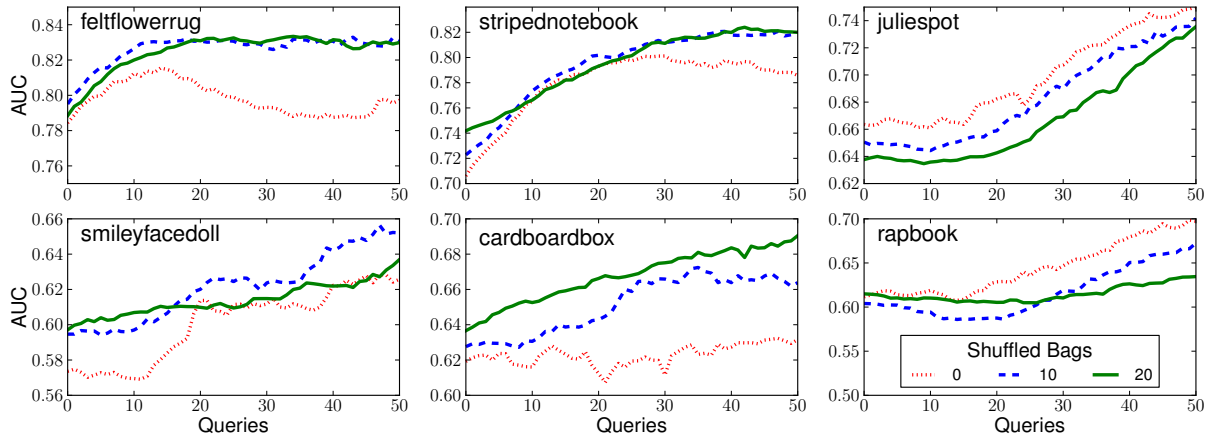

Figure 2: **(MI Active Learning)** Learning curves for the active learning experiments on selected SIVAL datasets showing a variety of behaviors. Each plot shows results using 0, 10, or 20 shuffled bags, starting with 20 initial bags of each class.

fects of introducing incorrectly-labeled shuffled bags to the dataset depends on the particular MI classifier used. Therefore, experimental evaluation and cross-validation should be used to determine an optimal number of shuffled bags.

The discussion above indicates that shuffling will be most beneficial when the false negative rate δ_- is initially high without shuffled bags, and that shuffling will not always add additional constraint information. For example, if δ_- is initially low, then bags corresponding to "new" constraints are likely to be incorrectly labeled. However, even though the constraint-propagation aspect of SMILe will not always work, noise reduction as in Proposition 2 might still be possible. We next show that in practice, SMILe is effective in significantly improving accuracy for MI classification and MI active learning.

Empirical Evaluation

We hypothesize that using SMILe to add shuffled bags to an MI dataset will increase the performance of an MI classifier trained on the augmented dataset. Numerous supervised learning approaches, such as decision trees (Blockeel, Page, and Srinivasan 2005), artificial neural networks (Ramon and Raedt 2000; Zhou and Zhang 2002), Gaussian models (Maron 1998; Zhang and Goldman 2001), logistic regression (Xu and Frank 2004; Ray and Craven 2005), and kernel methods such as support vector machines (SVMs) (Gärtner et al. 2002; Andrews, Tsochantaridis, and Hofmann 2003) have been extended to the MI setting. In this work, we use SVM classifiers for our empirical analysis. Many MI SVM approaches modify the constraints of the standard SVM quadratic program (QP) to take into account the MI relationship between instance and bag labels. Another MI SVM approach, which we use in our empirical analysis, is the normalized set kernel (NSK) (Gärtner et al. 2002). The NSK uses an average of pairwise instance kernel values to compute a kernel between two bags:

$$k_{\text{NSK}}(B_i, B_j) = \frac{1}{|B_i|\,|B_j|} \sum_{x \in B_i} \sum_{x' \in B_j} k_{\text{I}}(x, x'). \qquad (1)$$

The NSK maps entire bags into a feature space, and can be used with a standard SVM QP formulation. The approach works well in practice and is efficient since the bottleneck optimization step is on the order of the number of bags rather than the number of instances.

The NSK uses bag-level information for classification, mapping positive and negative bags into a feature space irrespective of the specific MI assumption of at least one positive instance per positive bag. Thus, shuffled *negative* bags are also used in the experiments since they might provide additional information to the bag-level classifier. Since all negative bag instances are negative, the resulting shuffled bags are also guaranteed to be negative, and no noise is added.

In our experiments, a linear instance kernel k_{I} is used to construct the NSK (see Equation 1) since more powerful kernels tend to overfit small datasets. The SVM regularization trade-off parameter is fixed at 1.0. All experiments are implemented in Python using the `scikit-learn` library (Pedregosa et al. 2011).

