Ensemble Noise Filtering for Streaming Data using Poisson Bootstrap Model Filtering

Ashwin Satyanarayana and Rosemary Chinchilla

Department of Computer Systems Technology,
New York City College of Technology (CUNY),
N-913, 300 Jay St, Brooklyn, NY - 11201.
asatyanarayana@citytech.cuny.edu
rosemary.chinchilla@mail.citytech.cuny.edu

Abstract. Ensemble filtering techniques filter noisy instances by combining the predictions of multiple base models, each of which is learned using a traditional algorithm. However, in the last decade, due to the massive increase in the amount of online streaming data, ensemble filtering methods, which largely operate in batch mode and requires multiple passes over the data, cause time and storage complexities. In this paper, we present an ensemble bootstrap model filtering technique with multiple inductive learning algorithms on several small Poisson bootstrapped samples of online data to filter noisy instances. We analyze three prior filtering techniques using Bayesian computational analysis to understand the underlying distribution of the model space. We implement our and other prior filtering approaches and show that our approach is more accurate than other prior filtering methods.

1 Introduction

Data stream mining is the process of extracting knowledge structures from continuous, rapid data records. The goal in many data stream mining applications is to form a generalization, from a set of previous labeled training data streams such that classification accuracy for previously unobserved instances is maximized. The maximum accuracy achievable depends on the quality of the data and on the chosen learning algorithms. This paper focuses on improving the quality of training data by identifying and eliminating mislabeled instances on streaming data.

Quinlan [1] demonstrated that as noise level increases, removing noise from the mislabeled training instances (class noise) increases the predictive accuracy of the resulting classifier. This has been empirically verified by many researchers over the last decade. For example, Brodley and Friedl ([2][3]) illustrate that for class noise levels of less than 40%, removing mislabeled instances from the training set resulted in higher predictive accuracy relative to classification achieved without "cleaning" the training data.

Gamberger et al [4] point out that the separation of noise detection and hypothesis formation has the advantage that noisy examples do not influence hypothesis construction. This is different from the approaches that handle noise in the hypothesis formation process by trying to avoid overfitting the noisy example set.

The remainder of the paper is organized as follows. Section 2 discusses three prior significant filtering techniques. In Section 3, we mention the contributions of this paper. In Section 4 we analyze the current filtering techniques with Bayesian computational analysis. In Section 5 we present our bootstrap model filtering algorithm. In Section 6 we show empirically how our technique compares with other filtering techniques, and we finally conclude in Section 7.

2 Prior Work in Ensemble Filtering

The problem of handling noise has been the focus of much attention in machine learning especially in online data stream mining. For example, pruning in decision trees is designed to reduce the chance that the tree is overfitting to noise in the training data.

Weisberg [5] suggested the use of regression analysis to identify outliers in the training data. Those cases that could not be described by the model and have the largest residual errors are outliers. Motivated by the same idea, filtering was introduced to remove mislabeled training data. In this section, we survey three significant results in the area of filtering.

2.1 Ensemble Filtering

An ensemble classifier detects noisy instances by constructing a set of classifiers (base level detectors) [2]. A majority vote filter tags an instance as mislabeled if more than half of the m classifiers classify it incorrectly. A consensus filter requires that all classifiers must fail to classify an instance as the class given by its training label. In their empirical results, it was shown that the majority filter performs better than a consensus filter.

2.2 Classification Filtering

Classification filtering [4] is a filtering approach in which the training set E is partitioned into n subsets, and the set of classifiers trained from the aggregation of any $n-1$ subsets are used to classify the instance in the excluded subset. There is only one base-level classifier used in classification filtering.

2.3 Partition Filtering

Partition filtering proposed by Zhu et al [6] partitions a dataset into equally sized subsets, and each instance is evaluated by more than one model, and the results are averaged to identify the noisy instances. The major difference between classification and partition filtering is that in partition filtering each instance is evaluated by $k-1$ hypotheses, whereas in classification filtering, each instance is evaluated once by a single model built from the remaining instances in the other $k-1$ partitions.

