ISOLATED-WORD RECOGNITION WITH PENALIZED LOGISTIC REGRESSION MACHINES

Øystein Birkenes[ab], Tomoko Matsui[a], and Kunio Tanabe[c]

[a]The Institute of Statistical Mathematics, Tokyo, Japan
[b]Department of Electronics and Telecommunications, NTNU, Trondheim, Norway
[c]Waseda University, Tokyo, Japan
{birkenes,tmatsui}@ism.ac.jp, tanabe.kunio@waseda.jp

ABSTRACT

We propose a new approach to isolated-word speech recognition based on penalized logistic regression machines (PLRMs). With this approach we combine the hidden Markov model (HMM) with multiclass logistic regression resulting in a powerful speech recognizer which provides us with the posterior probability for each word. Experiments on the English E-set show significant improvements compared to conventional HMM-based speech recognition.

1. INTRODUCTION

Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ be a random pair drawn according to an unknown probability distribution $p(x, y)$. We consider $x \in \mathcal{X}$ to be a suitable representation of an observable speech signal, and $y \in \mathcal{Y}$ to be an unobservable label describing the linguistic content of x. For example, x can be a sequence of Mel-frequency cepstral coefficients (MFCC) and y can represent the word label of x. We are concerned with the problem of predicting the unknown label y from the observed feature representation x.

The plug-in maximum a posteriori (plug-in MAP) decision rule [1, 2] predicts the unknown label by

$$\hat{y} = \arg\max_y \hat{p}(x|y)\hat{p}(y), \tag{1}$$

where $\hat{p}(x|y)$ and $\hat{p}(y)$ are estimates of the class-conditional distribution $p(x|y)$ and the prior probability $p(y)$, respectively. In speech recognition, $\hat{p}(x|y)$ is typically a hidden Markov model (HMM) whose parameters are estimated using the maximum likelihood (ML) criterion. With this approach, the HMM parameters for a class y are optimized independently of the other classes so as to best describe the generative process of a sample x. As a consequence, due to incorrect model assumption for $p(x|y)$ and a limited amount of training data, the error rate of the plug-in MAP rule with ML training may be far from the optimal error rate.

In [3], the authors acknowledge the sub-optimality of the plug-in MAP rule with ML training and present the minimum classification error (MCE) method which optimizes the HMM parameters discriminatively in order to minimize a smoothed error count on the training set. Their method improves the error rate considerably compared to the plug-in MAP rule with ML training, but it has some difficulties in the specifications of a number of hyperparameters in the criterion function.

In this paper we propose a new method for isolated-word speech recognition. Our approach makes use of the penalized logistic regression machines (PLRMs) presented in [4, 5]. In particular, we use multiclass logistic regression to model the posterior distribution $p(y|x)$ and perform deterministic prediction by

$$\hat{y} = \arg\max_y \hat{p}(y|x), \tag{2}$$

where $\hat{p}(y|x)$ is a model for $p(y|x)$. Our chosen model has two sets of parameters; a set of HMM parameters and a set of weights used for multiclass logistic regression. The parameters are optimized by maximizing a penalized logistic regression likelihood. Unlike MCE, this criterion function depends only on one hyperparameter. In addition, we obtain the posterior probability for each word. These posterior probabilities could, for example, be used as confidence measures [6].

In the next section we present a PLRM that is tailored to the speech recognition problem. Then, in Sec. 3, we show some experimental results on the English E-set. Section 4 concludes the paper.

2. A PENALIZED LOGISTIC REGRESSION MACHINE FOR SPEECH RECOGNITION

The penalized logistic regression machine (PLRM) [4, 5] was originally designed to be used with vectorial inputs. Here we extend this framework and present a PLRM that allows variable-length sequences of feature vectors as input, as this is the common representation for speech signals.

For mathematical convenience, let the label $y \in \mathcal{Y}$ be represented using a one-of-K coding scheme, i.e., y is one of the vectors in the K-dimensional Euclidean basis $\mathcal{Y} = \{(1, 0, 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, (0, 0, 0, \ldots, 1)\}$ with $K = |\mathcal{Y}|$ denoting the number of words in the vocabulary. Put differently, the kth word in the vocabulary is represented by a K-dimensional unit vector whose kth element is one.

A part of this work was supported by JSPS Postdoctoral Fellowship for Foreign Researchers and JSPS Grant-in-Aid for Scientific Research (B) 16300036 and (C) 16500092.

