Jensen-Shannon Divergence in Ensembles of Concurrently-Trained Neural Networks

Aaron Mishtal and Itamar Arel
Machine Intelligence Lab
Department of EECS
University of Tennessee
amishtal@utk.edu, itamar@ieee.org

***Abstract*—Ensembles of neural networks have been the focus of extensive studies over the past two decades. Effectively encouraging diversity remains a key element in yielding improved performance from such ensembles. Negatively correlated learning (NCL) has emerged as a promising framework for concurrently training an ensemble of learners while emphasizing the cooperation among them. The NCL methodology relies on negatively correlating the errors of the learners as means of diversifying their outputs. In this paper, we extend this framework by employing the Jensen-Shannon divergence (JSD) - an information-theoretic measure of similarity between probability distributions - as a richer measure of diversity between learners. It is argued that for classification problems, utilizing the JSD is more appropriate than negatively correlating the errors. We analyze the new formulation and derive an upper bound on the parameter that balances accuracy and diversity among the learners. Simulation results applied to standardized benchmarks clearly demonstrate the advantages of the proposed method.**

I. Introduction

Neural networks have enjoyed renewed attention from the machine learning community in recent years, mainly due to their ability to deliver state-of-the-art performance results on various widely-studied benchmark challenges [4], [3]. In most cases, their success was attributed to a considerable advancement in computing capabilities introduced by graphics processing units (GPUs) [11], [3]. As a result, there is growing interest in investigating novel ways in which ensembles of concurrently-trained learners can further improve overall convergence rate and generalization properties. With an adequate balance between accuracy and diversity, an ensemble of learners can collectively deliver performance higher than that of any individual learner.

The majority of algorithms proposed in the literature consider a setting in which learners are trained independently and/or sequentially. For example, boosting [12] trains an ensemble sequentially, modifying the training set so that later learners learn to compensate for mistakes made by earlier learners. Bagging [1] is similar, but starts by creating a number of training sets. Each learner is then trained on a different training set independently of the rest of the ensemble. Substantial body of work has been presented proving the optimality of such schemes under reasonable assumptions, such as the boosting variation AdaBoost [12], [13]. However, with the introduction of cost-efficient massively parallel computing platforms, such as GPUs, there is great interest in studying architectures that can train ensembles of learners concurrently, rather than sequentially. Additionally, these methods are limited to situations where entire training sets are already available.

This paper introduces a method for training neural networks in parallel as an ensemble, whereby diversity between the different networks is guaranteed based on an information-theoretic measure. It is uniquely suited for classification problems, as can be appreciated intuitively and is demonstrated through evaluation on standard benchmarks. The proposed method offers enhanced performance relative to existing parallel-processing ensemble schemes, while retaining computational complexity that is comparable to the more simple formulations appearing in the literature. Moreover, we provide analysis for determining an upper-bound on the diversity parameter thus setting an appropriate balance between accuracy and diversity.

The rest of the paper is structured as follows. In Section 2 background information on existing work in ensemble methods is given. Section 3 introduces the proposed method, along with the relevant information-theoretic measures of diversity. Simulation results on standard benchmarks are discussed in Section 4, while Section 5 provides conclusions and some thoughts regarding future possibilities for this direction of study.

II. Background: Diversity in Ensembles of Neural Networks

It should be obvious that creating and ensemble using a group of identical learners is simply a waste of resources. If each member of the ensemble gives the same answer for each given input, then nothing is gained from combining them into an ensemble. This important observation has led to a wide variety of methods designed to create, exploit, and quantify a property of ensembles that has come to be called diversity [2]. While diversity is not quite a well-defined concept in this context, it generally refers to the ability of the ensemble members to produce different errors and implement different hypotheses. Intuitively, with each member implementing a different hypothesis, they will each learn a different aspect of the training data or become specialized on a particular subset of the input space. More diverse hypotheses should then correspond to learning more distinct aspects or subsets

978-0-7695-4913-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ICMLA.2012.198

of the input spacea. Combining several such hypotheses should create a much stronger hypothesis.

