

Class Noise Mitigation through Instance Weighting

Umaa Rebbapragada and Carla E. Brodley

Dept. of Computer Science, Tufts University
161 College Ave., Medford, MA 02155, USA
urebbapr,brodley@cs.tufts.edu

Abstract. We describe a novel framework for class noise mitigation that assigns a vector of class membership probabilities to each training instance, and uses the confidence on the current label as a weight during training. The probability vector should be calculated such that clean instances have a high confidence on its current label, while mislabeled instances have a low confidence on its current label and a high confidence on its correct label. Past research focuses on techniques that either discard or correct instances. This paper proposes that discarding and correcting are special cases of instance weighting, and thus, part of this framework. We propose a method that uses clustering to calculate a probability distribution over the class labels for each instance. We demonstrate that our method improves classifier accuracy over the original training set. We also demonstrate that instance weighting can outperform discarding.

1 Introduction

Cleaning training data of mislabeled instances improves a classifier’s accuracy. Past research has focused on cleaning the training data by either discarding or correcting mislabeled instances. This paper proposes a different approach: assign a probability vector of class membership to each training instance, and use the confidence on the current label as a weight during training. Correcting and discarding techniques can also make use of this probability vector, and indeed, are special cases of instance weighting. A technique that discards is effectively assigning a 0 to the current label of an instance it discards and a 1 to the label of an instance it keeps. Similarly, a correcting technique assigns a confidence of 1 to the new class label, and 0 to all others.

An objective then is to find an accurate method for assigning confidences such that incorrect labels receive a low confidence and correct labels receive a high confidence. This paper presents pair-wise expectation maximization (PWEM), a confidence-labeling technique that clusters instances from pairs of classes and assigns to each instance a probability distribution over the class labels. We use the EM [3] algorithm to perform the clustering because it conveniently outputs the probability distribution of the instance’s cluster membership, but our approach can be used with any partitioning clustering technique.

We validate our method on three data sets in which we introduce both random and rule-based noise of up to 40%. Our experiments demonstrate that PWEM correctly assigns a low confidence to the mislabeled examples. We discuss limitations on PWEM to generate accurate confidences caused by noise level and class separability. We demonstrate that these confidences can be used to implement instance discarding and weighting, and show empirical evidence in favor of instance weighting over discarding. Finally, we demonstrate that PWEM in conjunction with instance weighting can significantly improve the accuracy of a classifier over the original mislabeled training set.

2 Instance Weighting

Instance weighting may be preferable to discarding when class noise mitigation techniques make errors. Discarding can lead to two types of errors: 1) discarding a clean instance and 2) retaining a mislabeled instance. Another disadvantage of discarding is that it reduces the training set’s size. If valuable minority class instances are mislabeled or erroneously identified as noise and discarded, the resulting classifier will not generalize well for examples of those classes. Correcting methods retain the full data set, but have the potential to maintain or introduce more noise into the labeling process via three types of errors: 1) changing the label on a clean instance, 2) retaining the label of a mislabeled instance, and 3) changing the label of a mislabeled instance to another incorrect label.

Instance weighting via confidences on the current label may be preferable over 0|1 weights because the full data set is retained and the penalty for making errors may be smaller. Let $P(l|x)$ be the confidence associated with instance x ’s current label l . The error associated with the instance is $P(l|x)$ if the instance is mislabeled, and $1 - P(l|x)$ if the instance is clean.

Consider a discarding technique that bases its decision on an instance’s confidence. One can synthetically assign these confidences to any discarding algorithm such that each instance it discards has a confidence below a threshold T and each instance it keeps has a confidence greater than T . As a best case scenario for instance weighting in comparison to discarding, imagine that both make mistakes on the same set of mislabeled and clean instances. Given a training set composed of a set of mislabeled instances M and a set of clean instances C (where x_i is an instance), and assuming all other confidences are correct, instance weighting errors are bounded by:

$$\sum_{i=1}^{|M'|} P(l|x_i) \leq |M'| \text{ where } M' = \{P(l|x_i) > T | \forall x_i \in M\}$$

$$\sum_{i=1}^{|C'|} (1 - P(l|x_i)) \leq |C'| \text{ where } C' = \{P(l|x_i) \leq T | \forall x_i \in C\}$$

Thus, in terms of error, instance weighting is penalized less for making mistakes on the same instances because the loss is not 0|1. Of course, instance weighting incurs a penalty on correct decisions while discarding does not. However,

our hypothesis is that weighting’s error gain on correct decisions is more than offset by its error reduction on mistakes. We defer an analytical proof of this hypothesis to future work. Meanwhile, Section 4 presents empirical evidence that weighting outperforms discarding. A similar argument applies for correcting versus weighting. The skew in error is even more pronounced under correcting as it is capable of three types of error rather than two. We defer both our analytical and empirical analysis of weighting versus correcting to future work.

