Avoiding Boosting Overfitting by Removing Confusing Samples

Conference Paper · September 2007

DOI: 10.1007/978-3-540-74958-5_40 · Source: DBLP

2 authors:

Alexander Vezhnevets
The University of Edinburgh
20 PUBLICATIONS **988** CITATIONS
SEE PROFILE

Olga Barinova
Lomonosov Moscow State University
30 PUBLICATIONS **999** CITATIONS
SEE PROFILE

Avoiding Boosting Overfitting by Removing Confusing Samples

Alexander Vezhnevets, Olga Barinova

Moscow State University, dept. of Computational Mathematics and Cybernetics,
Graphics and Media Lab
119992 Moscow, Russia
{avezhnevets, obarinova}@graphics.cs.msu.ru

Abstract. Boosting methods are known to exhibit noticeable overfitting on some datasets, while being immune to overfitting on other ones. In this paper we show that standard boosting algorithms are not appropriate in case of overlapping classes. This inadequateness is likely to be the major source of boosting overfitting while working with real world data. To verify our conclusion we use the fact that any overlapping classes' task can be reduced to a deterministic task with the same Bayesian separating surface. This can be done by removing "confusing samples" – samples that are misclassified by a "perfect" Bayesian classifier. We propose an algorithm for removing confusing samples and experimentally study behavior of AdaBoost trained on the resulting data sets. Experiments confirm that removing confusing samples helps boosting to reduce the generalization error and to avoid overfitting on both synthetic and real world. Process of removing confusing samples also provides an accurate error prediction based on the work with the training sets.

1. Introduction

Problem of overfitting is one of the key problems in machine learning. Boosting was first believed to be immune to overfitting. It was even reported to eventually lower its test error while training after the training error reaches zero. Later, Dietterich [4] found that boosting is very sensitive to noise and overfits it greatly. Grove [11] and Friedman et al [9] noted that boosting actually overfits on some real-world datasets, although much less then one should expect from such general model after considerable amount of iterations.

The best explanation of boosting generalization capabilities so far is margin theory [26]. It was put under serious doubt by Briemans experiments, but was rehabilitated recently [22]. Margin theory provides an upper generalization bound independent of number of iterations made by boosting. This bound suggests that boosting may not overfit even if ran for many rounds. But as stated by authors: "unfortunately, however, in their current form, our upper bounds are too pessimistic to be used as actual numerical estimates of the error". Although margin theory explains why boosting may not overfit it does not provide any explanation why boosting actually does overfit in practice on real world data even with constant complexity base learner (stump). Domingos [5] showed the relation between margin theory explanation of boosting and bias-variance explanation. He also made an interesting statement that reducing vari-

ance (increasing margins) is beneficial only for unbiased samples, while for biased samples it is preferable to have high variance (lower margin). Biased sample is a sample, for which the optimal prediction, for a given loss and the family of classifiers, differs from its current label. Freund & Schapire [7] in their discussion on Friedmans paper suggest that for overlapping classes (when Bayesian error is not zero) "AdaBoost is not the optimal method in this case". It should be noted that in real world applications it is a rare case if Bayesian error is zero and classes are perfectly separable due to imperfect feature vector representation of objects, limitations of measuring equipment and noise.

2. Related work

After the discovery of the fact that boosting does overfit many works were devoted to explaining and avoiding this phenomenon. Several authors reported that boosting tends to increase the weights of few hard-to-learn samples, which leads to overfitting. Several modifications of the reweighting scheme were proposed that make weights change more smoothly. Domingo et al. [6] propose a modification of AdaBoost in which the weights of the examples are kept bounded by its initial value, however the authors admit no significant difference between AdaBoost and its modification in experiments on noisy data. Friedman [10] suggests that shrinking the weights of base hypothesis would increase boosting generalization capability.

The property of concentrating on few hard-to-learn patterns can be interpreted in terms of margin maximization; this view leads to regularized modifications. Ratsch et al. [21] view boosting as minimization of cost functional through an approximate gradient descent with respect to a margin. The authors propose regularization methods to achieve "soft margins" for AdaBoost which should avoid boosting from concentrating on misclassified samples with large negative margins. Friedman [10] also considers regularization methods through introducing proportional shrinkage into gradient boosting.