We hypothesize that the addition of shuffled positive bags (and new instance constraints) is particularly useful for the active learning setting, in which users are iteratively queried for labels in a manner that maximizes classifier accuracy while minimizing labeling effort (Cohn, Ghahramani, and Jordan 1996). Because active learning labels are acquired by querying a human user (at some expense), initial training sets contain very few labeled examples.

Multiple-Instance Active Learning

Active learning in the MI setting was explored in prior work (Settles, Craven, and Ray 2008), where instance labels are queried to improve an instance classifier initially trained with few labeled bags. In these experiments, we instead focus on the bag classification task in which a user is queried for the labels of bags to train a classifier for predicting the labels of bags in a test dataset. Previous work has specifically established the use of the NSK with MI active learning for image classification (Liu et al. 2009).

In our active learning experiments, we use the 25

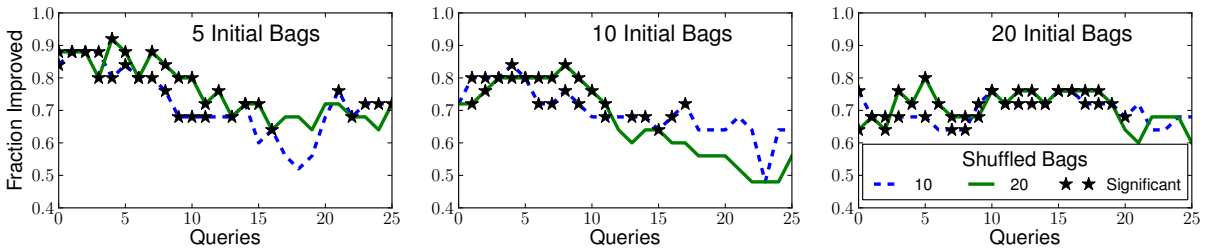

Figure 3: **(MI Active Learning)** Each plot shows an active learning experiment starting with 5, 10, or 20 initial bags. For both 10 and 20 shuffled bags, the y-axis plots the fraction of times (across the 25 datasets) SMILe outperforms the baseline of no shuffled bags as the number of bag label queries increases along the x-axis. Stars indicate a significant difference between SMILe and baseline AUC values using a 2-sided Wilcoxon signed-rank test at 0.05 significance.

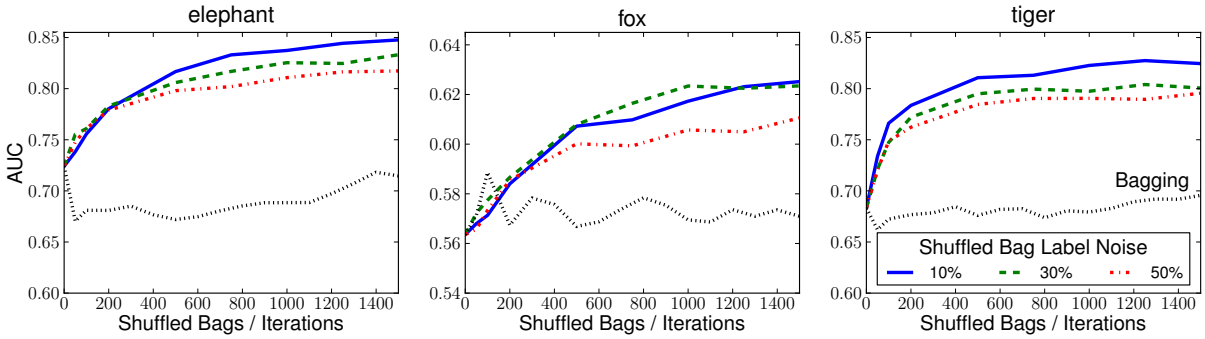

Figure 4: **(MI Classification)** Results of adding shuffled bags to MI classification datasets with shuffled bag sizes corresponding to various noise levels. The performance of bagging is also shown for comparison. The x-axis indicates either the number of shuffled bags for SMILe, or the number of bagging iterations used. Using a permutation test with a 0.001 significance level, the advantage of SMILe over the baseline (no shuffled bags) becomes significant after the addition of 50 shuffled bags for `elephant` and `tiger`, and 200 shuffled bags for `fox`. Similarly, SMILe significantly outperforms bagging on all three datasets.

instance-annotated Spatially Independent, Variable Area, and Lighting (SIVAL) datasets from prior work (Settles, Craven, and Ray 2008), but unlike that work, we only use bag-level labels during the training process. Results are averaged over 10 repetitions. Within each repetition, 5 folds are used to compute a pooled area under receiver operating characteristic (ROC) curve (AUC), which is averaged across repetitions. Within the training set corresponding to each fold, either 5, 10, or 20 initial bags of each class are randomly selected and labeled, and either 10 or 20 shuffled bags of each class are added to the initial dataset. The remaining training set bags are placed in a pool. Then, at each iteration of the active learning process, the label of an unlabeled bag is queried from the pool. We use the "simple margin" strategy (Tong and Koller 2002) for querying bag labels by choosing the bag closest to the SVM separator in the NSK feature space. After each query, the classifier is retrained and evaluated on a test set.

