NOISE ROBUST PHONETIC CLASSIFICATION WITH LINEAR REGULARIZED LEAST SQUARES AND SECOND-ORDER FEATURES

Ryan Rifkin[*] Honda Research Insitute, USA
Ken Schutte[†] MIT CSAIL, USA
Michelle Saad[‡] AUB, Lebanon
Jake Bouvrie[§] MIT CBCL, USA
Jim Glass[†] MIT CSAIL, USA

ABSTRACT

We perform phonetic classification with an architecture whose elements are binary classifiers trained via linear regularized least squares (RLS). RLS is a simple yet powerful regularization algorithm with the desirable property that a good value of the regularization parameter can be found efficiently by minimizing leave-one-out error on the training set. Our system achieves state-of-the-art single classifier performance on the TIMIT phonetic classification task, (slightly) beating other recent systems. We also show that in the presence of additive noise, our model is much more robust than a well-trained Gaussian mixture model.

***Index Terms*—** Acoustic noise, Artificial Intelligence, Speech Analysis, Speech Recognition

1. INTRODUCTION

The primary acoustic modeling technique in automatic speech recognition is the Gaussian mixture model (GMM), which represents a probability distribution as a mixture of (usually diagonal) Gaussians. The GMM is a generative model that can represent arbitrary distributions given a sufficient number of Gaussians, and is computationally tractable to train. On the other hand, in recent years the machine learning community has had great success with discriminative models such as support vector machines (SVMs) [1] and related methods. The central insight is that estimating a discriminative boundary between regions in which one distribution is larger than another may be much easier than estimating those distributions accurately everywhere, and that good performance can be obtained by concentrating our modelling efforts on finding this boundary.

While SVMs give excellent performance on a range of tasks, training a general nonlinear SVM on n data points will frequently scale (in practice) as $O(n^2)$ or worse in both time and memory, making them intractable for the large problems with hundreds of thousands or millions of points arising in ASR. In this paper, we avoid these problems via the combination of two techniques: explicitly lifting our low-dimensional features into a higher dimensional space in which a *linear* decision boundary will perform well, and using regularlized least squares (RLS) [2, 3], a regularization algorithm closely related to the SVM which allows linear classifiers to be trained efficiently, with $O(n)$ dependance on the size of the data set. Using this approach, we are able to tractably train a system that achieves state-of-the-art performance on TIMIT phonetic classification. Furthermore, because our system relies only on decision boundaries and not on posterior probabilities, it is much more robust than a GMM-based system under additive noise.

We now outline the remainder of the paper. Section 2 describes our learning architecture: Section 2.1 develops the RLS algorithm in detail, including the critical issue of parameter tuning, Section 2.2 describes our method for constructing nonlinear second-order features, and Section 2.3 explains how we combine a collection of binary classifiers to make a multiclass classifier. Section 3 describes experiments on the TIMIT classification task, where we compare to both a state-of-the-art GMM system and other more recent systems on clean data, and to the GMM on noisy data, showing our system gives excellent performance in all cases. Section 4 places these results in context, suggests future directions, and compares and contrasts our approach to another recent work with similar inspirations, the large-margin Gaussian mixture model [4].

2. LEARNING ARCHITECTURE

2.1. Regularized Linear Least Squares

We are given a data set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. The x_i represent points to be classified, while

[*]Honda Research Institute USA, Inc., One Cambridge Center Suite 401, Cambridge, MA 02142

[†]MIT CSAIL, 32 Vassar Street, Cambridge, MA 02139

[‡]The American University of Beirut, P.O. Box 11-3730, Riad El Solh, Beirut, 1107 2020, Lebanon

[§]MIT Center For Biological and Computational Learning, Bldg. 46-5155, 77 Massachusetts Avenue, Cambridge, MA 02139 the y_i are the desired *labels*. The data points are collected in a matrix $X \in \mathbb{R}^{n \times d}$ (note that x_i^t is the ith *row* of X) and the labels are collected in $Y \in \mathbb{R}^n$. The linear regularized least squares problem is to find a vector $w \in \mathbb{R}^d$ minimizing

$$\frac{1}{2}||Y - Xw||_2^2 + \frac{\lambda}{2}||w||_2^2,$$

where λ is a positive regularization parameter controlling the tradeoff between fitting the observations and finding a w with small norm (i.e., a "smooth" w), and $||v||_2 = \sqrt{v^t v}$. This is a differentiable, convex optimization problem, and straightforward calculus and linear algebra [3] shows that the optimal w is given by

$$w = (X^t X + \lambda I)^{-1} X^t Y.$$

The regularization parameter λ must be chosen by the user. Defining $w_{\setminus i}$ to be the hyperplane obtained when we remove the ith training point and train on the remaining $n-1$ points, it seems reasonable that a good value of λ is one that yields a small *leave-one-out error*

$$\sum_i (y_i - w_{\setminus i}^t x_i)^2.$$

Linear regularized least squares has the remarkable property that this leave-one-out (LOO) error can be computed quickly, as we now show. Define the vector $Y^i \in \mathbb{R}^n$ via

$$Y_j^i = \begin{cases} y_j & j \neq i \\ w_{\setminus i}^t \mathbf{x}_i & j = i \end{cases}$$