3 Contributions of this paper

The novel contributions of this paper are as follows:

1. We perform Bayesian analysis to understand the underlying noise model distribution in ensemble techniques, and show that the three prior filtering techniques use crude approximations of model averaging (Section 4).
2. We present our novel ensemble filtering technique which extends the model space by Poisson bootstrap samples and learns on those samples using multiple inductive learners (Section 5).
3. We compare our approach with a single filter, a prior filtering technique (partition filtering) and online bagging, and plot the different techniques using (a) an accuracy vs noise level graph and (b) learning curves.

4 Bayesian Analysis

Ensemble techniques can be considered as performing *model averaging*. Model averaging framework is to build a set of models, obtain model specific estimates and use the rules of probability to average over all the instances. It has been shown that model averaging works by extending the model space [9] and that several classifiers produces a higher accuracy as compared to any single model [7]. In this section we use Bayesian analysis to understand the noise model distribution in ensemble techniques.

Let $\vec{x}$ represent the original training set, $\vec{y}$ the corresponding class labels and θ a model (or hypothesis) in the model space Θ. Mathematically, an unseen test instances x, is assigned to a class y that maximizes the following equation:

$$Pr(y|\vec{x},\Theta) = \sum_{\theta\in\Theta} Pr(y|x,\theta).Pr(\theta|\vec{x},\vec{y}) \tag{1}$$

By Bayes' theorem, and assuming the instances are drawn independently, the posterior probability of θ is given by:

$$Pr(\theta|\vec{x},\vec{y}) = \frac{Pr(\theta)}{Pr(\vec{x},\vec{y})}\prod_{i=1}^{n} Pr(x_i, y_i|\theta) \tag{2}$$

where $Pr(\theta)$ is the prior probability of θ, and the product of $Pr(x_i, y_i|\ \theta)$ is the likelihood. The data prior $Pr(\vec{x},\vec{y})$ is the same for all models and can thus be ignored.

In order to compute the likelihood in equation (2), it is necessary to compute the probability of a class label y_i, given an unlabeled instance x_i, and the hypothesis θ since:

$$Pr(x_i, y_i|\theta) = Pr(x_i,\theta).Pr(y_i|x_i,\theta) \tag{3}$$

The probability $Pr(y_i|x_i,\theta)$ is called the *noise model*. We assume a uniform class model in which each instance's class is corrupted with probability ϵ, and thus:

$Pr(y_i|x_i, \theta) = 1 - \epsilon$ if θ predicts the correct class y_i, and
$Pr(y_i|x_i, \theta) = \epsilon$ if θ predicts an incorrect class
Equation (2) then becomes:

$$Pr(\theta|\vec{x}, \vec{y}) \propto Pr(\theta)(1 - \epsilon)^s \epsilon^{n-s} \tag{4}$$

where s is the number of instances correctly classified by θ, and the noise level can be estimated by the models' average error rate.

Using Equation (4) in (1) we get:

$$Pr(y|\vec{x}, \Theta) \propto \sum_{\theta \in \Theta} Pr(y|x, \theta).Pr(\theta).(1 - \epsilon)^s \epsilon^{n-s} \tag{5}$$

As $n \to \infty$ the distribution of the noise model tends to $Poisson(1)$ distribution. Therefore for each training instance (as discussed in Section 5), we choose that instance $K \sim Poisson(1)$ times independently. We now analyze the computation of the three prior filtering techniques.

4.1 Bayesian Analysis of Ensemble Filtering

In the case of a majority vote filter, the probability of classifying an unseen instance x to a true label t, can be represented by the following equation:

$$Pr(y = t|x, \Theta) = \frac{1}{|\Theta^*|} \sum_{\theta \in \Theta^*} \delta(t, \theta(x)) \tag{6}$$

where $\delta(t, \theta(x)) = 1$ if the class label predicted by the classifier for the test case x is the true label t, and 0 otherwise. Θ^* represents the most probable classifiers of the model.