In contrast to regularization methods and weight shrinking methods, we do not penalize algorithm's behavior that can lead to overfitting and concentrate on removing samples that we prove to be harmful for boosting (Section 3.). We consider such approach more appropriate because it explicitly and strictly defines samples to be ignored, rather then penalize behavior that seems to lead to overfitting, but also may be the fitting of hard data.

Other authors see unboundness of loss function as the major source of overfitting. Viewing boosting as a gradient descent search for a good fit in function space allows modifying loss functions. Mason et al. [15] views boosting as gradient descent on an appropriate cost functional in a suitable inner product space. Proposed modification of boosting algorithm with normalized sigmoid cost function was reported to outperform AdaBoost when boosting decision stumps, particularly in the presence of label noise. Friedman [10] considers Huber loss function.

Rosset [23] proposes an approach of weight decay for observation weights which is equivalent to "robustifying" the underlying loss function. However the author admits that in experiments on real-world data there is no consistent winner between the non-decayed and the decayed versions of boosting algorithm.

In contrast to referenced methods, we see the main source of boosting overfitting not in an inappropriateness of particular loss function, but in general concept of average loss minimization (Section 3.). We show that this procedure is not adequate for tasks with overlapping classes'.

A large body of research addresses learning in the presence of noise. Robust statistics emerged in the 1960s in the statistical community [12]. However in classification tasks we cannot use the usual definition of robustness from statistics. In random classification noise model, the binary label of each example which the learner receives is independently inverted from the true label with fixed probability, which is referred to as the noise rate.

Krause & Singer [14] suggest employing EM algorithm and changing loss function to make boosting tolerant to known level of noise. Takenouchi & Eguchi [27] develop a modification of AdaBoost for classification noise case. The authors modify loss function and show that proposed method moderates the overweighting for outliers using a uniform weight distribution.

In contrast to the works described above we do not assume presence of classification noise and do not request any prior knowledge about data such as noise level. It should be noted that our method provides performance gain on real world data without adding artificial noise, while most methods show performance gain only if noise is present, and several even note degraded performance if no artificial noise added.

Previous research demonstrates that removing hard examples is worthwhile [17][16][1]. The main goal of these approaches is to enhance the classification accuracy by improving the quality of training data. These methods have various mechanisms to identify examples "suspicious, surprising or close to the boundary" [1]. In most works the decision as to which of the examples should be excluded is based on observations of algorithm behavior. In [17] analysis of dynamical evolution of AdaBoost weights is performed for estimating 'hardness' of every training example. Then the points with hardness above certain threshold, which is a parameter of the algorithm, are removed from the training set. In contrast to referenced works we explicitly and strictly define samples to be removed and propose non-parametric algorithm for removing these samples.

Our underlying concept resembles comments provided by Domingos [5] who states that increasing margin (lowering variance) for some samples can actually be harmful for the learner.

In this paper we study the reasons of overfitting of boosting in case of noiseless data with overlapping classes. In this case both samples $(x,+1)$ and $(x,-1)$ (considering binary classification task) can occur with positive probabilities. We show that minimization of average loss over the training set, which is used by all boosting-like algorithms, is not adequate in case of overlapping classes. We call training samples that have conditional probability of their own label lower than of the opposite "confusing samples". Forcing classifier to fit confusing samples by minimizing average loss over the training set can lead to non-optimal solutions. We view this as one of the main reasons of boosting overfitting on real-world data.

Removing confusing samples from the dataset leads to a deterministic task with the same Bayesian separating surface. This finding suggests that by removing these samples from training set we can enhance boosting capabilities and avoid overfitting.

Described below is the algorithm for removing confusing samples, which requires no prior information about data. We also show how generalization error can be predicted from the training set only, by estimating the amount of confusing samples. In order to support our conclusions we perform experiments on both synthetic and real world data from UCI-repository. Experiments confirm that removing confusing samples helps avoiding overfitting and increasing the accuracy of classification. The error prediction is also experimentally confirmed to be quite accurate.