Then $w_{\setminus i}$ "solves" the RLS problem where Y is replaced with Y^i:

$$\begin{aligned} & \min_{w \in R^n} \frac{1}{2} \sum_j (Y_j^i - w^t x_j)^2 + \lambda ||w||_2^2 \\ \geq \quad & \min_{w \in R^n} \frac{1}{2} \sum_{j \neq i} (Y_j^i - w^t x_j)^2 + \lambda ||w||_2^2 \\ = \quad & \frac{1}{2} \sum_{j \neq i} (Y_j^i - w_{\setminus i}^t x_j)^2 + \lambda ||w_{\setminus i}||_2^2 \\ = \quad & \frac{1}{2} \sum_j (Y_j^i - w_{\setminus i}^t x_j)^2 + \lambda ||w_{\setminus i}||_2^2, \end{aligned}$$

where the first equality follows from the definition of $w_{\setminus i}$ and the final equality follows from the definiton of Y_i^i. We have therefore shown that

$$w_{\setminus i} = (X^t X + \lambda I)^{-1} X^t Y^i.$$

This formula seems useless, in that it tells us how to get $w_{\setminus i}$ (which we don't really care about) if we already know $w_{\setminus i}^t x_i$ (which is what we want), but in fact it can be used to solve directly for the desired quantity:

$$\begin{aligned} (w_{\setminus i} - w)^t x_i &= \left((X^t X + \lambda I)^{-1} X^t (Y^i - Y)\right)^t x_i \\ &= (Y^i - Y)^t X (X^t X + \lambda I)^{-1} x_i \\ &= (w_{\setminus i}^t x_i - y_i) x_i^t (X^t X + \lambda I)^{-1} x_i, \end{aligned}$$

which immediately implies

$$w_{\setminus i}^t x_i = \frac{w^t x_i - y_i x_i^t (X^t X + \lambda I)^{-1} x_i}{1 - x_i^t (X^t X + \lambda I)^{-1} x_i}. \qquad (1)$$

We see that we can easily obtain the entire vector of leave-one-out errors if we can compute the diagonal elements of the n by n matrix $X(X^t X + \lambda I)^{-1} X^t$.

Define $k = min(n, d)$. In time $O(ndk)$, we can compute the "economy-sized" singular value decomposition [5] $X = USV^t$, where $U \in \mathbb{R}^{n \times k}$, $S \in \mathbb{R}^{k \times k}$, $V \in \mathbb{R}^{d \times k}$, $U^t U = V^t V = I$, and S is a diagonal matrix with nonnegative singular values on its diagonal. In terms of the SVD,

$$\begin{aligned} X^t X + \lambda I &= VS^2 V^t + \lambda I \\ &= V(S^2 + \lambda I)V^t + \lambda(I - VV^t) \\ (X^t X + \lambda I)^{-1} X^t &= V(S^2 + \lambda I)^{-1} S U^t \\ w &= V(S^2 + \lambda I)^{-1} S U^t Y \\ X(X^t X + \lambda I)^{-1} X^t &= US^2 (S^2 + \lambda I)^{-1} U^t \\ \left(X(X^t X + \lambda I)^{-1} X^t\right)_{ii} &= \sum_{j=1}^k \frac{U_{ij}^2 S_{jj}^2}{S_{jj}^2 + \lambda} \end{aligned}$$

The matrix $S^2 + \lambda I$ is diagonal, so its inverse can be computed in linear time. For any λ, assuming we have precomputed the SVD and matrix quantities that do not depend on λ, we can use the above formulas[1] to compute w in $O(dk)$ time, Xw in $O(nd)$ time, the diagonal entries of $(X(X^t X + \lambda I)^{-1} X^t)$ in $O(nk)$ time, and finally the leave-one-out error in $O(n)$ time. We see that the total time required to evalute a λ is $O((n + d)k)$. Therefore, we can rapidly and effectively find a λ with small LOO error, with the total computation time being dominated by the time required to compute the SVD, assuming we try $\ll k$ different values of λ.

Linear regularized least squares is an instance of Tikhonov regularization [2, 7], a very general framework for learning that includes many common algorithms, including support vector machines. RLS is unique in that the regularization parameter can be "multiplexed" — we pay about the same computational costs to find a good λ as we do to solve the problem for a single λ. That RLS admits an explicit formula for the LOO values is fairly well-known in the statistics and machine learning communities, but methods for "multiplexing" λ via matrix decompositions such as the SVD do not seem to be well-known; further details are available in [6].