If $Pr(y = t|x, \Theta) < \frac{1}{2}$, then the instance is noisy and filtered.

4.2 Bayesian Analysis of Classification Filtering

As mentioned in the previous section, classification filtering performs k-fold cross validation. K-fold cross validation is a commonly used technique which takes a set of n examples and partitions them into k sets of size $\frac{n}{k}$. For each fold, a classifier is trained on the other folds and tested on the current fold. Thus k hypotheses $\theta_1, \theta_2, \ldots, \theta_k$ are generated. This prediction is equivalent to outputting the average of k-hypotheses as shown below:

$$Pr(y = t|x, \Theta) = \frac{1}{k} \sum_{i=1}^{k} \delta(t, \theta_i) \tag{7}$$

where δ is a 0-1 loss function, which returns 1 if θ_i predicts the correct label t, else returns 0. Equation (7) captures the cross validation step of the classification filtering algorithm. If t is the true class label, and θ is the final model built, then it performs the following computation:

$$Pr(y = t|x) = Pr(y = t|\vec{x}, \theta).Pr(\theta|\vec{x}, \vec{y}) \tag{8}$$

If $Pr(y = t|x) < \frac{1}{2}$, then the instance is noisy and filtered.

4.3 Bayesian Analysis of Partition Filtering

In the case of partition filtering, there are k different hypotheses and unlike the previous case, each instance is evaluated on each of the $k-1$ hypotheses. If more than half of the hypotheses classify the instance as noisy, then it is eliminated. That is, each instance is evaluated by an average of the other models built.

$$Pr(y = t|x) = \frac{1}{k-1} \sum_{i=1}^{k-1} \delta(t, \theta_i) \tag{9}$$

where δ is a 0-1 loss function, which returns 1 if θ_i predicts the correct label t, else returns 0

If $Pr(y = t|x) < \frac{1}{2}$, then the instance is noisy and filtered.

5 Bootstrap Model Filtering

In all the prior filtering techniques, we find that by having more than one model to evaluate the instances, they extend the model space as compared to a single filter. We also observe that in all the three techniques, a crude approximation of *model averaging* is performed.

Consider the typical view of mining where a single training set of size n is available from the underlying distribution that generated the data F. However this view masks the underlying uncertainty in the data, namely that the training data we have is one of many that could have been chosen. If we were to build a model for each possible data set we would have a probability distribution over the model space. However, typically we do not have the luxury of many different data sets drawn independently of the same size so that we can compute the uncertainty over the model space from them. To approximate this distribution using a single dataset, Efron [8] created the bootstrapping approaches. Non-parametric bootstrapping is an example of attempting to simulate draws from F, where no knowledge of its form is known. Formally, the computation performed would be:

$$Pr(y|\vec{x}, \Theta) = \sum_{D', \theta \in \Theta} Pr(y|x, \theta).Pr(D'|D) \tag{10}$$

where the datasets D' are the different bootstrap samples generated from the original dataset D.

The standard bootstrap procedure creates each simulated data set by drawing observations from x with replacement. In any given resample, each observation may occur 0, 1 or more times according to $Binomial(n, \frac{1}{n})$. And since the total number of observations is constrained to be n, the counts are jointly $Multinomial(n, \frac{1}{n}, \ldots, \frac{1}{n})$. We also note that:

$$\lim_{n \to \infty} Binomial(n, \frac{1}{n}) = Poisson(1) \tag{11}$$

Therefore for each training instance, we choose that instance $K \sim Poisson(1)$ times independently. We refer to this as *Poisson Bootstrap Sampling*.

