Study on Parameter Selection Using SampleBoost

Mohamed Abouelenien and Xiaohui Yuan
{*mohamed, xiaohui.yuan*}*@unt.edu*
Computer Science and Engineering Department,
University of North Texas, Denton, Texas, USA

Abstract

SampleBoost is an intelligent multi-class boosting algorithm that employs an error parameter combined with stratified sampling during training iterations to accommodate multi-class data sets and avoid problems associated with traditional boosting methods. In this paper we investigate the choice of the error parameter along with the preferred sampling sizes for our method. Experimental results show that lower values of the error parameter can lower the performance while larger values lead to satisfactory results. The parameter choice has noticeable effect on low sampling sizes and has less effect on data sets with low number of classes. Varying sampling sizes during training iterations achieves the least variance in the error rates. The results also show the improved performance of SampleBoost compared to other methods.

Introduction

Ensemble Learning has gained a lot of attention owing to its improved performance compared to single classifiers. AdaBoost (Freund and Schapire 1997) represents one of the most successful ensemble learning methods. The idea is to iteratively combine decisions of different weak classifiers into an improved one. However, AdaBoost faces several challenges when it is employed for real-world problems, among which early termination can turn the whole ensemble learning process into a single classifier (Abouelenien and Yuan 2012). This issue is caused by the repetition of misclassified examples which rapidly increase the weighted error to the maximum error bound. AdaBoost additionally suffers in finding a direct transformation to accommodate multi-class data sets and requires extended training time.

AdaBoost.M1 (Freund and Schapire 1997) was introduced as an initial attempt to address multi-class problem, which was followed by two updated methods AdaBoost.M2 (Freund and Schapire 1997) and AdaBoost.MH (Schapire and Singer 1999). They converted the multi-class classification problem into a sequence of binary classification problems. SAMME (Zhu et al. 2009) method was developed to accommodate any number of classes by relaxing the error bound. The method eased the error bound to avoid early termination that is caused by increased weighted error. Random downsampling was integrated with AdaBoost mainly to learn from imbalanced data sets as well as to address efficiency issue with large data set. The main purpose of sampling was to even the skewness of the learning process towards majority classes. Geiler et al. (Geiler, Hong, and Yue-jian 2010) and Seiffert et al. (Seiffert et al. 2010) introduced random sampling of majority classes in E-Adsampling and RusBoost algorithms, respectively, to improve learning from minority classes. Lenth (Lenth 2001) specified practical guidelines to decide a sample size effectively. However, methods suggesting sampling sizes depend on the existence of prior information about the collected data. Therefore, determining a general sampling size rule seems infeasible. However, investigating the best combination between the sampling size and the error parameter might shed some light on determining the optimal parameters for our method.

Our SampleBoost (Abouelenien and Yuan 2012) method was developed to address the aforementioned challenges. In this paper we study the SampleBoost method in search for the optimal settings for the parameters involved. SampleBoost downsamples the multi-class training set at each iteration using a class-based weighted sampling. It employs a weighted stratified sampling technique where each class represents a stratum. Stratified sampling divides the examples into groups referred to as strata and then applies random sampling to each individual group (stratum). This scheme assigns higher chances of selecting hard-to-classify examples for the next training iteration for each class independently. The method also introduces an error parameter that is added to the loss function of the boosting method. The parameter eases the error condition to accommodate multi-class classification. Along with the sampling scheme, SampleBoost avoids early termination and aims at improving both efficiency and accuracy.

Boosting with Class-based Stratified Sampling

Improvement achieved using AdaBoost and its multi-class extensions rely on the variation in the decisions provided by individual weak classifiers. However, no improvement is achieved if all the classifiers created the same decision boundary. Once the same group of examples is repeatedly misclassified, the algorithm terminates regardless of the

weak classifier accuracy (Abouelenien and Yuan 2012).

SampleBoost is a multi-class boosting method that employs a class-based stratified sampling. In a training iteration, a subset of examples identified using downsampling is used. Yet, each classifier is evaluated using the whole training set. Based on this evaluation, the weights for all examples are updated such that the weights for the misclassified ones are increased. The weighted sampling process then selects examples for the next iteration according to the updated data distribution. Based on the evaluation process, a weight is assigned to the classifier to decide its contribution to the overall decision. In our SampleBoost method, we introduced an error parameter γ in the calculation of the classifier weights (see Equation (2)) to accommodate multi-class data sets and ensure diversity among classifiers. This adjustment is translated to the loss function and accordingly to the weighted error in Equation (1). Upon completion of ensemble training, the decisions of all weak classifiers are combined using weighted voting.