3 Computing Confidence on the Class Labels

Instance weighting is only as effective as the quality of the confidences associated with each instance. PWEM is our method for assigning a probability distribution over the class labels to each instance. For each instance x , PWEM outputs the confidence $P(l|x)$ that the label of x is $l \in L$, where L is the set of class labels. In this section, we describe how we use clustering to find $P(l|x)$.

Intuitively, one expects instances from the same class to cluster together. We can use clustering to create a set of class probability vectors by having each instance inherit the distribution of classes within its assigned cluster. The drawback of this method is that there is no guarantee that a multi-class data set will cluster perfectly along class lines. Feature selection may improve class separability, but it is possible that two or more classes may not separate under any circumstances because their distributions overlap.

We improve class separability by clustering pairs of classes, and leave feature selection as an area of future work. For each of the $\binom{|L|}{2}$ pairs of classes, we cluster only those instances assigned a label in that pair. Thus, each instance belongs to only $|L| - 1$ clusterings. If an instance’s label has a low confidence in one clustering due to class inseparability, it may still receive a high confidence in its other clusterings if its assigned class separates well from others.

Our method, called PWEM, uses the EM algorithm to perform clustering.¹ Given a set of $|L| - 1$ clusterings for instance x , we calculate the probability that x belongs to class l as follows:

$$P(l|x) = \sum_{\theta} P(\theta)P(l|x, \theta) = \sum_{\theta} P(\theta) \sum_{c=1}^k P(l|c, \theta)P(c|x, \theta) \quad (1)$$

where l is a class label, x is an instance, c is a cluster, k is the number of clusters (determined by the Bayesian Information Criterion [9]), and θ is the given clustering model. $P(l|x)$ represents the probability that instance x should have class label l . $P(l|x, \theta)$ represents the probability that x should have label l given the clustering θ . This probability is calculated by summing the probability that x belongs to cluster c (as calculated by EM) times the probability that c should be labeled as l . Summing over all clusters results in the probability that x should be labeled l . If $P(l|c, \theta)$ and $P(c|x, \theta)$ form probability distributions, it

¹ Our implementation of EM estimates both the mean and variance of a finite mixture of Gaussians.

is trivial to show that $P(l|x)$ also forms a probability distribution over the class labels. We assume each clustering (θ) is equally likely. Thus, $P(\theta)$ is $\frac{1}{L-1}$. Each $P(l|x)$ acts as a confidence on the class label l for instance x .

4 Experiments

Our experimental goals are to 1) assess the quality of the confidences produced by PWEM, 2) demonstrate that PWEM in conjunction with instance weighting improves classifier accuracy over the mislabeled data set and 3) show empirical evidence in favor of instance weighting over discarding as a technique for class noise mitigation.

4.1 Data

We perform experiments on three data sets, referred to as segmentation (or segm), road, and modis. These are multi-class data sets with 2310, 2056 and 3398 instances and 7, 9 and 11 classes respectively. Segmentation has an even distribution of classes, while road and modis do not. For detailed descriptions, we refer the reader to Brodley & Friedl (1999) [1].

We perform ten runs of each experiment. For each run we randomly shuffle the data set and reserve 2/3 for training and 1/3 for testing. Both test and training sets are stratified samplings. Copies of the training set are mislabeled to have up to 30% random and 40% rule-based noise. Our experiments use Weka's [13] implementation of the C4.5 classifier, which allows the input of instance weights.

Random noise is introduced by randomly flipping $n\%$ instances of the training set, with noise introduced in proportion to class distribution. The new label is chosen uniformly among the other classes. We denote random noise levels of 10, 20, and 30% as MA10, MA20, and MA30.

Rule-based noise is introduced with the assistance of rules provided by a domain expert for each data set. These rules reflect the natural confusions that exist between classes. We use the rules outlined in Brodley & Friedl (1999) [1]. Each instance has a $n\%$ chance of being flipped according to its rule. Thus, the data set potentially has $n\%$ noise. The actual noise level is usually less than $n\%$. Mislabeled to ensure $n\%$ noise in the data can create pathological mislabeling among minority classes in cases where minority classes have mislabeling rules and majority classes do not. For rule-based noise, MU10, MU20, MU30 and MU40 indicate the potential noise levels of 10, 20, 30 and 40%. Table 2 provides the mean actual noise levels. In this paper, the unqualified use of the word 'noise' refers to potential noise. Only under rule-based noise is potential noise not equal to actual noise.

