# Maximum Entropy Classification Applied to Speech

Maya Gupta
Dept. of Electrical Engineering
Stanford University
Stanford, CA
guptama@stanford.edu

Michael P. Friedlander
Dept. of Management Science and Engineering
Stanford University
Stanford, CA
mpf@stanford.edu

Robert M. Gray
Dept. of Electrical Engineering
Stanford University
Stanford, CA
rmgray@stanford.edu

## Abstract

*We present a new method for classification using the maximum entropy principle[1], allowing full use of relevant training data and smoothing the data space. To classify a test point we compute a maximum entropy weight distribution over a subset of training data and constrain the weights to exactly reconstruct the test point. The classification problem is formulated as a linearly constrained optimization problem and solved using a primal-dual logarithmic barrier method, well suited for high-dimensional data. We discuss theoretical advantages and present experimental results on vowel data which demonstrate that the method performs competitively for speech classification tasks.*

## 1 Introduction

Speech processing may use supervised classification to determine what is being said (speech recognition), to determine who spoke (speaker recognition), and in subproblems, such as segmenting the acoustic waveform.

Speech recognition systems begin with an acoustic processor that accepts a speech waveform and outputs either feature vectors over time (such as cepstral coefficients) or a sequence of symbols (such as estimated phones). The feature vector signal or symbol sequence may then be passed onto a linguistic processing unit. If the acoustic processor is to output a sequence of symbols it must contain a classification unit.

[1]This work was partially supported by the NSF under grant 2DTA442.

Speaker recognition systems receive speech waveforms, compute feature vectors, and then classify the feature vectors as one of the categories of speakers.

Both of these classification problems are examples of *supervised classifiers*—the classifier has at its disposal a set of labeled training data and is given the task of automatically classifying similar test data. Effective classification algorithms vary widely from simple nearest-neighbor-type methods to more complex techniques such as discriminant analysis, decision trees, and neural nets [4]. Each classification method models the data space differently and thus may be more or less suited to a particular real-world application or feature representation.

We present a new classification algorithm that uses the maximum entropy principle (see, for example, [8]) and suggest its application to the classification of speech feature vectors. In Section 2 we explain the maximum entropy classification (MEC) algorithm and present its theoretical underpinnings in Section 3. An efficient implementation of the algorithm is described in Section 4. The results of vowel recognition experiments on benchmark reflection coefficient vowel data are discussed in Section 6 and compared to standard classification methods.

## 2 The MEC algorithm

Define a *labeled training* point as a pair $(\mathbf{x}, g) \in \mathbb{R}^n \times \mathcal{G}$, where $\mathcal{G}$ is a set of class labels. Given a set of $p$ labeled training points $(\mathbf{x}_1, g_1), (\mathbf{x}_2, g_2), \ldots, (\mathbf{x}_p, g_p)$, the proposed method classifies a test point $\widehat{\mathbf{x}}$ as belonging to a class $\widehat{g}$ as follows:

0-7803-6514-3/00/$10.00©2000 IEEE

**Step 1** Choose a subset of the training points as a *basis* for $\widehat{\mathbf{x}}$ (e.g., all training points falling within a fixed radius of $\widehat{\mathbf{x}}$). Denote the basis by the subset $\beta = \{j_1, j_2, \ldots, j_k\}$, where $k \leq p$.

**Step 2** Calculate the distribution of weights $\mathbf{w} = [w_{j_1}, w_{j_2}, \ldots, w_{j_k}]^T$ that solves

$$\underset{\mathbf{w}}{\text{maximize}} \quad -\sum_{i \in \beta} w_i \log w_i \tag{1}$$

subject to the constraints

$$\sum_{i \in \beta} w_i \mathbf{x}_i = \widehat{\mathbf{x}} \tag{2}$$

$$\sum_{i \in \beta} w_i = 1 \tag{3}$$

$$\mathbf{w} \geq 0. \tag{4}$$

**Step 3** Classify $\widehat{\mathbf{x}}$ as belonging to class $\widehat{g}$, the solution to

$$\underset{g \in \mathcal{G}}{\text{maximize}} \sum_{i \in \beta} w_i \delta(g_i, g),$$

where

$$\delta(g_i, g) = \begin{cases} 1 & \text{if} \quad g_i = g; \\ 0 & \text{if} \quad g_i \neq g. \end{cases}$$

In words, the algorithm says that for each test point, find the training points in its neighborhood, and then solve for the weighting that each of these local training points should receive so that the linear weighted sum reconstructs the test point. Finding these weights is equivalent to solving a matrix equation of the form $Ax = b$. However, if there are more points in the chosen neighborhood than feature dimensions (the linear problem is underdetermined), then there will be more than one possible solution for the weight vector. In those cases we choose from those weight vectors which satisfy the linear reconstruction constraint the weight vector that has the maximum entropy. If there are not enough training points in the neighborhood to reconstruct the test point (i.e. the linear problem is not feasible), then we give equal weighting to all the neighborhood points.