SampleBoost has two parameters that need to be specified in advance in addition to the total number of iterations. The first is the error parameter and the second is the sampling size for each class. In AdaBoost.M1, the error bound was set to 0.5 which is a very strict condition to learn from multi-class data sets. In SAMME, the loss function was modified by including a constant term $K-1$ that accounts for the error rate of random guessing of K classes, i.e., $1-\frac{1}{K}$. We investigate whether the range between those two error bounds can accommodate multi-class data sets. Moreover, it was never determined if the error bound is limited to SAMME's $1-\frac{1}{K}$. We extended our error parameter to $10K$ to observe its ability of accommodating multi-class data sets.

We additionally investigate different sampling sizes. The sizes can be fixed or varying through different iterations. Our gaol here is not to determine the best sampling size since the data is assumed to exist from an unknown probability distribution. Yet, we provide guidelines for choosing the suitable sampling size relative to the size and number of classes of the data set. Moreover, we explore if there exists a preferred combination of the error parameter and the sample size for improved performance.

Given a training set $TR = (x_1, y_1), \ldots, (x_N, y_N)$. $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{1, \ldots, K\}$. K is the number of classes. N is the total number of examples and T is the total number of iterations. "s" is the downsampled size per class to form a total sample size of $S = s * K$. $[\![\cdot]\!]$ denotes an indicator function that returns 1 if true and -1 otherwise. $\mathcal{J}[.]$ is an indicator function that returns 1 if true and 0 otherwise.

Algorithm 1 SampleBoost

initialize example weights with $w_1(i) = 1/N$
for $t = 1, \ldots, T$ **do**
  Select a subset $S \subset TR$, $S = s * K$, according to the distribution
  Train a weak classifier f_t with S

$$\epsilon_t = \sum_{i=1}^{N} w_t(i) \mathcal{J}[y_i \neq f_t(x_i)] \quad (1)$$

  if $\epsilon_t > \frac{\gamma}{\gamma+1}$ **then**
    return $\alpha_t = 0$
  else
    Calculate α_t of the weak classifier

$$\alpha_t = \log(\frac{1-\epsilon_t}{\epsilon_t}) + \log(\gamma) \quad (2)$$

    Adjust the examples distribution to,

$$w_t \Leftarrow w_t e^{\alpha_t [\![y_i \neq f_t(x_i)]\!]}$$

    Normalize w_t
  end if
end for
Combine weak classifiers f_t into $F(x)$

$$F(x) = \arg\max_y \sum_{t=1}^{T} \alpha_t f_t(x)$$

Experimental Results and Discussion

Five data sets with varying sizes, number of classes, and dimensionality were trained for our experiments. We created a 2D Gaussian synthetic data set. Image Segmentation and Letter Recognition data sets were downloaded from UCI machine learning repository (Frank and Asuncion 2010). AR (Martinez and Benavente 1998) and Yale (Georghiades, Belhumeur, and Kriegman 2001) face recognition databases were collected. Details and characteristics of each data set can be seen in Table 1. We trained the data sets using SampleBoost, AdaBoost.M1, and SAMME for 100 iterations in all experiments. Two fold cross validation was used to evaluate all data sets except for AR database where leave-one-out cross validation was used. Decision Trees were used as the weak classifiers owing to its recognition with boosting as the best off-the-shelf classifier (Breiman 1998). We used four different sampling sizes for SampleBoost as shown in Table 1. Three of the sizes, SB=A, SB-B, and SB-C, are fixed and the last one, SB-V, changed though different iteration. In particular, every 25 iterations, an increased sample size is used in training new iterations as shown in the table.