Other sections of this paper are organized as follows. In section 3. we present reasoning explaining boosting inadequateness in case of overlapping classes. In section 4. we describe an algorithm for removing confusing samples from training set. Section 5 describes our experiments and section 6. is left for conclusion.

3. Average loss and confusing samples

Let $T = (x_i, y_i), i = 1,...,n$ be the training set, where $x_i \in X$ is the vector of attributes and $y_i \in \{-1,+1\}$ is the class label (for simplicity we consider binary classification task). We take the assumption, that the pairs *(x,y)* are random variables distributed according to an unknown distribution $P(x, y)$.

We consider the general case of overlapping classes, which means that for some instances x the probability of both labels is positive.

$$\exists x : p(x,+1) > 0,\ p(x,-1) > 0$$

Definintion 1. Let us call $\{(x_i, y_i) \in T : P(-y_i \mid x_i) > 0.5 > P(y_i \mid x_i)\}$ "confusing samples". Samples $\{(x_i, y_i) \in T : P(-y_i \mid x_i) < 0.5 < P(y_i \mid x_i)\}$ will be called "regular samples".

Lemma 1. The fraction of confusing samples in the training set converges (in probability) to Bayesian rate with training set size increasing indefinitely.

Proof. Let us denote Bayesian rule by $B(x)$. The exposition immediately follows from classical Bernoulli theorem and the fact that confusing samples are those samples, which are misclassified by the perfect Bayesian classifier:

$$P(-y_i \mid x_i) > 0.5 > P(y_i \mid x_i) \Leftrightarrow B(x_i) = -y_i .$$

Lemma 1 says that in case of overlapping classes training set $T = (x_i, y_i), i = 1,...,n$ contains a mixture of regular and confusing samples; the fraction of confusing samples in the training set is governed by the value of Bayesian rate. This lemma provides us with error prediction algorithm, which will be described in section 4.1 .

Lemma 2. Removing all confusing samples from the training set reduces overlapping classes' task to a deterministic classification task with the same Bayesian separating surface.

Proof. Removing confusing samples leads to a deterministic classification task with conditional class distribution

$$\tilde{P}(y \mid x) = \begin{cases} 1, & P(-y \mid x) < 0.5 < P(y \mid x) \\ 0, & P(-y \mid x) \geq 0.5 \geq P(y \mid x) \end{cases}.$$

One can see that Bayesian rule for this derived task and original task are the same, which proves the lemma.

In standard boosting algorithms, training set is sequentially reweighed and fitted by weak learner in order to minimize average loss $C:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ on the training set:

$$\frac{1}{n}\sum_{i=1}^{n} C(y_i, F(x_i)) \to \min_F,$$

where $F(x)$ is the current classifiers ensemble. In case of probabilistic setup, one seeks to minimize the conditional expectation of loss for all instances *x* [9]

$$E\left(C\left(y, F(x)\right) | x\right) = C\left(1, F(x)\right) P(1 | x) + C\left(-1, F(x)\right) P(-1 | x)$$

Consider overlapping classes' task. Let *n* stand for a number of samples with feature vector *x* in a given training set

$$\frac{1}{n}\sum_{i=1}^{n} C(y_i, F(x_i)) = \sum_{x\in T}\left(\frac{n_x}{n}\times\frac{1}{n_x}\sum_{x_i=x} C\left(y_i, F(x)\right)\right) =$$

$$= \sum_{x\in T}\frac{n_x}{n}\times\left(\frac{1}{n_x}\cdot\sum_{x_i=x, y_i=+1} C\left(+1, F(x)\right) + \frac{1}{n_x}\cdot\sum_{x_i=x, y_i=-1} C\left(-1, F(x)\right)\right) =$$

$$= \sum_{x\in T}\frac{n_x}{n}\times\left(\frac{n_{x,+1}}{n_x}\cdot C\left(+1, F(x)\right) + \frac{n_{x,-1}}{n_x}\cdot C\left(-1, F(x)\right)\right),$$

where $n_{x,+1}, n_{x,-1}$ respectively denote amount of samples $(x,+1)$ and $(x,-1)$.