Shuffled bag size is determined using the formula in Proposition 1, with $\epsilon_s = 0.1$ for a worst-case label noise level of 10% on the shuffled bags. Because instances in these SIVAL datasets are labeled (but these labels are not used during training), we can compute the actual noise level, which is much smaller at 0.002%. The discrepancy between worst-case and actual noise occurs because there are frequently several positive instances per positive bag, whereas Proposition 1 assumes that there is only one positive instance per positive bag. Thus, in practice SMILe adds almost zero noise to the training set.

Figure 2 shows example learning curves for several SIVAL datasets. A plot for each dataset shows the AUC of classifiers trained with 20 initial bags, and either 0, 10, or 20 shuffled bags augmenting the initial dataset. These datasets illustrate cases when SMILe does (`feltflowerrug`, `stripednotebook`, `smileyfacedoll`, `cardboardbox`) and does not (`juliespot`, `rapbook`) improve performance.

Looking at the curves for `feltflowerrug` and `stripednotebook`, we see how shuffling appears to prevent overfitting. Without shuffled bags, overfitting seems to occur after approximately 15–25 bag label queries. However, with the addition of shuffled bags, classifier performance is maintained or improved even after 25 queries. For the `smileyfacedoll` dataset, shuffling seems to be especially useful during the first 20 queries when the training set is small. Finally, SMILe appears to accelerate the learn-

ing rate for the cardboardbox dataset so that shuffling improves performance across all 50 queries.

For the juliespot dataset, SMILe decreases performance by a nearly constant amount, perhaps because shuffled bags introduce more noise than useful constraint information in this case. The performance on the rapbook dataset begins similarly across various numbers of shuffled bags, but does not improve as quickly when shuffling is used. Although classifier performance is occasionally hindered by shuffled bags, there are many more cases in which it is improved.

Figure 3 summarizes the results of active learning curves across all 25 datasets, showing the relative performance of SMILe with various numbers of shuffled bags to the baseline algorithm that does not use shuffling. The plots show, for various numbers of initial labeled bags, the fraction of the 25 datasets for which SMILe outperforms the baseline as up to 25 bags are queried via the active learning process. A 2-sided Wilcoxon signed-rank test with a 0.05 significance level is used to compare paired AUC values across datasets for SMILe and the baseline at each point along the curve. Significant results are indicated with stars.

From Figure 3, we see that the addition of shuffled bags increases classifier performance, especially during the first 10–20 queries. With more than 5 initial bags, the performance of classifiers using various numbers of shuffled bags seem indistinguishable after approximately 25 queries. This is expected since in these experiments, additional shuffled bags are not generated as new labeled bags are added to the training set. If shuffled bags were continually added to the training set with each query returning a positive bag, performance would likely continue to improve above the baseline.

Multiple-Instance Classification

In this set of experiments, we test the hypothesis that the addition of shuffled bags improves classifier performance in the standard MI setting using the CBIR datasets elephant, fox, and tiger (Andrews, Tsochantaridis, and Hofmann 2003). As in the active learning experiments, results are averaged over 10 repetitions. For these datasets, 10 folds are used to compute pooled AUC. Within each fold, between 0 and 1500 additional positive and negative shuffled bags are added to the training set. The worst-case noise level of shuffled bag labels varied from 10% to 50% with the formula in Proposition 1 used to derive the appropriate shuffled bag size. We expect that the true label noise on shuffled bags is smaller due to multiple positive instances per positive bag, but that the utility of shuffled bags decreases as the label noise is increased.

The plots in Figure 4 show the results of shuffling for MI classification. For each dataset, the addition of shuffled bags clearly improves classifier performance, even after several hundred shuffled bags have been added. Furthermore, the performance improves even with a theoretical worst-case noise level of 50%. This seems to suggest that there are in fact multiple positive instances per positive bag, so the noise level is actually much lower than worst-case estimates. A permutation test to compare ROC curves of individual classifiers shows that improved performance becomes significant after 50 bags are added to elephant and tiger datasets, and after 200 bags are added to the fox dataset.