Principal Component Analysis was used for dimensionality reduction for AR and Yale face databases. The error parameter γ ranged from 1 as in AdaBoost.M1 up to ten times K including SAMME's $K-1$. Average Error Rates (AER) denoted with $\bar{E}$, average efficiency across all γ values in seconds, variance (σ^2) of the AER across different values of γ, and average number of effective weak classifier (EWC) are used to measure the performance of all experiments. σ^2 for each data set is calculated according to

$$\sigma^2 = \frac{1}{\mathcal{G}} \sum_{i=1}^{\mathcal{G}} (\bar{E}_i - \mu)^2 \quad (3)$$

where μ denotes the mean of all $\bar{E}\%$ and $\mathcal{G}$ denotes the total number of γ values used. The lower the variance, the more stable the method is across different γ. EWC presents the

average number of weak classifiers that did not exceed the error bound, i.e., classifiers with non-zero weights α.

Table 1: No. of Classes (K), size of each class ($|K|$), no. of features (feat), and 4 different sampling sizes per class using SampleBoost (SB) for 6 data sets

Data set	K	$\|K\|$	feat	SB-A	SB-B	SB-C	SB-V
Synthetic	50	100	2	10	25	40	10,20,30,40
UCI Image	7	330	18	30	75	150	30,60,90,120
UCI Letter	26	700	16	100	200	300	75,150,225,300
Face AR	50	11	13200	2	5	8	2,4,6,8
Face Yale	38	64	896	8	18	28	4,12,20,28

In Figure 1, it can be seen for the 2D Gaussian synthetic data set that lower values of γ achieve low error rates specifically for the lowest sampling size SB-10. Extending γ in the loss function to ten times K, i.e., 500, did not show any significant change. However, SB-10 achieves the highest AER. SB-25 and the variable sampling size SB-V achieve the lowest AER. All sampling sizes achieve lower AER compared to AdaBoost.M1 and SAMME as shown below the figure. The figure also shows that lower values of γ achieve the lowest EWC. As the sampling size decreases the number of EWC increases. The variable sampling size presented by SB-V had the second highest EWC. The number of EWC for AdBoost.M1 and SAMME (2.5 and 1.5) is very low compared to SampleBoost for all data sets due to the early termination problem. Table 2 show that the variance of the AER of all sampling sizes is below one except for SB-10. As the sampling size increases, the efficiency decreases. SB-V achieves an average efficiency. Generally, SB has higher efficiency compared to AdaBoost.M1 and SAMME.

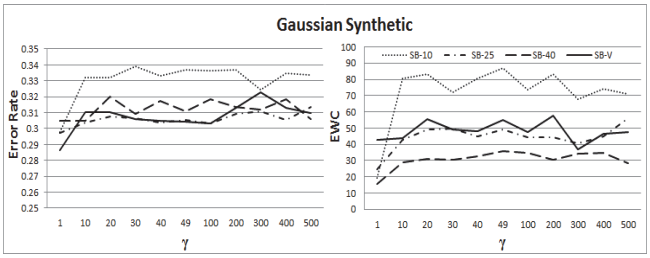

Figure 1: AER and EWC of 2D Gaussian Set using SampleBoost (*left and right*). AER and EWC of AdaBoost.M1 and SAMME are 0.34, 0.34 and 2.5, 1.5 respectively.

Figure 2 shows the results of the UCI Image Segmentation set. AER across different γs and up to ten times K, i.e., 70, shows no significant variation (see AER variance in Table 2). This is attributed to the small number of classes which is seen in γ values ranging from one to just 70. The highest sampling size SB-150 and SB-V achieved the lowest AER while SB-30 achieved the highest AER in most cases. EWC results also show that higher number of EWC is achieved with lower sample sizes. SB-V achieves the second highest EWC. Efficiency results in Table 2 show that smaller sample size has better efficiency.

UCI Letter Recognition results are shown in Figure 3.

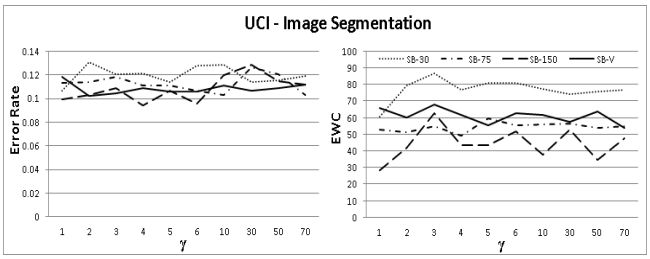

Figure 2: AER and EWC of Image Segmentation using SampleBoost (*left and right*). AER and EWC of AdaBoost.M1 and SAMME are 0.21, 0.213 and 2, 1 respectively.