Some methods encourage diversity by modifying the datasets that each learner is trained on. Bagging is a technique that creates new training sets for each learner by drawing randomly and with replacement from the original training set. This creates new training sets that emphasize different examples and allows learners to be trained in parallel without any interaction between them. Boosting is a similar technique that trains ensemble members sequentially. This allows errors of earlier learners to be compensated for by later learners through modifications to the importance of each training example. This leads to highly competitive performance, but at the cost of increased training time due the sequential nature of the algorithm. There are countless variations and on these two methods that have been developed over the years [14], [7], [13].

With the increasing importance of online and very large datasets, methods that rely on modifications to the training set become less attractive. What is needed instead are methods that allow ensemble members to be trained in parallel on the same data. Such methods come naturally for neural networks by way of modifying the network cost functions to incorporate some interaction between the networks. One such method, called negative correlation learning (NCL) [9], [10], relies on the intuitive paradigm that one can enforce diversity between neural networks by augmenting their cost function such that it includes an explicit diversity term. As such, it naturally emphasizes the interaction and cooperation among the different learners. NCL employs the standard back-propagation (BP) algorithm to train the individual neural networks in parallel. In particular, negatively correlating the errors of the learners is considered as means of guaranteeing diversity.

Let a training set, D, be defined by

$$D = \{(x_1, t_1), ..., (x_N, t_N)\}, \tag{1}$$

where $x \in R^p$, is the p dimensional training pattern, d is the target output, and N the number of training patterns. In NCL, the output of the ensemble, $y(n)$, is assumed to be the average of the output of its M members, such that

$$y(n) = \frac{1}{M} \sum_{i=1}^{M} y^{(i)}(n), \tag{2}$$

where $y^{(i)}(n)$ is the output of learner i on the n^{th} training sample. The core idea behind NCL is to enforce diversity among the learners in addition to accuracy. When the n^{th} training pattern is presented, the i^{th} network is trained to minimize the following error function:

$$E_i(n) = e_i(n) + \lambda p_i(n), \tag{3}$$

where λ is a positive parameter determining the balance between the classification accuracy and the diversity term, $e_i(n)$ is the original cost function of learner i, and

$$p_i(n) = \left(y^{(i)}(n) - y(n)\right) \sum_{j \neq i}^{M} \left(y^{(j)}(n) - y(n)\right). \tag{4}$$

In this formulation, diversity is encouraged by explicitly forcing the networks to be negatively correlated. When $\lambda = 0$ one obtains a set of independently trained neural networks (i.e. simple ensemble of learners). This particular framework has been successfully applied to a broad range of applications, including classification, regression and time-series prediction. It has consistently demonstrated very competitive results with other popular ensemble learning techniques such as bagging, boosting, and evoluationary learning methods[9].

While negatively correlating the outputs of the individual learners makes perfect sense in the context of yielding diversified outputs, this formulation is sub-optimal when it comes to classification tasks. When considering such tasks, when presented with an input, each learner produces a posterior distribution (or an approximation to one) over the classes. Rather than negatively correlating the elements of these distributions, it would intuitively be more efficient if a distance measure between the various distributions was employed, which when maximized yields richer diversity between the learners. Fortunately, there have been several information-theoretic measures that express the divergence between two probability distributions.

III. Jensen-Shannon Divergence Method

Once such measure is the Jensen-Shannon divergence (JSD) [6], which is a symmetric variation of the popular Kullback-Liebler divergence (KLD) [8]. Both divergences are commonly utilized in information theory and statistics for measuring similarity between probability distributions. Although they measure similarity, or lack thereof, they are not metrics. Rather, the KLD gives the number of additional units of information required to encode samples from one probability distribution using samples from another. The KLD has some undesirable properties, namely that it is not symmetric and is unbounded. The JSD has been proposed, among other things, as a variation on KLD that is both symmetric and bounded.