2.2. Second-Order Features

The linear RLS algorithm (obviously) produces linear classifiers, but we can choose the feature space in which the classifiers live. In this work, we start with a base set of features, and

[1]We don't actually use Equation 1 to compute the LOO error, preferring a more numerically stable formula (with the same computational costs) that can be derived with some additional effort; Equation 1 gives correct answers in exact arithmetic. See [6] for further details.

explicitly lift these features into a higher dimensional space by taking pairwise products of the features. Conceptually, if we start with an initial feature vector $x \in \mathbb{R}^d$, we first form $\hat{x} = [1\ x^t]^t$, and then take as our new features all entries on or above the diagonal of the outer product $\hat{x}\hat{x}^t$, reshaped into a vector in $\mathbb{R}^{d(d+1)/2}$. (In practice, we use an indexing scheme to avoid redundant computations).

2.3. Learning Architecture

Given a multiclass data set, we train a binary RLS classifier for each pair of classes (for c classes, we train $c(c-1)/2$ binary classifiers). For each pair of classes i and j ($i < j$), we train a binary classifier using the points in class i with $y = 1$ and the points in class j with $y = -1$. Given a new test point, we threshold the outputs of all the classifiers at 0, forcing each classifier to make a hard vote for one of the two classes it was trained on. We then classify the example into the class that received the most votes. In the case of ties, we restrict our attention to the k classes that received the most votes and recount votes from only the $k(k-1)/2$ classifiers on these classes (in the simple case of a two-way tie between classes i and j, we choose in accord with the i-vs-j classifier). If a tie remains after this restriction, we pick the class with the highest prior (most training examples) from among the classes receiving the most votes.

3. EXPERIMENTS

The described classification technique was applied to phonetic classification on the TIMIT corpus using the standard practices for this task [8, 4]. While 61-class classification is performed, the hypotheses are mapped to a smaller 39-class set before scoring. Glottal stops (q) are ignored. Training was done on the standard NIST training set (462 speakers, 3696 utterances, 140225 tokens), and results are shown on the standard core test set (24 speakers, 192 utterances, 7215 tokens). The 400 utterance development set was not utilized in the final version of our system. To test the classification performance in noise, pink noise from the NOISEX database [9] was added to the test set at four signal-to-noise ratios (SNRs): 0, 10, 20, and 30dB. All training was done on clean data.

The features used are the "S2" features from [8]. Short-time Fourier analysis is done with a 30ms Hamming window every 5ms. For each frame, we compute 12 Mel-frequency cepstral coefficients (MFCCs). To get a fixed-length feature vector for each phonetic segment, the MFCCs are averaged over five regions: the 30ms before and after the segment, and three regions within the segment (in 3-4-3 proportion). The log duration is also included, giving a total of 61 dimensions. These features are then whitened using a principle component analysis (PCA) matrix derived from the training set. We also include for comparison an RLS system that works directly with the S2 features rather than second-order features.

The baseline system is a regenerated version of the aggregated 4-fold Gaussian mixture model (GMM) used in [8], which used K-means intitialization and maximum–likelihood training. The model has a total of 8801 diagonal-covariance Gaussians for the 60 mixtures (61 classes minus glottal stop).

Classification results on the core test set are shown in Table 1. The RLS second-order system (RLS2) results in a 20.93% error rate in clean data, which is a 10.4% relative reduction in error rate over the baseline GMM. In moderate noise, the reduction compared to the GMM is even larger. We note that the RLS1 system, which builds all-pairs linear classifiers directly on the S2 features, performs worse than the GMM in clean data, but better under noisy conditions, indicating that the discriminative architecture has a strong tendency to produce more noise robust models.

	Clean	30dB	20dB	10dB	0dB
GMM	23.37	29.02	46.03	68.90	85.43
RLS1	25.60	28.21	41.34	63.12	80.03
RLS2	20.93	23.40	33.79	57.42	77.80
1 - RLS2/GMM	10.44	19.37	26.59	16.66	8.93

Table 1. Phonetic classification error rates on the TIMIT test set. Signal-to-noise ratios indicate levels of additive pink noise. All numbers are expresed as percentages.

Our results also compare favorably to recently reported results using hidden conditional random fields [10] and large-margin Gaussian mixture models [4], who reported error rates of 21.7% and 21.1%, respectively on the core test set. Finally, it is worth noting that not all GMMs are created equal — different feature sets, numbers of clusters, and training techniques all have an effect. For instance, the best GMM result reported by [10] is 25.9% error, compared to the 23.37% from our (aggregated) GMMs. We feel that the GMM baseline we report represents the state-of-the-art for GMM-based classifiers on these features.