Higher AER are achieved with low values of γ. The lowest sample size is the most affected by lower γ values. Extending γ to ten times K, i.e., 260, still achieve improved results than very low values of γ. The highest sampling size achieves the highest AER. The lowest AER is achieved by SB-V for $\gamma < K - 1$ and SB-200 for $\gamma \geq K - 1$. All sampling sizes outperform the AER of AdaBoost.M1 and SAMME, 0.405, 0.40, respectively. The variance of AER% in Table 2 shows also that a higher variance is achieved with SB-100 compared to other sizes. EWC results show that as the sampling size decreases the number of EWC increases. The increase in the number of EWC for SB-V when $\gamma < K - 1$ is reflected in the improvement of its AER. The very low number of EWC in SB-100 with $\gamma = 1$ is also reflected in its AER results.

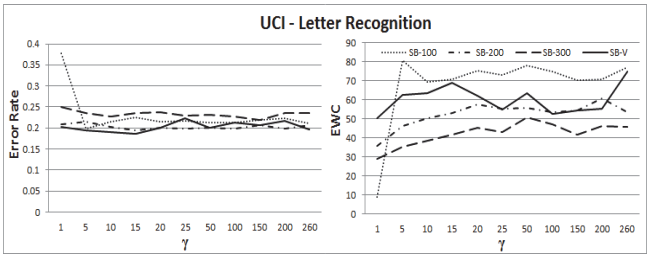

Figure 3: AER and EWC of Letter Recognition using SampleBoost (*left and right*). AER and EWC of AdaBoost.M1 and SAMME are 0.405, 0.40 and 2.5, 4 respectively.

Figure 4 presents results of face recognition AR database. Lower values of γ result in higher AER. The two lowest sampling sizes, SB-2 and SB-5, are the most affected by the lower values of γ. Extending γ to ten times K was successful for all methods. It is also reflected in the EWC results that lower values of γ result in very low number of effective classifiers. Larger sample sizes achieves better performance overall. This can be caused by the lower number of samples per class in this data set. SB-8 achieves better error improvement compared to AdaBoost.M1 and SAMME while SB-2 has higher AER. Variance results in Table 2 show that SB-V and SB-8 achieve the lowest AER variance.

Face recognition Yale results are shown in Figure 5. The trend of the results is very similar to that of the AR database. Lower values of γ result in lower number of EWC and, accordingly, higher AER. The lowest sampling sizes, SB-8 and

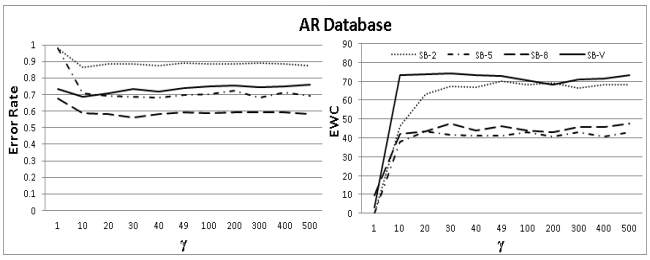

Figure 4: AER and EWC of AR database using SampleBoost (*left and right*). AER and EWC of AdaBoost.M1 and SAMME are 0.74, 0.74 and 1.2, 1.8 respectively.

SB-18, have significantly higher AER with lower values of γ. Extending γ to ten times K, i.e., 380, was successful for this set. All sample sizes outperformed AdaBoost.M1 and SAMME. Table 2 show that SB-8 and SB-18 have higher AER variance compared to SB-28 and SB-V.

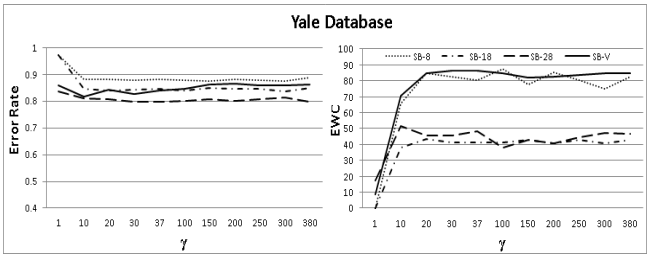

Figure 5: AER and EWC of Yala database using SampleBoost (*left and right*). AER and EWC of AdaBoost.M1 and SAMME are 0.88, 0.88 and 3, 2 respectively.