Our Bootstrap Model Filtering algorithm as shown in Figure 1 begins with n Poisson bootstrapped samples of our dataset E (step 1) and an empty output set A of detected noisy examples (step 2). The main loop (steps 3-12) is repeated for each bootstrap sample E_i. E_i is used as an input for the k inductive learning algorithms to generate models k models $\theta_{i,1}, \theta_{i,2} \cdots \theta_{i,k}$. In step 7, we form a set E_t which includes all the examples from E except E_i. The set E_t is evaluated by our k models in steps 8-11. If more than half of the models misclassify an instance, then it is treated as noise and eliminated.

```
Algorithm  BootstrapModelFiltering ($E, n, k$)
Input: $E$ (training set),
    $n$ (number of Poisson bootstrap samples),
    $k$ (number of inductive learning algorithms)
Output: $A$ (a detected noisy subset of $E$)
 1: Form $n$ Poisson bootstrap samples $E_{1,...,n}$ of $E$
 2: $A \leftarrow \emptyset$
 3: for $i = 1, \ldots, n$ do
 4:    for $j = 1, \ldots, k$ do
 5:       $\theta_{i,j} \leftarrow$ model built from bootstrap sample $E_i$ and inductive algorithm $j$
 6:    end for
 7:    $E_t \leftarrow E \backslash E_i$
 8:    for every $e \in E_t$ do
 9:       If $e$ is misclassified by more than half of the $\theta_{i,j}$ models built, then it is noisy
          and needs to be eliminated.
10:       $A \leftarrow A \cup \{e\}$
11:    end for
12: end for
```

Fig. 1. Bootstrap Model Filtering Algorithm

6 Empirical Results

In this section, we discuss experiments using J48, RandomForest and RandomTree as the base classifier models for the bootstrap model filtering approach. We tested our approach on several UCI [12] and other datasets of varying (small, medium and large) sizes. For each dataset, we compare the accuracies after filtering using the following techniques:

1. Single Model: We used decision trees (J48) as our single filtering base model.
2. Partition Filtering: Partition Filtering is chosen as it outperforms Classification and Ensemble Filtering techniques [6]. We use Decision Tree as the base model.
3. Online Bagging: We implemented online bagging as illustrated by Oza [10] using Nave Bayes as the base model.

6.1 Small Datasets

We compare our approach on small UCI datasets (in the area of medicine). Medicine is an area where the quality of the data is essential for accurate prediction. The datasets selected are: Diabetes, Hepatitis, Liver Disorders and Dengue [11] disease as shown in Table 1.

Table 1. Sizes of Small Datasets

Dataset	**# of instances**	**# of attributes**	**# of classes**
Diabetes	768	9	2
Liver Disorder	345	7	2
Hepatitis	155	20	2
Dengue	846	18	2

For each dataset, for bootstrap model filtering we use 30 Poisson bootstrap samples with 3 different classifiers (J48, RandomForest and RandomTree) on each sample. The results of our experiments on small datasets are shown in Table 2.

Table 2. Classification Accuracy after eliminating noisy instances

Dataset	**Decision Tree (J48)**	**Partition Filtering**	**Online Bagging**	**Bootstrap Model Filtering**
Diabetes	0.73	0.75	0.79	**0.88**
Liver Disorder	0.68	**0.74**	**0.76**	**0.79**
Hepatitis	0.83	0.84	0.88	0.89
Dengue	0.74	0.77	0.81	**0.92**

6.2 Medium and Large Datasets

In this section, we used our approach on several medium and large datasets as shown in Table 3. In each of the medium datasets (Abalone, CMC, Car Evaluation, Multiple Features, Waveform and Wine Quality) we use 50 Poisson bootstrap samples. For the massive datasets (Nursery, Letter, Electricity board, and Localization) we used 100 Poisson bootstrap samples. The results are as shown in Table 4.

The boldface entries in Table 2 and Table 4 represent cases when the ensemble technique significantly (t-test, $\alpha = 0.05$) outperformed a single filtering technique. As we can see from the results, for small (e.g. Diabetes) and very large sized datasets (e.g. Localization, Electricity Board), we see that our technique outperforms the other techniques significantly. For medium sized, the ensemble techniques show a very small improvement in accuracy.