As we discuss in the analysis above, although SMILe is a resampling technique similar to bagging, we hypothesize that by resampling *instances* rather than *bags*, SMILe can provide an additional information about instance constraints. To demonstrate this effect, the MI bagging algorithm described in prior work (Zhou and Zhang 2003) is implemented and used to produce the results in Figure 4. These results show the advantage of SMILe over bagging and are also significant with $p = 0.001$ using ensembles of up to 1500 classifiers trained on bootstrap replicates.

Related Work

In the standard supervised learning setting, resampling and ensemble approaches such as bagging (Breiman 1996) and boosting (Freund and Schapire 1996) can improve classifier performance, for example by reducing the variance of algorithms that are unstable given small changes in the training set. Prior work investigates extending bagging to the MI setting by resampling entire bags to create bootstrap replicas of B (Zhou and Zhang 2003). Similarly, boosting is extended to the MI setting by iteratively training a weak classifier on a set of weighted bags (Auer and Ortner 2004). SMILe differs from previous approaches by resampling at the instance level rather than at the bag level. Furthermore, unlike previous MI resampling techniques, SMILe is not formulated as an ensemble method, but uses resampling to construct a single augmented dataset used to train any MI classifier. Future work will explore an instance resampling technique like in SMILe combined with an ensemble approach like bagging.

In the CBIR domain, some image classification tasks require stronger assumptions relating instance and bag labels (Zhou, Sun, and Li 2009). For example, the Coke cans in Figure 1 are segmented such that several positive instances are required before an image properly contains an entire Coke can. However, our analysis uses the standard MI setting assuming that it is sufficient for an image of an object to contain at least one part of that object.

Conclusion

We have presented a new resampling technique for MI learning, called SMILe. The approach works by resampling instances within positive bags to generate new bags that are positive with high probability. In addition to variance reduction that many resampling techniques afford, SMILe can introduce additional information in the form of instance label constraints to an MI classifier. In practice, we show that SMILe can significantly improve the performance of an MI classifier, and is particularly well-suited for domains such as active learning in which few initial labeled bags are available to a classifier.

Acknowledgements

We thank the anonymous reviewers for their feedback. G. Doran was supported by GAANN grant P200A090265 from the US Department of Education. S. Ray was partially supported by CWRU award OSA110264.

References

Andrews, S.; Tsochantaridis, I.; and Hofmann, T. 2003. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, 561–568.

Auer, P., and Ortner, R. 2004. A boosting approach to multiple instance learning. In *Machine Learning: ECML 2004*, volume 3201 of *Lecture Notes in Computer Science*. Springer. 63–74.

Blockeel, H.; Page, D.; and Srinivasan, A. 2005. Multi-instance tree learning. In *Proceedings of the 22nd International Conference on Machine Learning*, 57–64.

Breiman, L. 1996. Bagging predictors. *Machine Learning Journal* 24(2):123–140.

Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research* 4:129–145.

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

Freund, Y., and Schapire, R. E. 1996. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, 148–156.

Gärtner, T.; Flach, P.; Kowalczyk, A.; and Smola, A. 2002. Multi-instance kernels. In *Proceedings of the 19th International Conference on Machine Learning*, 179–186.

Liu, D.; Hua, X.; Yang, L.; and Zhang, H. 2009. Multiple-instance active learning for image categorization. *Advances in Multimedia Modeling* 239–249.

Maron, O., and Ratan, A. L. 1998. Multiple-instance learning for natural scene classification. In *Proceedings of the 15th International Conference on Machine Learning*, 341–349.

Maron, O. 1998. *Learning from Ambiguity*. Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research* 12:2825–2830.

Ramon, J., and Raedt, L. D. 2000. Multi instance neural networks. In *Proceedings of the ICML 2000 workshop on Attribute-Value and Relational Learning.*

Ray, S., and Craven, M. 2005. Supervised versus multiple instance learning: an empirical comparison. In *Proceedings of the 26th International Conference on Machine Learning*, 697–704.

Settles, B.; Craven, M.; and Ray, S. 2008. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, 1289–1296.

Tong, S., and Koller, D. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research* 2:45–66.

Xu, X., and Frank, E. 2004. Logistic regression and boosting for labeled bags of instances. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 272–281.

Zhang, Q., and Goldman, S. 2001. EM-DD: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems*, 1073–1080.

Zhou, Z.-H., and Zhang, M.-L. 2002. Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology*.

Zhou, Z., and Zhang, M. 2003. Ensembles of multi-instance learners. In *Machine Learning: ECML 2003*, volume 2837 of *Lecture Notes in Computer Science*. Springer. 492–502.

Zhou, Z.; Sun, Y.; and Li, Y. 2009. Multi-instance learning by treating instances as non-IID samples. In *Proceedings of the 26th International Conference on Machine Learning*, 1249–1256.