Then the probability distribution of the labels is the conditional multinomial distribution with parameters equal to the posterior probabilities $p_k = p(y_k = 1|x)$, for $k = 1, \ldots, K$.

In the following, we choose a parameterized model for the posterior probabilities p_k, choose a criterion function which is to be minimized for the purpose of obtaining estimates of the model parameters, and propose a numerical optimization algorithm for the minimization of the criterion function.

2.1. The Model

We use a parameterized model with parameter θ of the posterior probability p_k of the form

$$\hat{p}_k = \hat{p}_k(x;\theta) = \frac{\exp f_k(x;\theta)}{\sum_{l=1}^{K} \exp f_l(x;\theta)}, \tag{3}$$

where $f_k(x;\theta)$ is a discriminant function for the kth word. This is known as the softmax function and is widely used in multiclass logistic regression. For the discriminant functions we choose

$$f_k = f_k(x;\theta) = w_k^T \boldsymbol{\phi}(x;\Lambda), \tag{4}$$

where $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^M, x \mapsto \boldsymbol{\phi}(x;\Lambda)$ is a nonlinear map into M-dimensional space parameterized by Λ, and w_k are M-dimensional weight vectors. The parameters of the model are jointly denoted by $\theta = (W, \Lambda)$, where W is a $K \times M$-matrix whose rows are w_k^T.

The nonlinear map $\boldsymbol{\phi}$ should preserve discriminative information embedded in the speech signals. We propose to use a mapping involving a set of HMMs for the words, e.g.,

$$x \mapsto \boldsymbol{\phi}(x;\Lambda) = [1, \phi(x;\lambda_1), \ldots, \phi(x;\lambda_K)]^T \tag{5}$$

where $\phi(x;\lambda_k)$ is the log-likelihood of the HMM with parameter λ_k corresponding to the kth word, with λ_k being the kth column of the matrix Λ. Thus, if $L = |\lambda_k|$ denotes the number of parameters in each HMM, Λ is an $L \times K$ matrix of all the HMM parameters. The inclusion of 1 as the first element of $\boldsymbol{\phi}(x;\Lambda)$ ensures that the discriminant functions in (4) are affine transformations of the HMM log-likelihoods, i.e., that one of the terms in each scalar product is a bias term. To be more specific with our choice of nonlinear map, let $x = (x_1, \ldots, x_T)$ be a sequence of T feature vectors. Furthermore let $\lambda = (\pi, A, b)$ denote the parameters of an HMM; $\pi = [\pi_i]$ is the vector of initial state probabilities, $A = [a_{i,j}]$ is the transition probability matrix, and $b = \{b_i(x)\}$ is the collection of state-conditional pdfs. Then[1]

$$\phi(x;\lambda) = \log \max_q \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1},q_t} b_{q_t}(x_t). \tag{6}$$

where $q = (q_0, \ldots, q_T)$ is a state sequence.

Figure 1 illustrates our choice of model. For a sequence of feature vectors, x, we calculate the log-likelihood (6) for each word HMM, perform an affine transformation (4) of these likelihood values in order to get a discriminant function for each word, and finally map the discriminant functions to the posterior probabilities using the softmax function (3). These posterior probabilities can be used in deterministic prediction by selecting the word which has the highest probability.

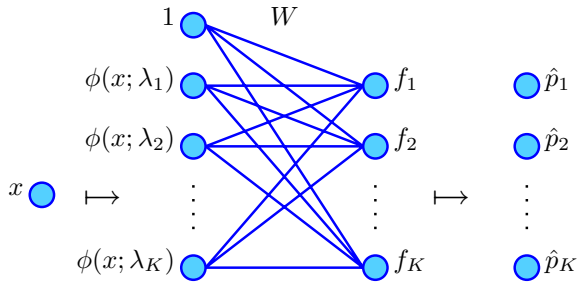

Fig. 1. The model of the posterior probabilities $\hat{p}_k$ for a word with feature representation x.

In the rest of this section we discuss how the parameter $\theta = (W, \Lambda)$ of the model can be estimated.

2.2. The Criterion Function

Given a set of training data $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$, the parameter θ could be estimated by maximizing the likelihood of the multinomial distribution which is

$$L(\theta;\mathcal{D}) = \prod_{n=1}^{N} \hat{p}_{y^{(n)}}, \tag{7}$$

where we let the label $y^{(n)}$ serve as an index. This means that $\hat{p}_{y^{(n)}}$ has the index corresponding to the index of the nonzero entry in the nth training label $y^{(n)}$.