4.2 Quality of Confidences

To assess the efficacy of PWEM, we examine the confidences associated with the current label of the clean (C) and mislabeled (M) instances. For PWEM to

Table 1. Mean and standard deviation on the confidence of the current class label for clean (C) and mislabeled (M) instances for random (MA) and rule-based (MU) noise.

DATA	M/C	MA10	MA20	MA30	MU10	MU20	MU30	MU40
segm	M	.45 ± .22	.46 ± .19	.47 ± .15	.39 ± .18	.45 ± .16	.53 ± .17	.58 ± .17
segm	C	.82 ± .11	.77 ± .11	.73 ± .11	.83 ± .12	.78 ± .11	.74 ± .11	.72 ± .11
road	M	.47 ± .15	.46 ± .13	.47 ± .12	.62 ± .16	.63 ± .15	.65 ± .13	.65 ± .13
road	C	.85 ± .16	.83 ± .16	.81 ± .15	.86 ± .15	.84 ± .15	.82 ± .15	.80 ± .15
modis	M	.46 ± .19	.47 ± .16	.47 ± .13	.70 ± .17	.72 ± .16	.75 ± .15	.78 ± .14
modis	C	.86 ± .13	.83 ± .13	.79 ± .14	.90 ± .11	.89 ± .11	.88 ± .11	.87 ± .12

work effectively with instance weighting, there must be a separation between the means of the C and M instances.

Table 1 shows average confidences on the current label of clean and mislabeled instances. There are several observable trends in the results. First, as the level of noise increases, the separation between the confidences decreases. At best, that separation is approximately 0.4 (modis, MA10), and at worst it is 0.09 (segmentation, MU40). As class noise increases, it becomes more difficult for PWEM to assign high and low confidences to the clean and mislabeled data respectively. PWEM also has more difficulty separating confidences under rule-based noise than random noise. This is an expected result, as rule-based noise is a tougher problem. However, in all cases, the mean confidences of the mislabeled instances are always lower than the mean confidences of the clean instances. This will downweight the effect of the mislabeled instances in relation to clean ones.

PWEM currently does a poor job at assigning a high confidence to the true labels of mislabeled instances, rendering it ineffective as a label correction method. We leave label correction as an area of future work.

4.3 Weighting versus Discarding

We now compare the accuracy of classifiers on training sets created by discarding and weighting instances. First, we establish two baselines for classifier accuracy. The first, MK, is the accuracy of a classifier trained from the mislabeled training set (MK stands for mislabeled kept). The second, MD, is the accuracy of a classifier trained from only the clean instances of the training set (MD stands for mislabeled discarded). This is the accuracy achieved under perfect discarding or perfect confidences. We discard instances (DIST(T)) whose confidence on their assigned class label is less than a user-specified threshold T . We weight instances (WGHT) according to the confidence on their assigned labels.

Table 2 shows our accuracy results. Each cell of the table shows the mean and standard deviation of the accuracy for a method at the noise level indicated. Our results show that, in general, instance weighting achieves a better classifier accuracy than discarding (with a sole exception at MA10 in segmentation). Indeed, in all cases, instance weighting achieves accuracy within three percentage points of the MD upper limit. For all potential noise levels greater than 20, weighting improves accuracy significantly over the original mislabeled data (MK).

Table 2. Accuracy on segmentation, road and modis using random (MA) and rule-based (MU) noise. The MISL column shows potential noise while ACT reports the actual noise. Results compare instance weighting (WGHT) and discarding (DISC) at thresholds 0.2, 0.5 and 0.8.

