STATISTICAL PERFORMANCE ANALYSIS OF MCE/GPD LEARNING IN GAUSSIAN CLASSIFIERS AND HIDDEN MARKOV MODELS

*Mohamed Afify**, *Xin-Wei Li*† *and Hui Jiang*†

* BBN Technologies
10 Moulton Street, Cambridge, MA, 02138
† Department of Computer Science & Engineering, York University
4700 Keele Street, Toronto, Ontario, M3J 1P3, CANADA

ABSTRACT

The minimum classification error, and generalized probabilistic descent (MCE/GPD) algorithm is a very popular and powerful framework for building classifiers with many practical applications. This paper first presents a theoretical analysis of MCE/GPD for a 2-class Gaussian classification problem. We show that the algorithm converges to the optimum classifier, and that further iterations lead to increasing the inter-class distance which increases the classifier variance without contributing to lowering its error. The theoretical results are supported by simulations for Gaussian classifiers, and generalize to a hidden Markov model speech recognition problem.

1. INTRODUCTION

A general paradigm for the design of parametric classifiers that capitalizes on minimizing the classification error was proposed in [1]. The basic idea is to develop a smoothed estimate of the classification error and minimize this estimate with respect to the parameters of interest using gradient descent. Thus, this approach is often referred to MCE/GPD, where MCE stands for minimum classification error, and GPD for generalized probabilistic descent. Since its advent this framework has found great success in many practical classification problems, and we refer the reader to [3] for a complete exposure.

This paper first focuses on a simple learning scenario, where the MCE/GPD algorithm is used to learn the means of a Gaussian classifier for a simple two-class problem. This setting leads to a relatively simple learning algorithm and allows us to derive detailed expressions for the evolution of the means, and both the true and smoothed errors, and to show the convergence of the classifier to the Bayes solution. In addition, we relate further iterations to increasing the variance of the classifier, and hence reducing its generalization capability. This behavior, referred to over-training, is often observed in practical applications of the MCE/GPD framework. In spite of the simplicity of the analyzed algorithm it can be readily identified as a special case, i.e., Viterbi training, of MCE/GPD learning of the means of Gaussian mixture hidden Markov models (HMMs)[5]. The latter algorithm is of great practical interest especially in speech recognition applications. This similarity motivates us to believe that the theoretical results obtained for the simple Gaussian classifier can carry over to MCE/GPD learning of Gaussian mixture HMMs. This theory is experimentally verified, using a setup to be detailed in the paper, in E-set speech recognition experiments, where it is shown that the behavior of the learning agrees with the theoretical results obtained for the simple Gaussian classifier.

The rest of the paper is organized as follows. The simple classification problem and the associated discriminative learning algorithm are formulated in Section 2. Section 3 contains the analysis results of the algorithm of Section 2. Experimental results on E-set speech recognition are given in Section 4. Finally Section 5 summarizes our findings.

2. ALGORITHM FORMULATION

This section presents a simple classification problem for which the discriminative training algorithm of [1] will be formulated and analyzed. Assume we have two classes C_j where $j \in \{0, 1\}$. For class C_j the probability density function (pdf) of the observations is Gaussian given by $\mathcal{N}(\theta_j, \sigma^2)$, where θ_j is the mean for class C_j, and σ^2 is a common variance. The solution of the above classification problem is known [2]. Assuming, without loss of generality, that the two classes are equiprobable, and $\theta_1 > \theta_0$, the optimal classifier reduces to comparing an observation x to a threshold $t = (\theta_1 + \theta_0)/2$, and deciding C_1 if $x > t$, and C_0 otherwise.

Define the misclassification function of class C_j as

$$\begin{aligned} d_j(x, \mu_j, \mu_{1-j}) &= \log p_{1-j}(x) - \log p_j(x) \\ &= \frac{(\mu_{1-j} - \mu_j)x}{\sigma^2} + \frac{(\mu_j^2 - \mu_{1-j}^2)}{2\sigma^2} \end{aligned} \quad (1)$$

where we also note that $d_{1-j}(x, \mu_j, \mu_{1-j}) = -d_j(x, \mu_j, \mu_{1-j})$. Also define the smoothed error for pattern x as $e(x, \mu) = \Phi(d_j(x, \mu)) \quad x \in C_j$, where $\Phi()$ is the standard Gaussian CDF (cumulative density function).[1] Applying the discriminative training paradigm of [1], with these definitions in mind, to learn the class means in the above classification problem, it is easy to derive the following update equations