However, due to the limited amount of training data, the maximum likelihood estimate is prone to overfitting, and may lead to poor prediction on unseen speech signals. For that reason, we introduce a penalty $\Omega(\theta)$ and find an estimate $\hat{\theta}$ by maximizing the penalized logistic regression likelihood

$$\mathcal{P}_\delta(\theta;\mathcal{D}) = L(\theta;\mathcal{D})\Omega^\delta(\theta), \tag{8}$$

where $\delta > 0$ is a hyperparameter used to balance the likelihood and the penalty factor.

It was proposed in [4] to use a penalty of the form

$$\Omega(\theta) = \exp\Big(-\frac{1}{2}\operatorname{trace}\Gamma W \Sigma W^T\Big), \tag{9}$$

where Γ is a $K \times K$ diagonal matrix whose kth diagonal element is the fraction of training samples with the kth class label, and $\Sigma = \Sigma(\Lambda)$ is an $M \times M$ positive definite matrix. The purpose of Γ is to compensate for possible differences in the number of training samples from each class. As for Σ, the simplest choice is $\Sigma = I$, where I denotes the identity matrix. Another choice is to let Σ be the sample moment matrix of the mapped speech data, that is, $\Sigma = 1/N \Phi\Phi^T$, where Φ is an $M \times N$ matrix with columns $\boldsymbol{\phi}(x^{(n)}, \Lambda)$.

[1] This is actually an approximation of the log-likelihood of an HMM. Nevertheless, since this is a common approximation in the speech literature, we refer to this simply as the log-likelihood.

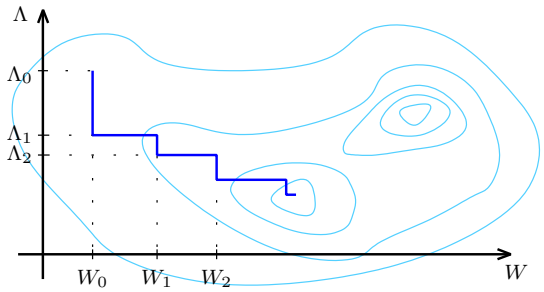

Fig. 2. The coordinate descent method.

With the penalty in (9), the negative logarithm of the penalized logistic regression likelihood is

$$\mathcal{P}_\delta^{\log}(\theta;\mathcal{D}) = -\sum_{n=1}^{N} \log \hat{p}_{y^{(n)}} + \frac{\delta}{2}\operatorname{trace}\Gamma W \Sigma W^T. \quad (10)$$

The next subsection concerns the minimization of the criterion function in (10) with respect to $\theta = (W, \Lambda)$.

2.3. Parameter Estimation

The function in (10) is convex with respect to W [4], but it is not guaranteed to be convex with respect to Λ. Therefore, the best we can hope for is to find a good local minimum. We propose to use a coordinate descent method to obtain a local minimum of $\mathcal{P}_\delta^{\log}(\theta;\mathcal{D}) = \mathcal{P}_\delta^{\log}(W,\Lambda;\mathcal{D})$, where the coordinates are W and Λ. The algorithm is initialized with Λ_0, for which a reasonable choice is the ML estimate of the HMM parameters. Then the initial weight matrix can be found as

$$W_0 = \min_W \mathcal{P}_\delta^{\log}(W, \Lambda_0; \mathcal{D}). \quad (11)$$

The iteration step is as follows:

$$\Lambda_{i+1} = \min_\Lambda \mathcal{P}_\delta^{\log}(W_i, \Lambda; \mathcal{D}), \quad (12a)$$

$$W_{i+1} = \min_W \mathcal{P}_\delta^{\log}(W, \Lambda_{i+1}; \mathcal{D}). \quad (12b)$$

The coordinate descent method is illustrated in Fig. 2.

In the following, we describe approaches to minimizing $\mathcal{P}_\delta^{\log}(W,\Lambda;\mathcal{D})$ with respect to W and Λ, which are needed in (12a) and (12b).