Consider the score of a fixed instance x from T in average loss:

$$Score(x) = \frac{n_{x,+1}}{n_x}\cdot C\left(+1, F(x)\right) + \frac{n_{x,-1}}{n_x}\cdot C\left(-1, F(x)\right).$$

One can see that the score of every instance *x* from *T* in average loss converges to the conditional loss expectation with indefinitely increasing number of copies of instance *x* in training set. Training set can contain several copies of an instance only in case of very large and dense training set, that almost never holds in practice.

Usually $n_x = 1$, and $Score(x) = C\left(y_i, F(x)\right)$. In this case minimizing the score of a confusing sample actually increases the true expectation of loss for the fixed instance. Broadly speaking, minimizing average loss means forcing classifier to fit all training samples, including the confusing samples, while correct classification of confusing samples is undesirable. Removing confusing samples reduces classification task with overlapping classes to deterministic, which makes PAC learning framework (in which boosting is formulated) directly applicable.

In this paper we rely on Lemma 2 and reduce task with overlapping classes' to a deterministic one with the same Bayesian separating surface. We propose an algorithm that removes confusing samples and experimentally study the performance of boosted stumps being trained on reduces training set.

4. Algorithm

Our goal is to roughly estimate the conditional probabilities of training set samples labels and to exclude those samples, for which

$$P(-y_i \mid x_i) > 0.5 > P(y_i \mid x_i).$$

In [19] Platt scaling is suggested to obtain calibrated probabilities from boosting. Platt's scaling approximates posterior as:

$$P(y \mid x) \approx \frac{1}{1+\exp(A \cdot F(x) + B)}$$

We noticed that Platt scaling may be unstable in case of unbalanced data, in this case simple logistic transform can be used [9]. We build an iterative process. During iterations we randomly divide data into 3 parts. We train a boosted committee on the first part, calibrate probabilities on the second and estimate posterior for samples labels in the third part (using the built classifier and estimated calibration parameters). We repeat this procedure several times. At each step we acquire an estimate of class probabilities for training instances. Posterior estimates for every sample are averaged.

Algorithm: Removing "confusing samples"

Input: A training set $T = \{(x_i, y_i)\}_{i=1}^{n}$; number of epochs K;

1. **for** k=1 to K
2. Divide a training set into three subsets $\bigcup_{i=1}^{3} T_k^i = T$ and $\bigcap_{i=1}^{3} T_k^i = \varnothing$
3. Train boosted weak learners on T_k^1 and obtain classifier $F^k(x)$
4. Estimate calibration parameters A and B for posterior output on T_k^2 using Platt scaling (or logistic transform)
5. Estimate posterior probability of labels for samples in T_k^3,

$$p^k(y \mid x_i) \approx \frac{1}{1+\exp\left(A \cdot F^k(x_i) + B\right)}$$

6. **end for**
7. Calculate the average of posterior probability:

$$p(y \mid x_i) = \underset{k:x_i \in T_k^3}{mean}\left(p^k(y \mid x_i)\right)$$

8. Construct reduced training set T' from those samples, for which

$$\arg\max_y \{p(y \mid x_i)\} = y_i$$

9. return T'

Those samples that have average posterior probability estimate of their label lower than of its opposite are considered to be "confusing samples". After removing "confusing samples" the reduced training set can be learned by boosted committee.

Averaging of estimates was extensively studied in context of regression tasks [28]. Perrone [20] proved that averaging estimates always generates improved estimate in the sense of any convex optimization measure even without independence assumption on the estimates. Brieman [3] experimented with averaging class probability estimates obtained by learning decision trees on bootstrap replicates of the training set. He showed that averaging decreases error of class probability estimation.

Proposed algorithm is in relation with data editing approach [13] [24] [17]. Data editing methods are developed for improving the Nearest Neighbor classification accuracy by removing (or relabelling) outliers from the training set. In [13] [24] ensembles of neural networks, built by bagging, are employed to identify outliers. In contrast, we use calibrated boosted trees, which provide better posterior estimation [19].