$$\mu_j(n+1) = \begin{cases} \mu_j(n) + \frac{\epsilon}{\sigma^2}\phi(d_j(x(n+1), \mu(n)))(x(n+1) - \mu_j(n)) \\ \quad \text{if } x(n+1) \in C_j; \\ \mu_j(n) - \frac{\epsilon}{\sigma^2}\phi(d_j(x(n+1), \mu(n)))(x(n+1) - \mu_j(n)) \\ \quad \text{if } x(n+1) \in C_{1-j}. \end{cases} \quad (2)$$

where $\mu_j(n)$, and $\mu_{1-j}(n)$ represent the mean variables in the misclassification function at instant n, $x(n+1)$ represent the observation at time $n+1$, ϵ is the learning rate,

[1] Without loss of generality, the sigmoid function in [1] is replaced by the Gaussian CDF here.

and $\phi()$ is the standard Gaussian function. These update equations are known to minimize the expected value of the smoothed error given by $E[e(x,\mu)]$, where E is the expectation operator with respect to $p(x)$. The expected true error can be obtained by replacing the Gaussian CDF by a unit step function. The mean update in Equation (2) together with the associated expected values of the smoothed error, and true error are the focus of the analysis in the following section.

3. ALGORITHM ANALYSIS

In this section we perform statistical analysis of the algorithm given in Section 2. In particular, we focus on deriving difference equations for $E[\mu_j(n+1)]$, and the associated decision threshold. In addition we calculate expressions for the expected values of the smoothed and true error. We finally study the evolution of the variance of the decision threshold. These are referred to as transient, error, and variance analysis respectively. It is worth noting the similarity of some used techniques to [4], and that many derivations are omitted for lack of space, and can be found in [6].

3.1. Transient Analysis

This subsection derives difference equations for $E[\mu_j(n+1)]$, and the associated decision threshold followed by studying the convergence behavior of the decision threshold. We start by the difference equation for class means. To this end we write

$$E[\mu_j(n+1)|\mu(n)] = P_j E[\mu_j(n+1)|\mu(n), x(n+1)\in C_j] + P_{1-j}E[\mu_j(n+1)|\mu(n), x(n+1)\in C_{1-j}] \quad (3)$$

where P_j, and P_{1-j} are the a priori probabilities of classes C_j, and C_{1-j} respectively. Both expectations on the right hand side of Equation (3) can be evaluated using techniques for calculating expectations of non-linear functions of Gaussian variables [6]. Once these expressions are obtained, and making the assumptions[2] $P_j = P_{1-j} = 0.5$, and $\mu(n)$ is concentrated[3] at $E[\mu(n)]$, and denoting $E[\mu_j(n+1)] = \overline{\mu_j(n+1)}$, we get a difference equation that describes the evolution of the mean $\overline{\mu_j(n+1)}$ during learning.

Further, if we define $\overline{t(n)} = (\overline{\mu_1(n)} + \overline{\mu_0(n)})/2$ as the evolution of the decision threshold during learning, and substitute in the difference equation of mean evolution, for $j = 0, 1$, together with some algebraic simplifications we arrive at the following recursion for the decision threshold

$$\overline{t(n+1)} = \overline{t(n)} + \frac{0.25\epsilon\overline{\Delta^{(1)}\mu(n)}}{\sigma\sqrt{\overline{\Delta^{(1)}\mu(n)}^2 + \sigma^2}} \times \left[\phi\left(\frac{\overline{\Delta^{(1)}\mu(n)}(\theta_1 - \overline{t(n)})}{\sigma\sqrt{\overline{\Delta^{(1)}\mu(n)}^2 + \sigma^2}}\right) - \phi\left(\frac{\overline{\Delta^{(1)}\mu(n)}(\overline{t(n)} - \theta_0)}{\sigma\sqrt{\overline{\Delta^{(1)}\mu(n)}^2 + \sigma^2}}\right)\right] \quad (4)$$

where $\overline{\Delta^{(j)}\mu(n)} = \overline{\mu_{1-j}(n)} - \overline{\mu_j(n)}$.

Now when $n \to \infty$, and the threshold has converged, we have $\overline{t(n+1)} = \overline{t(n)} = t^*$. The steady state threshold t^* can be calculated by equating the second term on the right hand side of Equation (4) to zero. After simple calculations we get, excluding the case that $\overline{\Delta^{(1)}\mu(n)} = 0$, $t^* = (\theta_1 + \theta_0)/2 = t$. Hence at steady state the decision threshold will converge to its optimal value. In addition, it can be shown that the threshold always moves in the direction of the optimal value during learning, which guarantees its convergence to the optimal value[4].