2.3.1. *Minimization with respect to W*

In [7], the author presents an efficient algorithm for the convex minimization of $\mathcal{P}_\delta^{\log}(W,\Lambda;\mathcal{D})$ with respect to W. The algorithm combines Newton's method with the conjugate gradient method. After choosing an initial weight matrix W^0, the iteration step in Newton's method is

$$W^{i+1} = W^i - \alpha \Delta W^i, \quad (\alpha > 0). \quad (13)$$

The update matrix ΔW^i is the solution to

$$\sum_{n=1}^{N} \left(\operatorname{diag}\hat{p}^{(n)} - \hat{p}^{(n)}\hat{p}^{(n)T}\right)\Delta W^i \phi^{(n)}\phi^{(n)T} + \delta\Gamma\Delta W^i\Sigma = (P - Y)\Phi^T + \delta\Gamma W^i\Sigma, \quad (14)$$

where $\hat{p}^{(n)}$ is a K-dimensional vector with elements $\hat{p}_k(x^{(n)};\theta)$, P is a $K \times N$ matrix whose columns are $\hat{p}^{(n)}$, and $\phi^{(n)} = \phi(x^{(n)};\Lambda)$ as given in (5). This equation can be solved by the conjugate gradient method. For more details see [7].

2.3.2. *Minimization with respect to Λ*

The HMM parameters in Λ have certain constraints on which values they can take on. For example, all variances must be greater than zero. In order to use an unconstrained optimization method such as the steepest descent method, we choose to first transform the parameters to a space where any value is valid. Then we can apply the steepest descent method with these transformed parameters, and finally transform the parameters back to the original space. We use the same parameter transformations as in [3], e.g., $\sigma \mapsto \tilde{\sigma} = \log\sigma$ and $\mu \mapsto \tilde{\mu} = \mu/\sigma$ for variances and means, respectively. Thus, after choosing an initial matrix Λ^0, we use a mapping $\Lambda^0 \mapsto \tilde{\Lambda}^0$ to the transformed feature space and iterate according to

$$\tilde{\Lambda}^{i+1} = \tilde{\Lambda}^i - \epsilon\nabla_{\tilde{\Lambda}}\mathcal{P}_\delta^{\log}(W,\tilde{\Lambda};\mathcal{D})\big|_{\tilde{\Lambda}=\tilde{\Lambda}^i}, \quad (\epsilon > 0). \quad (15)$$

Finally, the resulting matrix $\tilde{\Lambda}^*$ is mapped back to the original parameter space, $\tilde{\Lambda}^* \mapsto \Lambda^*$.

Straightforward, but tedious calculation gives the gradient needed in (15). We omit the expression for the gradient due to space limitations.

3. EXPERIMENTS

We did experiments on the E-set of the TI46 database. This set consists of 1433 training utterances and 2291 test utterances of one of the letters in the set {B,C,D,E,G,P,T,V,Z} spoken in isolation. From each speech signal we extracted a sequence of 39-dimensional feature vectors, including MFCC, delta and acceleration coefficients. We used a 25 ms Hamming window and a window shift of 10 ms. Each word was modeled by a 6-state left-to-right HMM with a Gaussian mixture model with 5 mixtures in each state.

We initially estimated the HMM parameters Λ_0 using standard ML estimation. Then we optimized W and Λ using the coordinate descent approach described in the previous section with stepsizes $\alpha = 1.0$ and $\epsilon = 0.1$, respectively. For each coordinate descent iteration (12a, 12b) we iterated once in the steepest descent method (15) and three times in the Newton method (13).

Table 1 shows the accuracy on the test set using PLRM with $\delta = 1000$ and 1000 coordinate descent iterations compared with the plug-in MAP rule with ML training and the MCE method in [3]. Deterministic prediction with PLRM outperforms both plug-in MAP with ML training and MCE.

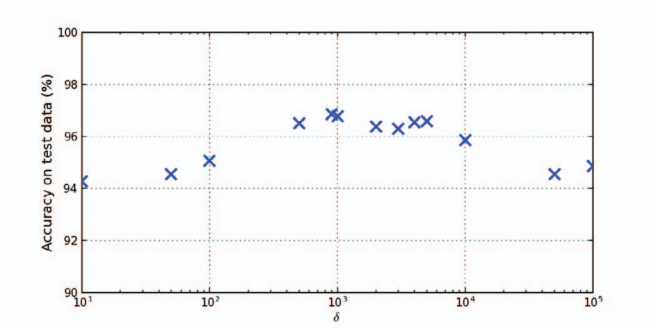

Fig. 3. Accuracy on the English E-set after 1000 iterations.