4.1 Error estimation

Since we are reducing classification task with overlapping classes to deterministic task, then the percent of detected confusing samples should, according to Lemma 1, be the prediction of error. Our experiments, described below, confirm this.

5. Experiments

In order to study the behavior of Boosting incorporated with removing of confusing samples we have conducted a set of experiments on both synthetic and real world data. We compare the performance of boosted stumps trained on full training set with boosted stumps trained on reduced dataset. We use Boosting algorithm described by Schapire, R., & Singer, Y. [25]. Stumps were chosen as base learners to avoid possible issues with base learner complexity [22]. We used Platt scaling for posterior approximation if data is balanced and logistic transform otherwise. Size of is T_k^1 15% of overall training data.

5.1 Synthetic data

We used two overlapping Gaussians to check our conclusions. Each Gaussian has standard deviation σ=1 and centers at *(1,0)* and *(-1,0)*.A perfect, Bayesian classifier would be a straight line coinciding with the second coordinate axis (a stump). We take 10000 random samples drawn from these Gaussians as a test set and randomly construct a training set of 200, 500 and 1000 samples. In each experiment the test set was fixed and the training set was randomly drawn from distribution. This was repeated for 100 times and the results were averaged. We measured the quality of pruning as the precision of detecting confusing and regular samples in the training set. We could do it explicitly since the Bayesian classifier is known. Precision is defined as

$$Pn = \frac{Tp}{Tp + Fp},$$

where *Tp* is the number of correct detections (of confusing or regular samples) and the *Fp* is the number of false positive detections. It is the ratio of correctly detected confusing or regular samples in the full set.

Figure 1 illustrates performance on our toy example. AdaBoost applied to full set overfitts greatly and produces cluttered separating surface. Removing confusing samples allows AdaBoost to produce smooth boundary, very close to Bayesian.

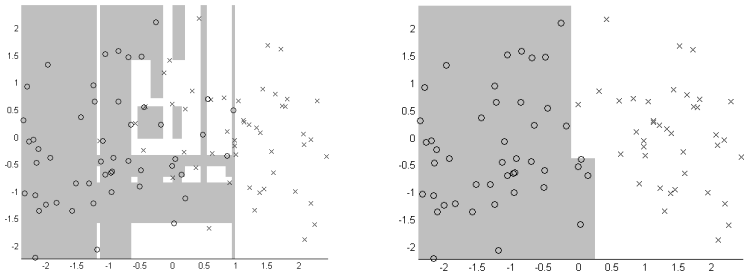

Fig. 1. Artificial data points and separating surfaces. From left to right: full dataset; dataset reduced by proposed algorithm.

Table 1 lists the results. It is clear that AdaBoost does overfit on confusing samples, especially in case of modest training set size. The estimated error comes to be quite consistent with actual error on the test set while is somewhat higher then error of the Bayesian classifier. Accuracy of error prediction is also confirmed by experiments on real world data presented in the next section. Looking at precision of pruning we can see that confusing samples precision is significantly lower then the precision of regular samples detection. This means that some percentage (10-15%) of removed samples is actually regular, while most (~98%) of samples marked as regular are marked correctly. It seems that losing some regular samples does not really degrade the performance of boosting, while removing the majority of confusing ones helps noticeably.

Table 1. Test error (%) of AdaBoost trained on raw and reduced data, error estimation and pruning precision on synthetic data.

Set Size	Estimated error	10 iterations		100 iterations		Bayesian classifier error		Regular samples precision	Confusing samples precision
		Full	Reduced	Full	Reduced	Train	Test		
200	16.48	17.38	**16.45**	19.72	**16.55**	15.49	16.07	97.93	89.26
500	16.39	16.13	**15.98**	17.38	**16.01**	15.70	15.66	98.05	86.04
1000	16.39	**16.14**	16.33	16.89	**16.37**	15.89	15.95	97.67	85.34

The more data we have, the closer is average loss to its expectation. Also, the less iterations boosting does, the smaller is the effect of overfitting. Thus, with a lot of data and after only few iterations AdaBoost should not noticeably overfit, while AdaBoost trained on reduced data may start suffering form underfitting, because of smaller training set. This happened in an experiment with training set containing 1000 samples and 10 iterations of boosting.