Table 3. Sizes of the Medium and Large datasets

Dataset	# of instances	# of attributes	# of classes
CMC	1473	10	3
Car Evaluation	1728	7	4
Multiple Features	2000	48	10
Abalone	2924	9	28
Waveform	3500	41	3
Wine Quality	3429	12	11
Nursery	12960	9	5
Letter Recognition	20000	17	26
Electricity Board	45781	5	31
Localization	164860	8	11

Table 4. Classification Accuracy after Filtering

Dataset	Decision Tree (J48)	Partition Filtering	Online Bagging	Bootstrap Model Filtering
CMC	0.52	0.65	**0.69**	**0.75**
Car Evaluation	0.92	0.94	0.95	0.97
Multiple Features	0.72	0.74	0.76	**0.78**
Abalone	0.20	0.34	**0.38**	**0.48**
Waveform	0.75	0.76	0.77	**0.82**
Wine Quality	0.56	0.58	0.61	**0.69**
Nursery	0.97	0.97	0.98	0.98
Letter Recognition	0.87	0.88	0.88	0.89
Electricity Board	0.65	0.66	0.68	**0.73**
Localization	0.76	0.73	0.79	**0.89**

6.3 Classification Accuracy vs Noise

Figure 2 shows the effect of filtering on classification accuracy on the test set by each of the four techniques. We introduce noise levels from 10% to 40% on each of the datasets by adding mislabeled instances in the test set. At 10% noise level we see that there is no significant difference for any of the ensemble techniques used, however there is significant difference when compared to the single filter. For noise levels up to 20%, we see that online bagging and bootstrap model filtering significantly outperform the other two techniques. For noise levels between 30% and 40%, bootstrap model filtering outperforms the other techniques significantly. We also see that bootstrap model filtering approach does well on very large datasets with higher noise levels.

6.4 Learning Curve Comparison

To illustrate the learning curve convergence in all the techniques, we tried the car evaluation dataset, which has about 1728 instances. We used 528 as the test set,

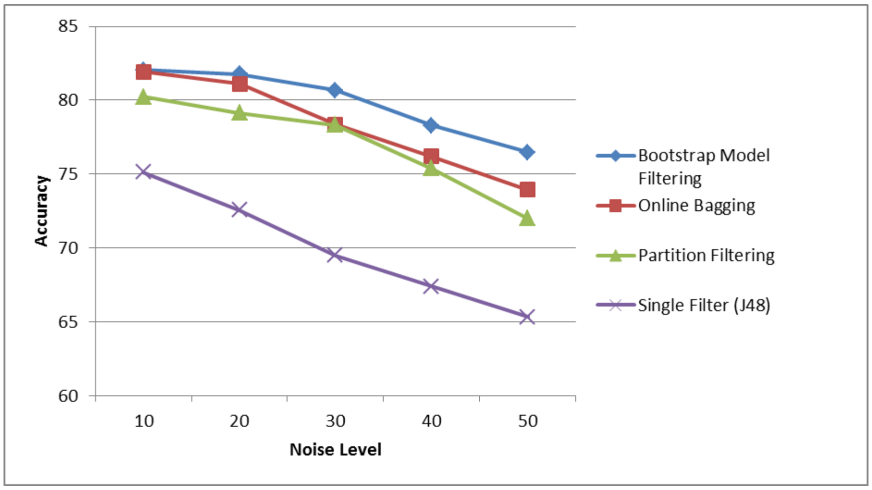

Fig. 2. Simulation results for the network.

and used 200, 400, 600, 800 and 1000 instances as streams of training data. We see in Figure 3 that bootstrap averaged ensemble technique performs better than the other two ensemble technique in terms of the number of instances needed for accuracy convergence, as the inductive learning algorithm learns from several small bootstrapped samples.