We consider choosing the neighborhood to be a parameter of the algorithm that can be trained on the training data (note that this the only training the algorithm requires). Both how to determine the neighborhood and the size of the neighborhood are open for experimentation. We have experimented with nine ways to define a local neighborhood, including using all $k$ nearest-neighbors to each test point (training the parameter $k$ on the training data).

Note several important aspects of the optimization problem in Step 2 of the algorithm. The objective function defined by (1) is concave. Thus, if the basis generated (the set of local training points) for a particular test point admits a feasible problem[2], the maximum entropy weight solution $\mathbf{w}^*$ will be necessarily unique.

If the generated basis does not admit a feasible problem, a regularization term is introduced into the objective and the constraints allow the weight distribution to be calculated in some least-squares sense. This approach will be made more precise in Section 4.

## 3 Theory of MEC

Intuitively, the maximum entropy weight distribution chosen by the MEC algorithm allows for the use of all relevant training data. By linearly reconstructing the test point we take into account the location of each neighborhood training point, instead of the just the distance, as is done in nearest-neighbor methods [5]. This is especially important in problems with a high number of feature dimensions because in high dimensions all points are far apart. A linear reconstruction is the least biased reconstruction possible. Solving for a distribution of weights over the training points that linearly reconstructs the test point is the same problem as solving for a probability distribution with known mean. When there is more than one distribution possible, the weight distribution that has the maximum entropy is in fact the maximum likelihood estimate of the weights [12]. In some cases the algorithm's performance may be improved by using an informed prior and then solving for the distribution with minimum relative information with respect to the prior. An important attribute of the maximum entropy weighting is that it creates smooth solutions over the data space, unlike separating hyperplane methods.

## 4 Efficient Implementation

Maximum entropy methods have been used in many applications, including image restoration and density estimation for hidden Markov models in speech recognition [9, 12]. Consequently, a number of methods for solving for the maximum entropy distribution have been proposed, including hill-climbing, iterative projection, the damped Newton method, and iterative scaling. Two papers that propose implementations are [2, 1]

The bulk of the computational work in the MEC algorithm resides in the optimization problem of Step 2. For the MEC algorithm to be efficient and scalable, a suitable method for calculating a constrained maximum entropy solution must be used. By efficient and scalable, me mean that it exploits structure in the problem to avoid unnecessary

[2] An optimization problem is said to be *feasible* if there exists a point satisfying all the constraints simultaneously. The set of such points is the *feasible set*. The optimization problem is *infeasible* if that set is empty.

storage requirements, has a high rate of convergence, and the computational complexity grows at rate proportional to the problem size.

Our solution method is based on a primal-dual log-barrier code implemented by Saunders [11]. This implementation belongs to the class of *interior-point primal-dual* methods, so called because all iterates remain in the interior of the inequality constraints (in this case, all iterates are strictly positive). See Nocedal and Wright [10] for a discussion on interior-point methods.

The implementation has several useful qualities. First, the iterates generated at each step of the algorithm remain strictly positive. Thus, the maximum entropy objective function is guaranteed to always be well defined. Second, the iterates do not have to be feasible and we can avoid the costly calculation of determining a feasible starting point. Third, the algorithm makes explicit use of the positive definite, diagonal structure of the Hessian.

In particular, the implementation solves a perturbed, linearly constrained optimization problem of the form

$$\begin{aligned} \underset{\mathbf{x},\mathbf{r}}{\text{minimize}} \quad & f(x) + \tfrac{1}{2}\|\gamma \mathbf{r}\|^2 + \tfrac{1}{2}\|\mathbf{x}/\delta\|^2 \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{r} = 0 \\ & \mathbf{x} \geq 0. \end{aligned}$$

The regularization term $\frac{1}{2}\|\gamma \mathbf{r}\|^2$ serves to guarantee that if $\mathbf{A}\mathbf{x} = \mathbf{b}$ is infeasible, a solution in some least-squares sense can be found. The second regularization term $\frac{1}{2}\|\mathbf{x}/\delta\|^2$ ensures that the solution will be bounded. A discussion of regularization can be found in Gill et al. [6]. The parameter $\gamma$ is chosen to balance the least squares solution of $\mathbf{w}$ against the maximum entropy solution. Regularization parameter values of $\gamma = 0$ and $\delta = 10$ (FILL IN) yielded good performance.

Classifiers for speech problems are trained on millions of data points, although with usually less than 36 feature dimensions. Normally we would want to exploit the *sparsity* of the underlying constraint matrix. However, the feature vectors generally have few nonzero entries and so $\mathbf{A}$ would be *dense*. Solving optimization problems of this magnitude would be extremely costly to solve without leveraging efficiencies in storage and computation. A novel feature of the chosen method that we are not exploiting is the ability to define the matrix $\mathbf{A}$ as an operator. One avenue of exploration still left to us is the question of whether efficient means for calculating $\mathbf{A}\mathbf{x}$ and $\mathbf{A}^T\mathbf{x}$ are possible. This would allow us to tackle a wider range of problem sizes. Effectively, this is a substitute for sparsity of $\mathbf{A}$.