For discrete probability distributions P and Q, the Kullback-Liebler divergence between P and Q is given by

$$D_{\text{KL}}(P||Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right). \tag{5}$$

The JSD is then defined as

$$\begin{aligned} D_{\text{JS}}(P||Q) \quad &= \quad \frac{1}{2}\sum_i P(i) \log\left(\frac{P(i)}{\frac{1}{2}(P(i)+Q(i))}\right) + \\ &\qquad \frac{1}{2}\sum_i Q(i) \log\left(\frac{Q(i)}{\frac{1}{2}(P(i)+Q(i))}\right). \end{aligned} \tag{6}$$

Hence, the JSD between two distributions is essentially the average of the KLD between each distribution and the mixture distribution. The value of the JSD between two distributions increases with increasing dissimilarity and is maximized when the probability mass of the each distribution is concentrated in differing elements. The maximal value the the JSD can take depends on the base of the logarithm used. Using the natural logarithm (base e) gives the maximum value of $\log(2)$, where $\log(x)$ is taken to be the natural logarithm, while using the base 2 logarithm gives 1 as the maximum value. As expected, the value of the JSD between two distributions decreases as the distributions become more similar. The JSD is always non-negative and is 0 when the distributions are identical. These properties, along with its differentiability, render JSD a primary candidate for encouraging diversity between ensembles of learners.

A. Jensen-Shannon Divergence and Neural Network Ensembles

In the context of classification, neural networks can be forced to produce a posterior distribution over the possible classes by applying the softmax activation function at the neural network outputs. Let x_k denote the weighted sum of inputs to the k^{th} output node of a neural network. The output of that node according to the softmax activation function is then

$$y_k = \frac{e^{x_k}}{\sum_k e^{x_k}}, \tag{7}$$

where the summation in the denominator is taken over all the output nodes in the network.

With the networks producing a probability distribution, the JSD ensemble method then consists of modifying the cost function of each network as follows

$$E_i = e_i - \lambda D_{\text{JS}}\Big(y^{(i)} || \bar{y}^{(i)}\Big) \tag{8}$$

where e_i is the original cost function of the i^{th} network in the ensemble, $y^{(i)}$ is its output vector, and $\bar{y}^{(i)} = \frac{1}{M-1}\sum_{j \neq i} y^{(j)}$ with M being the number of networks in the ensemble. The additional term is thus the JSD between the network's output and the average of the other networks' outputs. Subtracting this term from the original network cost function encourages the network to attempt to maximize its value. Since larger values of the JSD between two distributions reflect higher degree of dissimilarity, this term yields diversity in the ensemble by encouraging networks to produce different distributions for the same input. The parameter λ controls the strength of this term and the tradeoff between minimizing error (i.e. increasing accuracy) and maximizing diversity. Expanding the JSD term gives the following cost function for the i^{th} network in the ensemble.

$$\begin{aligned} E_i \;\; = \;\; & e_i - \frac{\lambda}{2}\left(\sum_k y_k^{(i)} \log\left(\frac{y_k^{(i)}}{\frac{1}{2}(y_k^{(i)} + \bar{y}_k^{(i)})}\right) + \right. \\ & \left. \sum_k \bar{y}_k^{(i)} \log\left(\frac{\bar{y}_k^{(i)}}{\frac{1}{2}(y_k^{(i)} + \bar{y}_k^{(i)})}\right)\right). \end{aligned} \tag{9}$$

We use the natural logarithm in the above equation to simplify future calculations. Differentiating this cost function with respect to the k^{th} network output yields

$$\frac{\partial E_i}{\partial y_k^{(i)}} = \frac{\partial e_i}{\partial y_k^{(i)}} - \frac{\lambda}{2}\log\left(\frac{y_k^{(i)}}{\frac{1}{2}(y_k^{(i)} + \bar{y}_k^{(i)})}\right), \tag{10}$$

which can then be backpropagated through the neural network to compute the gradients with respect to the network weights.

B. Upper Bound on Strength Parameter

Using the mean squared error cost function and an approximation to the natural logarithm, we analytically determine an upper bound for the strength parameter. Experimental evidence suggests that this bound is tight and applies to the case of the cross entropy cost function as well. Substituting in the mean squared error as the original cost function gives the function

$$\begin{aligned} E_i \;\; = \;\; & \frac{1}{2}\sum_k \left(y_k^{(i)} - t_k\right)^2 - \\ & \frac{\lambda}{2}\left(\sum_k y_k^{(i)} \log\left(\frac{y_k^{(i)}}{\frac{1}{2}\left(y_k^{(i)} + \bar{y}_k^{(i)}\right)}\right) + \right. \\ & \left. \sum_k \bar{y}_k^{(i)} \log\left(\frac{\bar{y}_k^{(i)}}{\frac{1}{2}\left(y_k^{(i)} + \bar{y}_k^{(i)}\right)}\right)\right), \end{aligned} \tag{11}$$