Table 2: Variance of $\bar{E}$ and Efficiency in seconds (E) for all SampleBoost sampling sizes and AdaBoost.M1 (A.M1) and SAMME (SA)

Dataset	SB-A		SB-B		SB-C		SB-V		A.M1	SA
	σ^2	E	σ^2	E	σ^2	E	σ^2	E	E	E
Syn	1.2	54.5	0.17	76.2	0.29	94.9	0.73	77.4	66.1	66.4
Image	0.53	8.77	0.51	14.7	1.07	21.5	0.19	14.2	14.9	14.7
Letter	22.4	200	0.27	334	0.54	447	1.16	310	393	397
AR	8.41	49.5	67.2	105	7.84	143	4.5	97.4	144	145
Yale	7.35	171	13.9	406	1.14	703	2.46	381	611	636

Conclusion

Our SampleBoost method employs weighted stratified sampling and integrates an error parameter to accommodate multi-class classification. The parameter relaxes the error bound for each base learner. In this paper, we shed some light on the choices of the error parameter with respect to sampling size.

Our experiments demonstrate that integrating low values of γ as in AdaBoost.M1 could result in weak classifiers that have zero contribution to the overall decision. This loss is also reflected in the sudden increase in the average error rate. However, for different sampling sizes, an increased number of effective weak classifiers has no implication on the improved generalization error. Lower sample sizes have deteriorated performance with lower values of γ. The problem is most prevalent when individual class in a data set has small number of examples. Some data sets might have a large total number of examples to train; however, what matters is the sufficiency of representative samples per class.

Data sets with small number of classes are less affected by the choice of γ. Extending γ to far beyond $K-1$ (as indicated in SAMME (Zhu et al. 2009)), e.g., ten times K, produced satisfactory results for all experiments, sampling sizes, and data sets. Our interpretation of this phenomenon is that the normalization factor also integrates γ in updating the data distribution. And this adjustment is directly reflected in the calculation of the weighted error.

Training a variable increasing sampling size has, on average, proven to be the least susceptible to large variance in the average error rates. It achieves improved efficiency compared to traditional boosting methods. However, when number of examples per class is small, using large sampling size is recommended to decrease the generalization error. We recommend training SampleBoost with a large fixed sampling size for small data sets and with variable sampling sizes otherwise. We also recommend using large γ value to avoid the uncertainty associated with lower values. With respect to accuracy, using large γ seems to always lead to satisfactory results.

References

Abouelenien, M., and Yuan, X. 2012. Sampleboost: Improving boosting performance by destabilizing weak learners based on weighted error analysis. In *2012 21st International Conference on Pattern Recognition (ICPR)*, 585–588.

Breiman, L. 1998. Arcing classifiers. *The Annals of Statistics* 26(3):801–824.

Frank, A., and Asuncion, A. 2010. UCI machine learning repository.

Freund, Y., and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(119–139).

Geiler, O. J.; Hong, L.; and Yue-jian, G. 2010. An adaptive sampling ensemble classifier for learning from imbalanced data sets. In *International MultiConference of Engineers and Computer Scientists*, volume 1.

Georghiades, A.; Belhumeur, P.; and Kriegman, D. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6):643–660.

Lenth, R. V. 2001. Some practical guidelines for effective sample size determination. *The American Statistician* 55(3):187–193.

Martinez, A. M., and Benavente, R. 1998. The AR face database. Technical Report 24, CVC Technical Report.

Schapire, R. E., and Singer, Y. 1999. Improved boosting algorithms using confidence-rated predictions. In *Machine Learning*, 80–91.

Seiffert, C.; Khoshgoftaar, T. M.; Hulse, J. V.; ; and Napolitano, A. 2010. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transaction on Systems, Man, and Cybernetics* 40(1).

Zhu, J.; Zou, H.; Rosset, S.; and Hastie, T. 2009. Multi-class adaboost. *Statistics and Interface* 2(3):349–360.