5.2 Measuring the quality of pruning on real world data

In case of synthetic data one can explicitly measure the error of confusing samples detection, but in case of real world data this is impossible. The important issue here is how to devise an appropriate metric for the quality of pruning algorithm for real world data. Consider dividing dataset into two parts and separately pruning both of them. If pruning is done correctly, a classifier trained on first reduced part is expected to misclassify samples marked as confusing in the other part and vice-versa. Thus, we propose to measure the precision of detecting regular and confusing samples by the classifier trained on the separate, reduced part of the same set.

The precision of regular samples detection is the ratio of samples that were marked as regular samples by pruning algorithm that were correctly classified by classifier trained on the separate, reduced subset. Analogously, precision of confusing samples detection is the ratio of samples marked as confusing in the set of samples that were misclassified by classifier trained on another separate, reduced subset.

Table 2. Test error (%) of AdaBoost trained on raw and reduced data; test error of MadaBoost and error estimation and pruning precision on various data sets.

Dataset	Esti-mated	100 iterations			1000 iterations			Regular samples precision	Conf. samples precision
		Full	Reduced	Mada	Full	Reduced	Mada		
BREAST	3.73	4.6±0.08	**3.65±0.09**	4.01±0.09	4.77±0.1	**3.67±0.09**	4.15±0.22	98.83	72.91
AUSTRALIAN	13.06	15.2±0.17	**13.8±0.17**	14.93±0.23	17.68±0.17	**13.88±0.16**	16.23±0.64	96.78	74.43
GERMAN	24.66	25.72±0.16	25.35±0.14	**23.4±0.38**	28.4±0.18	25.05±0.16	**24.9±0.03**	91.22	71.54
HABERMAN	25.96	29.67±0.29	**26.33±0.31**	27.78±0.31	34.50±0.36	**26.37±0.32**	34.97±0.1	92.14	77.14
HEART	18.34	21.41±0.36	18.36±0.30	**16.66±0.58**	23.13±0.36	**18.07±0.30**	20.37±0.35	92.60	68.26
PIMA	24.03	25.58±0.20	**23.99±0.16**	25.26±0.17	28.07±0.20	**24.10±0.17**	28.26±0.12	93.38	79.29
SPAM	5.79	6.19±0.04	6.02±0.04	**5.59±0.03**	6.35±0.04	**5.97±0.04**	6.26±0.03	98.83	78.01
TIC-TAC-TOE	6.49	**8.47±0.20**	13.59±0.29	12.84±0.03	2.04±0.05	2.12±0.08	**1.67±0.07**	97.83	35.70
VOTE	4.51	4.75±0.13	**4.61±0.1**	5.51±0.43	5.90±0.14	**4.63±0.10**	7.35±0.29	99.51	88.84

5.3 UCI-repository data

In order to measure the performance of our algorithm on real world data we have selected 9 datasets from UCI-repository [2]. In our experiments we split the dataset in two subsets of equal size and use one subset for training and the other for testing and vice-versa. We measured the test error, pruning precision (as described above) and the error prediction from the training set (as described in section 2.3). This procedure was repeated 50 times for a total of 100 runs (50x2 cross-validation). We had to use equally sized training and test set to be able to measure the quality of pruning. We

also compared our approuch with one of the regularized boosting methods, namely MadaBoost [6].

Table 2 presents the results of our experiments, the lowest error is shown in bold. AdaBoost trained on reduced dataset has lowest test error on 5 of 9 datasets if ran for 100 iterations and is best on 7 of 9 if ran for 1000 iterations. It performs better than AdaBoost trained on full set on all, but Tic-Tac-Toe dataset. The failure on Tic-tac-toe dataset is actually anticipated, because the data is perfectly separable and the Bayesian error is zero. As the number of learning iteration increases, AdaBoost trained on full dataset tends to overfit on most datasets, while AdaBoost trained on reduced data has almost non-increasing error very close to the predicted estimate. Moreover, MadaBoost is also prone to overfitting when ran for 1000 iterations despite regularization. This confirms our conclusions that the source of boosting overfitting are the confusing samples, and that for most real world data class overlapping is present.