where t_k denotes the target for the k^{th} network outputs. To establish an upper bound on the strength parameter, we find the largest λ that keeps the second derivative (with respect to $y_k^{(i)}$) of (11) positive, ensuring the existence of a minimum. Differentiating this function gives

$$\frac{\partial E_i}{\partial y_k^{(i)}} \;\; = \;\; y_k^{(i)} - t_k - \frac{\lambda}{2}\log\left(\frac{y_k^{(i)}}{\frac{1}{2}(y_k^{(i)} + \bar{y}_k^{(i)})}\right) \tag{12}$$

$$\begin{aligned} = \;\; & y_k^{(i)} - t_k - \frac{\lambda}{2}\Big(\log\big(y_k^{(i)}\big) - \\ & \log\big(\frac{1}{2}(y_k^{(i)} + \bar{y}_k^{(i)})\big)\Big) \end{aligned} \tag{13}$$

$$\begin{aligned} \approx \;\; & y_k^{(i)} - t_k - \frac{\lambda}{2}\Big(\big(y_k^{(i)} - 1\big) - \\ & \big(\frac{1}{2}(y_k^{(i)} + \bar{y}_k^{(i)}) - 1\big)\Big) \end{aligned} \tag{14}$$

Note the use of the linear approximation $\log(x) \approx x - 1$. This approximation is fairly reasonable and avoids the unbounded behavior of $\log(x)$ in the interval $[0, 1]$, within which the inputs are guaranteed to be contained due to the use of the softmax activation functions for the network outputs. Differentiating a second time gives

$$\frac{\partial^2 E_i}{\partial \big(y_k^{(i)}\big)^2} \approx 1 - \frac{\lambda}{4}. \tag{15}$$

Setting up the inequality and rearranging the terms results in

$$1 - \frac{\lambda}{4} \;\; > \;\; 0 \tag{16}$$

$$\lambda \;\; < \;\; 4. \tag{17}$$

These calculations suggest that λ should be set to some value less than 4 in order to ensure that the network cost function can be minimized.

IV. Simulation Results

To test the effectiveness of the JSD ensemble method, it was compared to negative correlation learning (NCL) on two popular image classification benchmarks. The first benchmark consisted of a set of features developed by Adam Coates [5] for the CIFAR-10 dataset. For the second benchmark, we used the MNIST handwritten digits dataset. To investigate the performance of the JSD methods, we compared it to NCL over a range of strength parameter values for each benchmark. The particular benchmarks used were chosen based on their typical performance profiles. CIFAR is a moderately difficult benchmark, with standard learning algorithms able to reach a classification rate on the test set of 70-80% and state of the art performance above 88%. On the other hand, the MNIST dataset is much easier, with typical performance withing 96-98% and state of the art performance above 99%. We suspected that in this situation encouraging diversity may be harmful since a large number of learners in the ensemble will likely be producing correct classifications on almost all the training and testing examples. With the CIFAR benchmark, there are still a fairly significant number of examples that are being incorrectly classified and we expect more of a benefit from diversity. Both datasets have 10 target classes. These are also fairly large datasets, with 50,000 training examples for CIFAR and 60,000 for MNIST. Both datasets have 10,000 testing examples. We performed preprocessing on the input data to scale the inputs to the range [-1, 1].

We used the same ensemble architecture for every experiment: 11 neural networks with 32 hidden nodes each. Each network also used the cross entropy error function, defined as

$$E_i = -\sum_k t_k \log\left(y_k^{(i)}\right) + (1-t_k)\log\left(1-y_k^{(i)}\right) \qquad (18)$$

which allowed for faster learning and is well-suited for classification problems. Standard backpropagation and gradient descent were used to update the weights of each network, with an step size of $\frac{1}{256}$. To determine the output of the ensemble, the individual network outputs were averaged together with equal weights. Data points were collected for parameter values in the range $[0,4]$ for the CIFAR benchmark and $[-4,4]$ for the MNIST benchmark. The reason for investigating negative values for the MNIST benchmark is the suspicion that in situations were individual networks are very accurate, consensus may be more valuable than diversity. To obtain smoother curves and reduce noise in areas of interest, some data points were averaged among up to seven individual runs with differing initial conditions. Figures 1-4 show the maximum classification rate achieved on the training and test sets for each benchmark over 100 training presentations as a function of the strength parameter.