DATA	MISL	ACT	MK	MD	DISC(.2)	DISC(.5)	DISC(.8)	WGHT
segm	MA10	10.0	94.1 ± 0.9	95.7 ± 0.8	94.8 ± 0.9	94.1 ± 0.8	70.3 ± 5.4	94.4 ± 0.8
segm	MA20	20.0	91.1 ± 1.2	95.3 ± 0.4	91.4 ± 1.3	91.7 ± 0.8	36.9 ± 9.6	93.4 ± 0.8
segm	MA30	30.0	82.2 ± 2.6	95.0 ± 1.0	82.9 ± 3.2	89.6 ± 1.8	21.3 ± 7.4	92.4 ± 1.5
segm	MU10	5.8	95.2 ± 0.7	95.9 ± 0.8	95.3 ± 0.7	94.9 ± 1.5	72.1 ± 6.4	95.3 ± 1.3
segm	MU20	11.4	93.5 ± 1.5	95.9 ± 0.9	94.0 ± 1.4	93.1 ± 1.3	56.0 ± 10.7	94.8 ± 1.2
segm	MU30	17.2	89.6 ± 1.9	95.9 ± 1.2	90.1 ± 1.8	91.8 ± 1.7	30.6 ± 10.7	94.8 ± 1.0
segm	MU40	22.9	84.9 ± 2.9	95.4 ± 1.0	84.7 ± 3.1	86.7 ± 4.3	25.6 ± 11.8	93.0 ± 1.6
road	MA10	10.0	78.5 ± 0.8	80.5 ± 2.0	78.3 ± 1.0	78.8 ± 1.5	78.4 ± 0.8	80.2 ± 1.3
road	MA20	20.0	76.9 ± 1.7	80.7 ± 1.4	76.9 ± 1.7	77.9 ± 1.1	74.2 ± 4.2	79.9 ± 0.5
road	MA30	30.0	69.6 ± 4.8	80.5 ± 1.8	69.6 ± 4.8	77.3 ± 1.2	71.1 ± 2.7	79.9 ± 1.0
road	MU10	9.7	79.1 ± 1.3	80.2 ± 1.2	79.2 ± 1.4	78.6 ± 1.5	77.1 ± 1.6	80.8 ± 1.0
road	MU20	19.2	76.3 ± 2.9	80.7 ± 1.3	76.7 ± 2.8	76.0 ± 2.4	76.2 ± 3.2	80.7 ± 1.1
road	MU30	30.0	67.5 ± 2.6	80.1 ± 1.4	67.7 ± 2.5	68.3 ± 2.8	72.3 ± 3.3	80.3 ± 0.7
road	MU40	39.6	59.7 ± 4.8	80.0 ± 1.2	60.8 ± 4.8	59.6 ± 4.6	71.0 ± 2.3	79.6 ± 0.8
modis	MA10	10.0	84.0 ± 0.9	85.5 ± 0.9	84.0 ± 0.9	84.6 ± 0.9	77.2 ± 1.8	85.4 ± 0.8
modis	MA20	20.0	80.5 ± 1.2	84.7 ± 1.0	80.5 ± 1.2	83.3 ± 1.5	65.7 ± 2.4	83.9 ± 1.4
modis	MA30	30.0	75.9 ± 2.4	84.5 ± 1.3	75.9 ± 2.4	81.9 ± 1.6	56.6 ± 5.5	83.7 ± 1.3
modis	MU10	6.6	84.4 ± 0.6	85.6 ± 0.8	84.4 ± 0.6	85.1 ± 1.1	81.3 ± 1.0	84.9 ± 0.7
modis	MU20	13.4	81.5 ± 1.4	85.1 ± 0.9	81.6 ± 1.6	82.4 ± 1.1	80.1 ± 1.3	84.2 ± 1.0
modis	MU30	19.5	79.1 ± 2.1	85.5 ± 1.0	79.3 ± 2.2	78.6 ± 1.4	76.9 ± 2.4	82.6 ± 1.8
modis	MU40	26.6	74.2 ± 2.7	84.5 ± 0.9	74.3 ± 2.8	75.1 ± 1.9	68.6 ± 2.5	77.9 ± 1.9

For space reasons, Table 2 shows discard thresholds of 0.2, 0.5 and 0.8 only. However, we performed experiments with discard thresholds 0.1, 0.2, . . . , 0.9. Our results show that weighting results in a higher classifier accuracy than discarding at all nine threshold levels. Discarding performs better than the original mislabeled training set (MK) up until a certain threshold. This point is generally just below the mean confidence value for clean instances. At this point, too many clean instances have been discarded and classifier performance deteriorates. Table 2 shows that weighting never falls below MK, and of the two methods, is the more reliable class mitigation technique.

5 Related Work

With the exception of Lawrence and Schölkopf [7], existing methods discard [15, 5, 4, 2, 1, 11, 12, 16], correct [14, 10] or are capable of both [8, 6]. The majority of existing work in class noise mitigation evaluates their methods on random-noise only. As demonstrated by our results, and discussed in Brodley and Friedl [1] and Zhu et al. [16], rule-based noise is more difficult to eliminate than random noise. Our approach differs from existing work by introducing instance weighting

as a technique for class noise mitigation. We also experiment on both types of noise.