Figure 2 provides test curves for two datasets. On Breast dataset AdaBoost does not have any significant gain in error with the increase of training iterations, what was also noted before [22], but it still benefits from pruning.

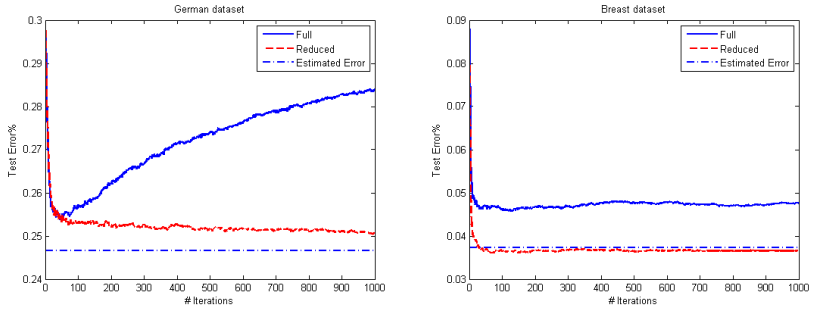

Fig. 2. Test and training error curves. From top left to bottom right: test error on German dataset; test error on Breast dataset.

5.4 Margins

It is common to interpret the performance of Boosting in terms of margins maximization. Figure 3 shows the cumulative margin for AdaBoost trained on full and reduced Breast and Tic-Tac-Toe datasets after 100 rounds of training. On Breast dataset AdaBoost trained on reduced dataset has lower minimal margin but has uniformly higher margin for the margins higher than a small threshold (0.03). Low minimal margin is a direct consequence of pruning – it is natural to expect negative margins for the removed samples. Pruning makes the data simpler and allows AdaBoost to maximize the margins of regular samples more efficiently. The behavior of margins on Tic-Tac-Toe is the same, but in case of Tic-Tac-Toe, gain in margins on lower cumulative frequencies is insignificant. Thus sacrificing minimal margin does not actually give any benefit and only worsens the performance.

As pointed out by Reyzin & Schapire [22], margin theory would suggest sacrificing minimal margin for the higher margin at low cumulative frequencies. Removing confusing samples seems to make AdaBoost perform in such manner.

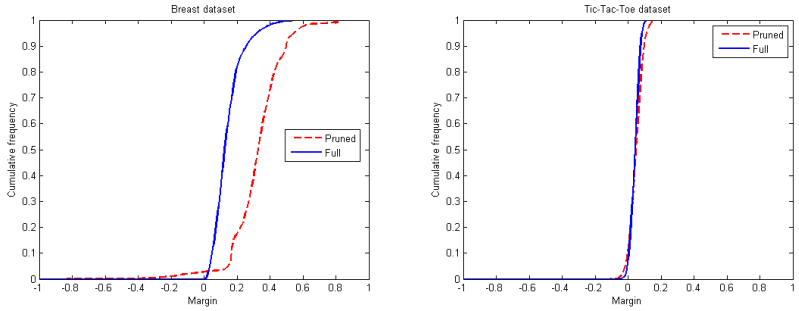

Fig. 3. Cumulative margins for AdaBoost trained on full and reduced dataset after 100 rounds. From left to right: Breast dataset; Tic-Tac-Toe dataset.

6. Conclusion

We described the problem of overfitting in boosting in case of overlapping classes. In this case, it is likely that overfitting is induced by fitting so called "confusing samples", that are samples misclassified by "perfect" Bayesian classifier. Overfitting in boosting seems to occur only when target distributions overlap or the noise is present, thus boosting could show no overfitting on some datasets and overfit greatly on others. An algorithm for removing the confusing samples is described, which is experimentally confirmed to help boosting get lower test error and avoid overfitting. We also show that by pruning confusing samples one can effectively predict the generalization error of the classifier by analyzing only the training set. We prove our conclusion by the experiments on both synthetic data and nine UCI-repository datasets.