For the CIFAR benchmark (Figures 1-2), we see a very noticable performance advantage that JSD offers over NCL for both the training and the test set, while both methods perform better than the baseline ($\lambda = 0$). A very significant drop in performance for the JSD method occurs around $\lambda = 3.7$ and continues afterward, suggesting that the upper bound derived in Section 3 is applicable to the cross entropy cost function.

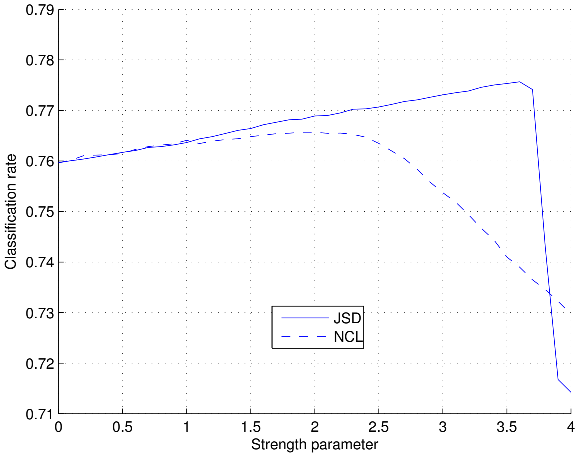

Figure 1. Maximum classification rate on the CIFAR training set over 100 training epochs.

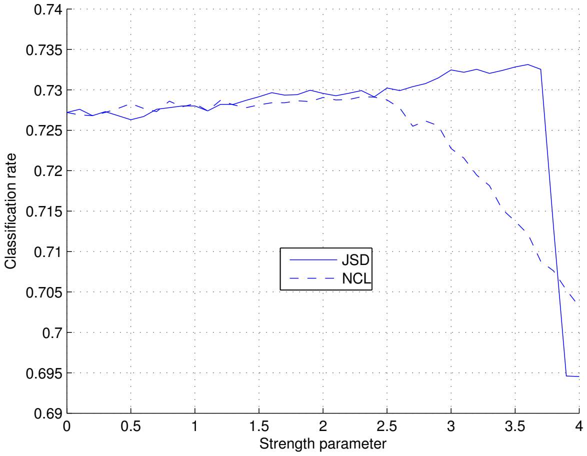

Figure 2. Maximum classification rate on the CIFAR test set over 100 training epochs.

The MNIST results (Figures 3-4) also show JSD outperformning NCL on the test set, but the training set data is less conclusive. Again, both methods outperform ensembles with no additional diversity terms. Thus the JSD method has provided superior generalization ability over NCL for this dataset. Results from both datasets suggest that, at least when using the cross entropy error function, the optimal strength parameter is in the range $[2, 3.7]$, with slightly smaller values being better suited for cases when the individual networks are highly accurate.

V. Conclusions

This paper explored the notion of employing an information-theoretic measure to encourage diversity among an ensemble of concurrently-trained neural networks. This forms an enhancement to a family of algorithms recently

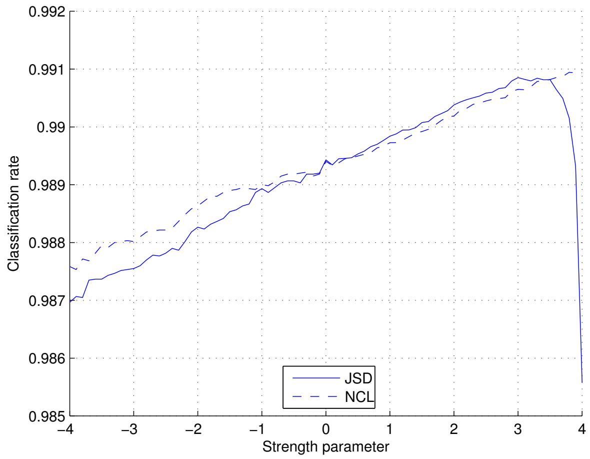

Figure 3. Maximum classification rate on the MNIST training set over 100 training epochs.