3.2. Error Analysis

In this section we derive expressions for the expectations of the smoothed error $E[e(x,\mu(n))]$, and the true error. In addition, we discuss some of their properties which will help us in studying the behavior of the learning. We start by the expectation of the smoothed error, which can be written as

$$E[e(x,\mu(n))|\mu(n)] = P_j E[e(x,\mu(n))|\mu(n), x\in C_j] + P_{1-j}E[e(x,\mu(n))|\mu(n), x\in C_{1-j}] \quad (5)$$

Evaluating the expectations on the right hand side of Equation (5), using assumptions similar to the previous subsection, and denoting $\overline{e(n)} = E[e(x,\overline{\mu(n)})]$, we arrive at [6]

$$\overline{e(n)} = 0.5\left[\Phi\left(\frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right) + \Phi\left(\frac{d_{1-j}(\theta_{1-j},\overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right)\right] \quad (6)$$

A similar expression can be obtained for expected value of the true error ($e_T(x)$) as

$$\overline{e_T(n)} = 0.5\left[\Phi\left(\frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{|\overline{\Delta^{(j)}\mu(n)}|}\right) + \Phi\left(\frac{d_{1-j}(\theta_{1-j},\overline{\mu(n)})\sigma}{|\overline{\Delta^{(j)}\mu(n)}|}\right)\right] \quad (7)$$

In Section 2 we assumed that $\theta_1 > \theta_0$, further assuming that $\overline{\mu_1(n)} > \overline{\mu_0(n)}$ is preserved during the learning, the error expression in Equation (7) can be simplified to

$$\overline{e_T(n)} = 0.5\left[\Phi\left(\frac{\overline{t(n)} - \theta_1}{\sigma}\right) + \Phi\left(\frac{\theta_0 - \overline{t(n)}}{\sigma}\right)\right] \quad (8)$$

[2]Any values can be used as long as we use the same values for presentation of training examples.

[3]This assumption is used to avoid integrating out the conditioning in Equation (3), and is known to be reasonable for small learning rate[4].

[4]This is because the difference between the initial and optimal values is finite and every step is in the right direction.

This is similar to the Bayes error (e_B)[2], except for replacing t by $\overline{t(n)}$, and will converge to e_B when $\overline{t(n)} \to t$. Hence as the decision threshold evolves towards its optimal value, as discussed in the previous section, the true error will decrease until it converges to its minimum value, the Bayes error. Next, examining Equations (6) and (7), it can be shown that under some reasonable conditions we have $\overline{e(n)} \geq \overline{e_T(n)}$ with equality when $|\overline{\Delta^{(j)}\mu(n)}|/\sigma \to \infty$. The latter upper bound property of the smoothed and true error developed in this section together with the transient analysis of the previous section suggest that the learning algorithm will continue to increase $|\overline{\Delta^{(j)}\mu(n)}|/\sigma$ to reach the true error which is a lower bound of the smoothed error objective function. Noting that $|\overline{\Delta^{(j)}\mu(n)}|/\sigma$ is proportional to the class mean distance. The increase of this ratio will thus cause the means to drift (continue to move apart) even after convergence to the optimal classifier. In the next subsection we will outline the relationship between this mean drift and the increase of variance of the classifier.

3.3. Variance Analysis

This subsection will relate the mean drift property discussed in the previous subsection to the increase of the variance of the classifier[5], and hence the reduction of its generalization capability, or over-training. This behavior, which contributes to the increase of test set error, is often observed when running excessive iterations of MCE/GPD or other discriminative learning methods. Hence the developed expression can be used to monitor variance increase and may help in overcoming this negative effect. By writing an expression for the decision threshold and using techniques similar to the previous subsections for the calculation of its variance, we arrive at the following expression for the variance of the decision threshold[6]

$$\begin{aligned} var[t(n+1)|\overline{\mu(n)}] \quad = \quad & -(\overline{t(n+1)} - \overline{t(n)})^2 + \\ & \frac{\epsilon^2}{8\sqrt{2\pi}\sigma}\left(\frac{\overline{\Delta^{(j)}\mu(n)}}{\sigma}\right)^2 \times \\ & \frac{1}{\sqrt{2\overline{\Delta^{(j)}\mu(n)^2} + \sigma^2}} \times \\ & \left[\phi\left(\frac{\sqrt{2}d_j(\theta_j, \overline{\mu(n)})\sigma}{\sqrt{2\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right)\right. \\ & \left. + \phi\left(\frac{\sqrt{2}d_{1-j}(\theta_{1-j}, \overline{\mu(n)})\sigma}{\sqrt{2\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right)\right] \end{aligned} \quad (9)$$

At convergence $\overline{t(n+1)} = \overline{t(n)}$, and hence the first term in Equation (9) will vanish leaving the second term which increases with the mean drift.