Acknowledgments. Authors would like to thank Dr. D. Vetrov, D. Kropotov and Dr. V. Vezhnevets for inspiring discussions; and E. Vezhnevets for proof reading.

References

1. Angelova, A., Abu-Mostafa, Y., Perona, P.: Pruning Training Sets for Learning of Object Categories, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2005)
2. Blake, C. L., & Merz, C. J.: UCI repository of machine learning databases (1998)
3. Breiman, L.: Bagging Predictors. Machine Learning, 24, 2, (1996) 123-140
4. Dietterich, T., G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 40 (2) (1999)
5. Domingos, P.: A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In Proc. of the 17th National Conference on Artificial Intelligence. (2000).
6. Domingo C., Watanabe O.: Madaboost: A modication of adaboost. In 13th Annual Conference on Comp. Learning Theory, (2000)
7. Freund, Y., & Schapire, R.: Discussion of the paper ``Additive logistic regression: a statistical view of boosting" by J. Friedman, T. Hastie and R. Tibshirani. The Annals of Statistics, 38, 2, (2000) 391-393
8. Freund, Y.: An Adaptive Version of the Boost by Majority Algorithm. Machine Learning, 43(3) (2001) 293-318

9. Friedman, J., Hastie, T., & Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. The Annals of Statistics, 28, 2, (2000) 337-407
10. Friedman, J.: Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29, 5. (2001)
11. Grove, A.J., & Schuurmans, D.: Boosting in the limit: Maximizing the margin of learned ensembles. In Proceedings of the Fifteenth National Conference on Artifical Intelligence (1998)
12. Hampel, F. R., Rousseeuw, P. J., Ronchetti, E. M. and Stahel, W. A.: Robust Statistics: the Approach Based on Influence Functions. Wiley, New York. (1986)
13. Jiang and Zhou. Editing training data for kNN classifiers with neural network ensemble. LNCS, (2004).
14. Krause N, Singer Y.: Leveraging the Margin More Carefully. ACM International Conference Proceeding Series; Vol. 69 (2004)
15. Mason L, Baxter J, Bartlett P, Frean M.: Boosting algorithms as gradient descent Neural Information Processing Systems 12, MIT Press, (2000) 512-518
16. Merler, S., Caprile, B., Furlanello, C.: Bias-variance control via hard points shaving, International Journal of PatternRecognition and Artificial Intelligence, (2004)
17. Muhlenbach, F., Lallich, S., Zighed, D. A.: Identifying and handling mislabelled instances. Intelligent Information Systems, 22, 1. (2004). 89-109
18. Nicholson, A.:Generalization Error Estimates and Training Data Valuation, Ph.D. Thesis, California Institute of Technology, (2002)
19. Niculescu-Mizil, A., Caruana, R.:Obtaining Calibrated Probabilities from Boosting Proc. 21st Conference on Uncertainty in Artificial Intelligence (2005)
20. Perrone M.: Improving regression estimation: Averaging methods for Variance reduction with extension to General Convex Measure Optimization, Ph.D. Thesis, Brown University, (1993)
21. Ratsch. G.: Robust Boosting and Convex Optimization. Doctoral dissertation, University of Potsdam, (2001)
22. Reyzin, L., & Schapire, R.: How boosting the margin can also boost classifier complexity. In Proceedings of the 23rd International Conference on Machine Learning (2006)
23. Rosset S.: Robust Boosting and Its Relation to Bagging. KDD-05 (2005)
24. Sanchez et al. Analysis of new techniques to obtain quality training sets. Pattern Recognition Letters, (2003)
25. Schapire, R., & Singer, Y.: Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning, 37, 3, (1999) 297—336
26. Schapire, R., Freund, Y., Bartlett, P., and Wee Sun Lee.: Boosting the margin: A new explanation for the effectiveness of voting methods. In Machine Learning: Proceedings of the Fourteenth International Conference (1997)
27. Takenouchi T., Eguchi S.: Robustifying AdaBoost by adding the naive error rate. Neural Computation 16 (2004)
28. Taniguchi M, Tresp V.: Averaging Regularized Estimators, Neural Computation (1997)