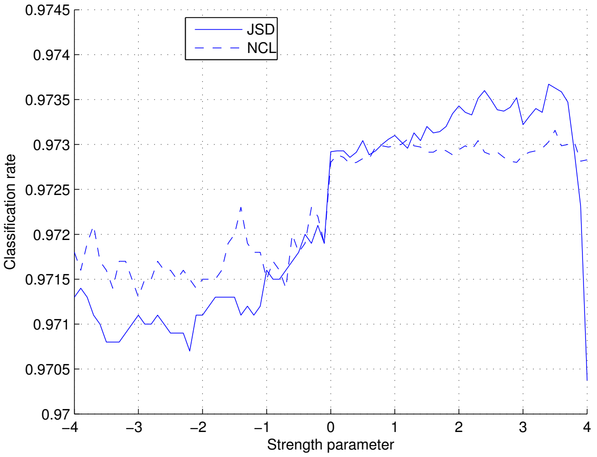

Figure 4. Maximum classification rate on the MNIST test set over 100 training epochs.

introduced for training multiple learners in parallel. It has been shown that compared to existing techniques, particularly negatively correlated learning, the proposed method yields higher overall performance in most cases while maintaining comparable computational complexity. A key element involved in successfully balancing accuracy and diversity has been addressed by deriving an upper-bound on the strength of the diversity penality term. The proposed methodology is particularly suitable for implementation on massively parallel processing platforms, which have become ubiquitous over the past few years.

There are still many issues to be investigated regarding the JSD ensemble method, including the effect of ensemble size, network complexity, and other cost functions. Furthermore, since both benchmark datasets used in this paper had a large number of examples and 10 target classes, additional datasets with different characteristics should be explored. Bounds on the strength parameter could likely be improved and extended to other cost functions, such as the cross entropy cost function. Since the performance of this method, as well as NCL, is highly dependent on the value of the strength parameter, an adaptive approach for creating diversity would be extremely useful. Finally, it would be interesting to investigate how JSD, or a similar information-theoretic measure, can be applied to regression problems, rather than just classification. At first glance, there seems to be a mismatch between techniques, since learners in regression problems do not typically produce probability distributions. However, this should definitely be explored in more detail.

Acknowledgment

This work was supported by the Intelligence Advanced Research Projects Activity (PARIAH) via Army Research Office (ARO) agreement number W911NF-12-1-0017. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of PARIAH, the Department of the Army, or the U.S. Government.

References

[1] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996.

[2] Gavin Brown. *Diversity in Neural Network Ensembles*. Computer science PhD thesis, University of Birmingham, 2004.

[3] Dan Ciresan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Handwritten digit recognition with a committee of deep neural nets on gpus. *CoRR*, abs/1103.4487, 2011.

[4] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[5] Adam Coates, Andrew Y Ng, and Serra Mall. The importance of encoding versus training with sparse coding and vector quantization. *Learning*, 2(1):921—928, 2011.

[6] Dominik Maria Endres and Johannes E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.

[7] S B Kotsiantis and P E Pintelas. Combining bagging and boosting. *Computational Intelligence*, 1(4):324–333, 2004.

[8] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[9] Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural networks: the official journal of the International Neural Network Society*, 12(10):1399–1404, 1999.

[10] Yong Liu and Xin Yao. Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 29(6), 1999.

[11] Kyoung-Su Oh and Keechul Jung. Gpu implementation of neural networks. *Pattern Recognition*, 37(6):1311 – 1314, 2004.

[12] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, July 1990.

[13] Robert E. Schapire. The boosting approach to machine learning an overview. *MSRI Workshop on Nonlinear Estimation and Classification*, 7(4):1–23, 2003.

[14] X. H. Shen, Z. H. Zhou, J. X. Wu, and Z. Q. Chen. Survey of boosting and bagging. *Computer Engineering and Applications*, 36(12):31–32, 2000.