Methods for discarding differ in how they determine which instances are mislabeled. Using a n -fold CV, Brodley and Friedl discard instances that fail to get either a majority or consensus vote from the ensemble that matches their label. Verbaeten [11, 12] builds ensembles on different subsets of the data (e.g., via bagging) and identifies mislabeled instances as those that receive high weights by boosting. Zhu et al. [16] propose an iterative algorithm that partitions the data set and creates rules for each subset. A set of “good rules” are used to classify an instance as noise by either majority vote or consensus. The noisy instances along with a small set of good examples are then set aside as the method repeats on the reduced data set until a stopping criterion is met. Gamberger et al. [5, 4] use compression-based measures to eliminate noise from the training set. A training instance is discarded if its elimination reduces the complexity of the hypothesis on the training set. This process iterates until a user-specified noise-sensitivity threshold is reached. Zeng and Martinez [14, 15] calculate a probability distribution over the class labels for each instance, which they use to discard or correct instances. One could adapt their method to use instances weights. Our hypothesis is their accuracy results will improve. Whether their method provides better confidences than PWEM is an open question to be addressed in the future.

Lawrence and Schölkopf’s [7] method for two class problems learns the conditional probabilities that a class label is flipped. These probabilities are learned at the class level rather than the instance level. The method uses a kernel Fisher discriminant in conjunction with the EM algorithm to iteratively estimate class conditional probabilities that indicate mislabeling.

Finally, class noise mitigation is closely related to instance selection techniques, which were designed either to improve computational efficiency or improve classification accuracy by discarding the data (see [14] for a good overview).

6 Conclusion and Future Work

This paper presents a new technique that reduces the effects of class noise. PWEM clusters pairs of classes to gain insight into the true class labels of the training set. We present empirical evidence that PWEM in conjunction with instance weighting significantly improves classifier accuracy close to the theoretical best on all noise levels. In cases where it does not, weighting significantly improves classifier accuracy over the original mislabeled set.

An area of future work is to improve PWEM’s ability to calculate the probability distribution over the class labels. In particular, improvement of PWEM’s ability to predict the true class label of a mislabeled instance will enable it to be used as a correcting technique. Other ideas to improve PWEM include weighting the influence of the each clustering according to its class separability and size, and incorporating feature selection. Finally, we will investigate a wider range of weighting techniques that use the full probability vector, rather than the confidence on the current label only.

References

1. Brodley, C. E., Friedl, M. A.: Identifying mislabeled training data. *JAIR* **11** (1999) 131–167
2. Brodley, C. E., Friedl, M. A.: Identifying and eliminating mislabeled training instances. In *Proc. of the 13th National Conference on Artificial Intelligence* (1996) 799–805
3. Dempster, A. P., Laird, N. M., Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* **39** (1977) 1–38
4. Gamberger, D., Lavrač, N., Grošelj, C.: Experiments with noise filtering in a medical domain. In *Proc. of the 16th ICML* (1999) 143–151
5. Gamberger, D., Lavrač, N., Džeroski, S.: Noise elimination in inductive concept learning: a case study in medical diagnosis. In *7th Int. Wkshp. on Algorithmic Learning Theory* (1996) 199–212
6. Lallich, S., Muhlenbach, F., Zighed D. A.: Improving classification by removing or relabeling mislabeled instances. In *Proc. of the 13th Int. Symp. on the Foundations of Intelligent Systems* (2002) 5–15
7. Lawrence, N. D., Schölkopf, B.: Estimating a Kernel Fisher Discriminant in the presence of label noise. In *Proc. of the 18th ICML* (2001) 306–313
8. Muhlenbach, F., Lallich, S., Zighed, D.A.: Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems* **22** 2004 89–109
9. Pelleg, D., Moore, A.: X-means: extending K-means with efficient estimation of the number of clusters. In *Proc. of the 17th ICML* (2000) 727–734
10. Teng, C.: Correcting noisy data. In *Proc. of the 16th ICML* (1999) (239–248)
11. Verbaeten, S.: Identifying mislabeled training examples in ILP classification problems. In *Proc. of the Machine Learning Conference of Belgium* (2002)
12. Verbaeten, S., Van Assche, A. Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems, 4th International Workshop* (2003)
13. Witten, I. H., Frank, E.: *Data mining: practical machine learning tools and techniques*, 2nd edition. (2005) Morgan Kaufmann, San Francisco
14. Zeng, X., Martinez, T.: An algorithm for correcting mislabeled data. *Intelligent Data Analysis* **5** (2001)
15. Zeng, X., Martinez, T.: A noise filtering method using neural networks. In *Proc. of the Int. Wkshp. of Soft Computing Techniques in Instrumentation, Measurement and Related Applications* (2003)
16. Zhu, X., Wu, X., Chen, S.: Eliminating class noise in large datasets. In *Proc. of the 20th ICML* (2003) 920–927