4. EXPERIMENTS: SPEECH RECOGNITION BASED ON HMMS

We have performed extensive simulation with Gaussian data, and in all experiments the predicted theoretical behavior perfectly match the simulation result [6]. In this section we will study the performance of MCE/GPD training of HMMs for an English E-set alphabet recognition task.

[5]Decision threshold in our case.

Motivated by the similarity of the Gaussian classifier update equation (Equation (2)) to MCE/GPD learning of the means of Gaussian mixture HMMs [5], we may expect that the obtained theoretical results will carry over to the HMM case. Thus, we empirically study the behavior of MCE/GPD iterations for Gaussian mixture HMMs to verify the suggested learning pattern. That is, the training set error (true error), and MCE objective function (smoothed error) will decrease, with the smoothed error being an upper bound of the true error. After the true error rate saturates the MCE objective function will continue to decrease causing only the "distance" between models to increase, and possibly hurting generalization (test set error). Details of the experimental setup used to verify this pattern are given below.

The experiments are performed on the English E-set vocabulary of ISOLET database, consisting of {B, C, D, E, G, P, T, V, Z}. ISOLET is a database of letters of the English alphabet spoken in isolation. The database consists of 7800 spoken letters, two productions of each letter by 150 speakers, 75 male and 75 female. The recordings were done under quiet, laboratory conditions with a noise-canceling microphone. The data were sampled at 16 kHz with 16-bit quantization. ISOLET is divided into five parts named ISOLET 1-5. In this experiment, only the first production of each letter in ISOLET 1-4 is used as training data. All data in ISOLET 5 is used as testing data. The feature vector is of 39 dimensions, which include 12-d static MFCC, log-energy, delta and acceleration coefficients.

An HMM recognizer with 16-state, 1-mixture per state whole-word based models is trained by HTK to be the initial models for MCE training. The recognizer achieves an accuracy of 93.4% for the total 26 letters and an accuracy of 85.56% for the E-set letters of the test data set.

In our MCE/GDP, discriminant function is normalized by the utterance length and feature dimension. Both means and variance will be updated for each training sample. The step size used in this experiment is $\epsilon = 3$. The weight η in the misclassification function is set to 4. The scale γ in sigmoid function is set to 2 and the shift θ is set to 0[5]. For the E-set test, the recognition rate for the training data set is improved from 93.89% to 99.91% (only one misclassification left), while the recognition rate for the testing data set is improved from 85.56% to 91.67%. In Figure 1 we plot the results for both the training and the testing sets as a function of the number of iterations in the MCE training procedure. Each iteration includes updates over all the training samples.

In Figure 1, "Train" stands for the recognition rate of the training data set. "Test" stands for the recognition rate of the testing data set. "Smoothed Rate" stands for the smoothed recognition rate of the training data set. These three curves use the y-axis on the left side. "Euclidean" stands for the summation of Euclidean distances between each pair of models [7]. "KL" stands for the summation of Kullback-Leibler distances between each pair of models [7, 8]. These two curves use y-axis on the right side. Note that recognition rate is used, in the figure, instead of error for convenience, hence, for example, the "Smoothed Rate" shows as a lower bound to the "Train".

The curve "Train" shows that the true error converges after about 25 iterations. After the convergence of the true recognition rate, the smoothed recognition rate for the training data set (curve "Smoothed Rate") continues to in-

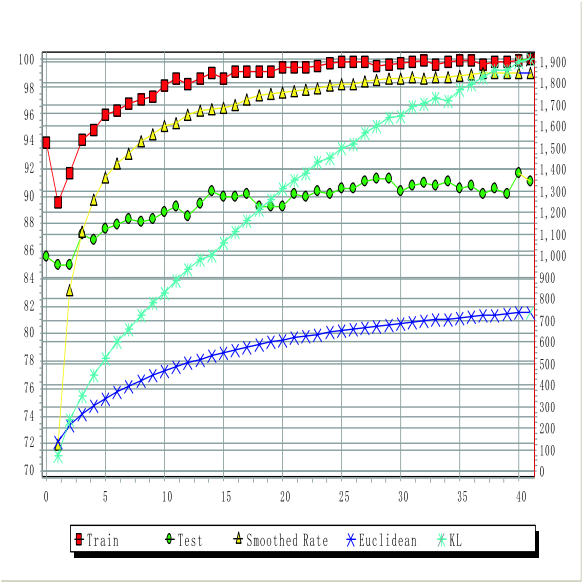

Figure 1: The MCE/GPD learning curves in HMM-based speech recognition.