## 5 Computational Complexity

Classification may be slowed by huge training data sets. There are at least two ways to ameliorate this problem, clustering and interior-point weeding. Clustering methods, such as the Lloyd algorithm or k-means, accept a parameter $k$ and for each class, iteratively create $k$ clusters to represent all the data points of that class. Each of the $k$ clusters is represented by its centroid. To minimize the training data set it may be suitable to replace data with their clusters' centroids. However, using centroids instead of the original clusters will not guarantee the same results offered by using the original training data.

On the other hand, interior-point weeding will be less efficient at reducing the number of used training points, but should not change the results. In interior-point weeding, any training point that is contained entirely within the convex hull formed by the other training points within the same class are removed. This can be determined using a straightforward linear program.

## 6 Vowel Recognition Experiment

As discussed in the Introduction, supervised classification problems may arise in speech processing both in speech recognition and speaker recognition. We demonstrate the classifier on a standard set of vowel data available from the Information and Computer Science Department at the University of California, Irvine [3]. The training data consists of 528 data points from eight mixed-gender speakers saying eleven different words and six data points taken from the steady-state vowel of each word. The test data is composed of 462 data points from seven speakers. The eleven words (classes, classes in this context) are the steady-state vowels of British English: hid, hId, hEd, hAd, hYd, had, hOd, hod, hUd, hud, hed. The speech signals were low pass filtered at 4.7 kHz and then passed through a 12 bit ADC with a 10 kHz sampling rate. Six 512 sample Hamming windowed segments were taken from the steady part of each word's vowel and then analyzed with twelfth order linear predictive analysis. Reflection coefficients were used to calculate ten log area parameters which are entered as a ten dimensional data point to the classifier.

For each test point we use as a basis the subset of the training data that falls within a radius of

$$(1+\alpha)\left[\min_{i\in\{1,\ldots,p\}} \|\mathbf{x}_i - \widehat{\mathbf{x}}\|\right],$$

where the parameter $\alpha$ was chosen to provide the best classification rate on the training data via cross-validation. The cross-validation was eight-fold with each speaker removed from the training set in turn and the remaining data used as the test set. For the results described in Table 1, $\alpha = .26$. As a control, we experimented with using our neighborhood selection but then setting the weights equally for all points in the neighborhood. We did not train the neighborhood

| Classification Method | Test Points Correctly Classified |
|---|---|
| Single-layer Neural Network | 33% |
| Linear Discriminant Analysis | 44% |
| Multi-layer Neural Network | 51% |
| CART (decision tree) | 54% |
| Nearest Neighbor | 56% |
| Flexible Discriminant Analysis | 61% |
| Proposed Maximum Entropy Classifier | 61% |
| Equal weight on all points in neighborhood | 62% |

**Table 1. Comparison of classifiers on the vowel dataset.**

parameter, but chose it to be $\alpha = .26$ to make a good comparison to the proposed maximum entropy classifier.

The best classifier we found for this data set is Flexible Discriminant Analysis [7] which achieves a correct classification rate for the vowel test set of 61%. Table 1 compares the performance of our algorithm against the published results of other classifiers [5]. The maximum entropy classifier also achieves 61%. Using the same neighborhood selection but giving all points equal weight, 62% of the points were classified correctly. This highlights the importance of a good neighborhood selection procedure. Since there are 462 test points, and the estimation of the error rate for the classifier is a random variable with a binomial distribution, the expected standard deviation for the probability estimates of Table 1 is upper bounded by 2.33%.

## 7 Conclusions

Maximum entropy classification is a new and well-founded method for supervised classification. There is only one parameter to set - the neighborhood selection - and thus it is easy to train. We have shown that it performs competitively on speech applications and hope it will find use in the future. Though not as readily applied to speech, the same algorithm may be applied to regression problems.

## References


[1] C. Byrne. Iterative image reconstructionn algorithms based on cross-entropy minimization. *IEEE Transactions on Image Processing*, 2(1):96–103, 1993.

[2] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mamthematical Statistics*, 43:1470–1480, 1972.

[3] D. Deterding, M. Niranjan, and T. Robinson. Vowel recognition: Deterding dataset, 1989. www.ics.uci.edu/pub/machine-learning-databases/undocumented/connectionist-bench/vowel.

[4] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, NY, 1996.

[5] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer-Verlag, New York, NY, 2001.

[6] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Solving reduced KKT systems in barrier methods for linear and quadratic programmin. Technical Report SOL91-7, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305-4026, USA, July 1991.

[7] T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89(428):1255–1270, 1994.

[8] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[9] F. Jelenik. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, 1997.

[10] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

[11] M. A. Saunders. PDSCO MATLAB code. Unpublished, June 2000.

[12] N. Wu. *The Maximum Entropy Method*. Springer-Verlag, Berlin, 1997.