4. DISCUSSION

One of the primary advantages of the RLS scheme is its ability to quickly and effectively set the regularization parameter. In the experiment described in Section 3, the optimal λ's returned by the classifiers ranged over 4 orders of magnitude. In an additional experiment, we forced all the classifiers to use the average λ value found, and performance declined to 21.6%, which is worse than the large-margin GMM.

Because all the training set sizes in the current work are moderate, we use the SVD to compute LOO errors on the second-order features. When the training sets become large enough that the second-order features for a pair of classes don't fit easily in memory, this strategy will not work. However, it is possible to use instead an eigendecomposition of the covariance matrix X^tX, which is only d by d and can easily be computed even when X does not fit in core [6]. This approach has the same asymptotic time complexity as the SVD approach (dominated by $O(nd^2)$ time to compute X^tX), but is much simpler to code (out-of-core SVD routines are not readily available). In this case, we can still find w rapidly as a function of λ, but can no longer find leave-one-out values, and we instead choose a λ that gives good performance on a held-out development set. Since our hypothesis was that the dataset was large, keeping a hold-out set is not too burdensome.

While the RLS system described provides a substantial gain in accuracy compared to our best GMM system, it is orders of magnitude slower to train, requiring overnight training on a network of workstations (it is comparable to the GMM at prediction time). The system is a proof-of-conept prototype, and there are a number of possibilities for speeding the system up. Selecting λ via an eigendecomposition of X^tX and a development set instead of an SVD of X and LOO error to select λ will be advantageous whenever the training set is much larger than the number of dimensions. With somewhat more effort, we could build a hierarchical all-pairs system with a top level that divides the phonemes into broad manner classes, reducing the number of classifiers by an order of magnitude.

Recently, Sha and Saul introduced large margin Gaussian mixture modelling for phonetic classification [4], achieving excellent performance (21.1% core test error for their best model). A key element of their approach is that they directly optimize ellipsoids to separate the classes. They argue that this has a computational advantage over nonlinear SVMs. Their optimization is performed over the semidefinite cone, requiring an iterative solver. For this reason they cannot quickly search regularization parameters to trade smoothness and fit, as we do here.[2] We agree with Sha and Saul that approaches that make use of the kernel trick will lead to computational difficulties for large data sets, and instead opt to build a second-order classifer by explicitly lifting our features into a second-order space. Conceptually, while the large margin GMM fits ellipsoids in the original space by learning a positive semidefinite matrix, we fit general quadratics in the original space by learning a linear function in the lifted space. As a consequence, our individual classifiers can be quickly solved via RLS and good regularization parameters found. In addition, our approach trivially decomposes onto a network of workstations, while the large-margin GMM is formulated as a single optimization problem over all the Gaussians simultaneously, which is more difficult to parallelize.

Another key contrast with the large margin GMM is that the large margin GMM fits a mixture of ellipsoids for each class, while we fit a single quadratic for each *pair* of classes. We are currently investigating models that combine some of the features of these approaches, fitting multiple quadratics for each pair of classes. In any case, the present work demonstrates that requiring ellipsoids is not strictly necessary, and that a simple RLS architecture that fits arbitrary quadratic functions can give excellent performance.

[2]The large margin GMM is a Tikhonov regularization algorithm, minimizing a sum of a data fitting term and a regularizer based on the trace of the ellipsoids. In their formulation they do not explicitly include a regularization parameter, implicitly fixing $\lambda = 1$.

5. REFERENCES

[1] Vladimir N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[2] G. Wahba, *Spline Models for Observational Data*, vol. 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, Society for Industrial & Applied Mathematics, 1990.

[3] R. M. Rifkin, *Everything Old Is New Again: A Fresh Look at Historical Approaches to Machine Learning*, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[4] F. Sha and L. K. Saul, “Large margin gaussian mixture modelling for phonetic classification and regression,” in *Proceedings of ICASSP*, 2006.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 3rd edition, 1996.

[6] R. Rifkin and R. Lippert, “What you should know about matrix decompositions and regularized least squares,” (2006, in preparation).

[7] T. Evgeniou, M. Pontil, and T. Poggio, “Regularization networks and support vector machines,” *Advances In Computational Mathematics*, vol. 13, no. 1, pp. 1–50, 2000.

[8] A. Halberstadt and J. Glass, “Heterogeneous measurements and multiple classifiers for speech recognition,” in *Proceedings of ICSLP*, 1998.

[9] A. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones, “The noisex-92 study on the effect of additive noise on automatic speech recognition,” Technical Report, DRA Speech Research Unit, 1992.

[10] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, “Hidden conditional random fields for phone classification,” in *Proceedings of Eurospeech*, 2005.