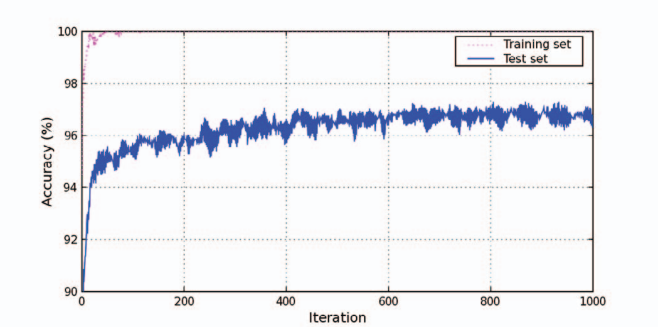

Fig. 4. Accuracy on the English E-set with $\delta = 1000$.

The error reduction rates are 72.6% and 36.0% for plug-in MAP with ML training and MCE, respectively.

Table 1. Recognition Accuracy (%)

Plug-in MAP (ML)	MCE	PLRM ($\delta = 1000$)
88.3	95.0	96.8

In Fig. 3 we have plotted the accuracy of PLRM after 1000 iterations for various values of δ in the range $[10, 10^5]$. We can see that values of δ around $\delta = 1000$ gives the best accuracy. Moreover, even the inferior accuracies shown considerably outperform the accuracy of the plug-in MAP rule and do nearly as well as the MCE approach.

Figure 4 shows the accuracy on both the training set and the test set as a function of number of iterations for $\delta = 1000$. The accuracy on the training set reaches near 100% after just a few iterations. The accuracy on the test set increases rapidly the first few iterations, and then increases more slowly as the number of iterations get larger.

With PLRM we obtain the probability for each word. In deterministic prediction the word with the highest posterior probability is chosen. A histogram of the highest posterior probabilities for all of the test utterances is shown in Fig. 5. The probabilities corresponding to incorrect decisions are displayed with a darker color than the probabilities corresponding to correct decisions. Note that the incorrect decisions have lower probability than the majority of the correct decisions. This means that these probabilities may serve as confidence measures or can be used in utterance verification.

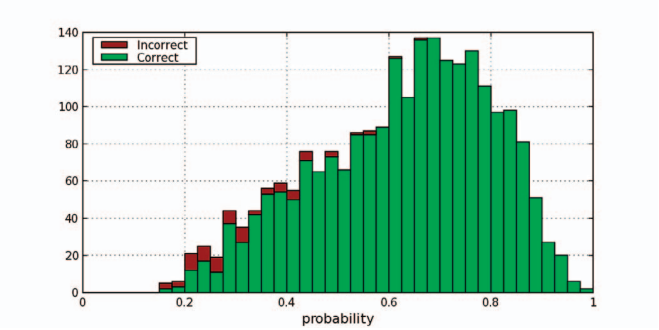

Fig. 5. Histogram of the highest posterior probability for all of the test utterances.

4. CONCLUSIONS AND FUTURE WORK

We have presented a new method for speech recognition based on penalized logistic regression machines (PLRMs). Experiments on the English E-set show the potential of this approach. Not only does PLRM achieve higher accuracy than the conventional plug-in MAP rule with ML training and MCE, but it also provides us with the posterior probability for each word. Posterior probabilities have many uses, including confidence measures and utterance verification. Also, the probabilities can be used for N-best rescoring of connected speech recognition. This is a topic for future research.

5. REFERENCES

[1] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, 1996.

[2] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

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

[4] K. Tanabe, "Penalized logistic regression machines: New methods for statistical prediction 1," *ISM Cooperative Research Report 143*, pp. 163–194, March 2001.

[5] K. Tanabe, "Penalized logistic regression machines: New methods for statistical prediction 2," in *Proc. IBIS*, Tokyo, Aug 2001, pp. 71–76.

[6] C.-H. Lee, "Statistical confidence measures and their applications," in *Proc. Int. Conf. on Speech Processing (ICSP)*, Dajeong, Korea, Aug 2001.

[7] K. Tanabe, "Penalized logistic regression machines and related linear numerical algebra," in *KOKYUROKU 1320, Institute for Mathematical Sciences*, Kyoto, 2003, pp. 239–250.