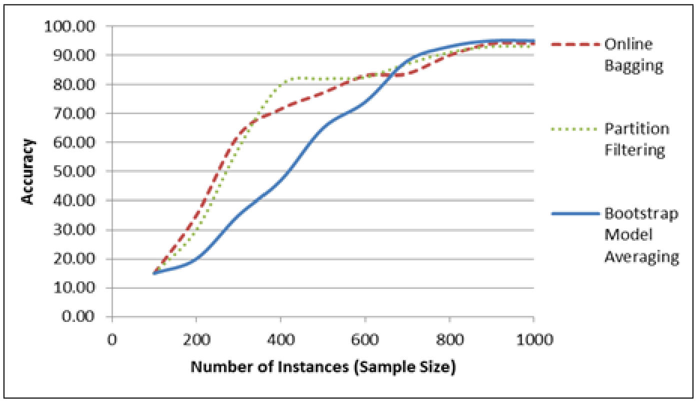

Fig. 3. Learning curve convergence of ensemble filtering techniques for the car evaluation dataset.

7 Conclusion

Resolving data quality issues in stream mining is often one of the biggest efforts in data mining. Prior filtering techniques required multiple passes of data. The main challenge in applying prior filtering techniques to data streams is that a stream is theoretically infinite and that means that data has to be processed in a single pass using little memory. Also, using all the available data is prohibitive due to memory and time constraints. In this paper, we attempt to address this problem for different sized datasets by using Poisson bootstrap samples with multiple base classifiers.

We analyzed prior filtering techniques by using Bayesian computation analysis. We then used that knowledge and extended the model space by using 1) several bootstrap samples and 2) multiple classifiers. Our empirical results show that although online bagging performs well for medium sized datasets in terms of computational size required, the bootstrap model filtering approach outperforms other approaches in the case of small and massive datasets.

Acknowledgment

We would like to thank the Emerging Scholars undergraduate research program at New York City College of Technology for supporting Rosemary Chinchilla in this research. Ashwin Satyanarayana's research was supported by PSC CUNY grant # 68180-00 46 of the Research Foundation of City University of New York (CUNY).

References

1. Quinlan, J. R., "Induction of decision trees," *Machine learning, 1(1)*, pp. 81-106, 1986.
2. Brodley, C. E., and Friedl, M. A., "Identifying and eliminating mislabeled training instances," In *AAAI/IAAI*, Vol. 1, pp. 799-805, 1996.
3. Brodley, C. E., and Friedl, M. A., "Identifying mislabeled training data," In *Journal of Artificial Intelligence Research*, pp. 131-167, 1999.
4. Gamberger, D., Lavrac, N., and Groselj, C., "Experiments with noise filtering in a medical domain," In *ICML*, pp. 143-151, 1999.
5. Weisberg, S., "Applied linear regression," (Vol. 528), John Wiley and Sons, 2005.
6. Zhu, X., Wu, X., and Chen, Q., "Eliminating class noise in large datasets," In *ICML* Vol. 3, pp. 920-927, August 2003.
7. Davidson, I., and Fan, W., "When efficient model averaging out-performs boosting and bagging," In *Knowledge Discovery in Databases: PKDD 2006*, pp. 478-486, 2006.
8. Efron, B., and Tibshirani, R. J., "An introduction to the bootstrap," CRC press, 1994.
9. Satyanarayana, A., "Data mining for large datasets: intelligent sampling and filtering," *Doctoral Dissertation.* State University of New York, Albany, 2006.
10. Oza, N. C., "Online bagging and boosting," In *IEEE International Conference on Systems, man and cybernetics*, Vol. 3, pp. 2340-2345, October 2005.
11. Shakil, K. A., Anis, S., and Alam, M., "Dengue disease prediction using weka data mining tool," arXiv preprint arXiv:1502.05167, 2015.
12. Lichman, M., "UCI Machine Learning Rep." [`http://archive.ics.uci.edu/ml`]. Irvine, CA. University of California, School of Information and Computer Science, 2013.