crease towards the true recognition rate. This agrees with what was shown in the Gaussian case that the smoothed error objective function is an upper bound of the true error on training data. In addition, the recognition rate for testing data (curve "Test") has no apparent improvement (even drops down a little) after the true error converges, which suggests that over-training starts to occur. This agrees pretty well with the pattern suggested at the beginning of this section. Also another important observation is that the distances between models continues to increase (in Euclidean and KL sense) even after the true error rate converges. This indicates the "mean drift", here better called "model drift", found in the theoretical analysis. This drift is expected to contribute to the increase of the variance of the classifier and hence reducing its generalization ability, though this is not theoretically proved for the HMM case.

A few comments regarding our experimental setup are worth mentioning here. Due to our desire of keeping a practical, yet simple, speech recognition scenario, we deviated from our theoretical framework in some aspects. First, a multiple class recognition problem was considered. It was shown in [9] how to formalize an MCE objective function using pairwise mis-classification measures, and this work can be considered as a starting point to generalize our analysis to the general multiple class problem. In this work to handle this generalization we used the averaged pairwise model distance in place of the normalized inter-class mean distance $\overline{|\Delta^{(j)}\mu(n)|}/\sigma$ of Section 3. Second, we also chose to update both means and variances while the analysis considered only mean estimation for fixed variance. The analysis of mean estimation for changing variance is a trivial extension to that of this paper by conditioning on the variance, while analysis of variance recursion is more difficult and was not addressed here.

5. SUMMARY

This paper first considers the analysis of learning the class means using the MCE/GPD framework for a two-class Gaussian classifier. It was shown that the algorithm converges to the Bayes solution and that after convergence the normalized inter-class mean distance increases ("mean drift"), this drift was related to the increase of the classifier variance, and hence the reduction of its generalization capability or over-training. One interesting implication is that over-training, which is not directly measurable, is now related to an inter-class distance which can be monitored during learning. The similarity of the considered learning algorithm to MCE/GPD updates of Gaussian mixture HMMs motivated us to empirically study the behavior of these updates, and they turned out to agree pretty well with the pattern suggested for the Gaussian classifier, where it was found that the averaged pairwise model distance continues to increase after the training error converges in a multi-class E-set recognition problem.

6. REFERENCES

[1] B.-H. Juang, and S. Katagiri, "Discriminative learning for minimum error classification," IEEE Trans. Signal Processing, vol. 40, no. 12, pp. 3043-3054, December 1992.

[2] R. Duda, P. Hart, and D. Stork, Pattern Classification (Second Edition). Wiley-Interscience, October 2000.

[3] S. Katagiri, B.H. Juang, and C.H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," Proceeedings of the IEEE, vol. 86, no. 11, pp. 2345-2373, Nov. 1998.

[4] N. Bershad, J. Shynk, and P. Feintuch, "Statistical analysis of single-layer backpropagation algorithm: Part I– Mean weight behavior, " IEEE Trans. Signal Processing, vol. 41, no.2, pp. 573-582, Feb. 1993.

[5] B.-H. Juang, W. Chou, and C.H. Lee, " Discriminative methods for speech recognition," IEEE Trans. Speech and Audio Processing, vol. 5, no. 3, pp. 257-265, May 1997.

[6] M. Afify, Statistical Analysis of Minimum Classification Error Learning: The Gaussian Case, manuscript in review. (available at http://www.cs.yorku.ca/~hj/afify.pdf)

[7] Homayoon S. M. Beigi, Stephane H. Maes and Jeffrey S. Sorensen, "A Distance Measure between Collections of Distributions and its Application to Speaker Recognition", Proc. of ICASSP'98, Seattle, WA, April 1998.

[8] Chao-Shih Huang, Hsiao-Chuan Wang and Chin-Hui Lee, "A Study on Model-Based Error Rate Estimation for Automatic Speech Recognition," IEEE Trans. Speech and Audio Procesing, vol. 11, No. 6, pp. 581-589, November 2003.

[9] E. McDermott, and S. Katagiri, "A new formalization of minimum classification error using a parzen estimate of classification chance," in Proc. ICASSP'03, Hong Kong, April 2003.