Maximum Margin Learning and Adaptation of MLP Classifers

Xiao Li, Jeff Bilmes and Jonathan Malkin

Department of Electrical Engineering
University of Washington, Seattle
{lixiao,bilmes,jsm}@ee.washington.edu

Abstract

Conventional MLP classifiers used in phonetic recognition and speech recognition may encounter local minima during training, and they often lack an intuitive and flexible adaptation approach. This paper presents a hybrid MLP-SVM classifier and its associated adaptation strategy, where the last layer of a conventional MLP is learned and adapted in the maximum separation margin sense. This structure also provides a support vector based adaptation mechanism which better interpolates between a speaker-independent model and speaker-dependent adaptation data. Preliminary experiments on vowel classification have shown promising results for both MLP learning and adaptation problems.

1. Introduction

Multilayer perceptron (MLP) classifiers have been popularly used in vowel classification and general phonetic recognition systems [1, 2, 3] because of their efficient discriminative training ability. They also have been integrated into HMMs to enhance speech recognition [4, 5]. The learning objective of an MLP classifier is usually minimum relative entropy. Ideally an MLP outputs the posterior probabilities of the classes given an observation, and this will naturally minimize classification errors. However, minimum relative entropy is too strict an objective to have an analytical solution. In practice, the optimization of this non-convex objective function is often achieved by back-propagation, which is not guaranteed to find a global optimum. Similarly, most of the existing solutions to MLP adaptation have the same objective as MLP learning, and a common adaptation strategy is either partially retraining network parameters, or adding augmentative, speaker-dependent neurons [4, 5, 6, 7, 8, 9].

In this work, we present an MLP classifier enhanced by support vector machines (SVM) [10]. The idea is to replace the hidden-to-output layer of an MLP by maximum margin hyperplanes. In fact this structure does not change the nature of an MLP; it is essentially an MLP learned with the maximum separation margin criterion. Maximum separation margin is a relatively relaxed objective compared to minimum relative entropy in the sense that it only intends to minimize classification errors instead of the divergence between two distributions. More importantly, this objective is guaranteed to converge to a unique optimal solution. Furthermore, we propose a support vector based adaptation strategy which offers an intuitive and flexible mechanism to balance the roles that the speaker-independent (SI) model and the adaptation data play in adaptation. A user adaptation scheme related to our work can be found in [11] for handwriting recognition. The difference is that [11] adopts an incremental learning approach to adaptation, while our approach attempts to minimize test errors on the adaptation data. Though we investigate the application of vowel classification in this work, our methods can be applied to general MLP classification and adaptation problems, e.g. to hybrid speech recognition systems [4, 5].

The rest of the paper is organized as follows: Section 2 and Section 3 discuss the learning and adaptation strategies respectively for the hybrid MLP-SVM classifier. Section 4 provides a background of our vowel classification application. Section 5 presents our preliminary evaluation, followed by conclusions in Section 6.

2. Hybrid MLP-SVM Classifier

The essential idea of a hybrid MLP-SVM classifier is to replace the hidden-to-output layer of an MLP by optimal margin hyperplanes [10]. We believe this hybrid MLP-SVM classifier is superior to pure MLPs in that it finds a unique solution to the last layer parameters via convex optimization with a primal-dual interpretation, and that it guarantees an upper bound on test errors [12]. Furthermore, this classifier can be implemented more efficiently than nonlinear SVMs trained in the input space. This is because a nonlinear SVM requires selecting and tuning a kernel to achieve a good nonlinear mapping from the input space to a transformed feature space where data are presumably linearly separable. In the case of a hybrid MLP-SVM classifier, this nonlinear mapping is implicitly optimized during the MLP training in the form of a sigmoid kernel.

Specifically, we first build up a simple MLP with one

This work was funded by NSF under Award ISS/ITR-0326382

hidden layer. The input layer consists of $D * W$ nodes, where D is the dimension of the feature vectors, and W is input window size. The hidden layer has N hidden nodes, and the output layer has K output nodes representing K class probabilities. The hidden-to-output layer weight vector and bias with respect to the k^{th} output are denoted as $\mathbf{w}_k$ and b_k. A sigmoid function is used as the nonlinear activation function at the hidden layer; and the output probabilities are normalized by a softmax function. At the stage of training, the network is optimized via back-propagation to minimize the relative entropy between the output distribution and the true label distribution. At classification time, the softmax function only serves as a normalizer, and the decision is essentially determined by a set of linear discriminant functions

$$d_k(\mathbf{h_t}) = \langle \mathbf{w}_k, \mathbf{h}_t \rangle + b_k, \tag{1}$$

where $\mathbf{h}_t$ is the hidden node vector of the t^{th} sample.

In the second training phase, we take as input the hidden node vectors computed from the training data using the optimized input-to-hidden layer parameters. We then train optimal margin hyperplanes, $\{\mathbf{w_k}, b_k\}$, for each class $k = 1..K$ on these inputs. Note that we use the SVM scheme of "one-versus-the-rest" [12] to deal with multiple classes for a better comparison with MLP classifiers. Also, the MLP labels {1,0} for a particular class are converted to {1,-1} to accommodate SVM formulas. In fact, the resulting classifier has exactly the same discrimination functions as in Equation (1). The only difference lies in the learning objective: among all the oriented hyperplanes for a specific binary classifier, there exists a unique optimal one which maximizes the margin between any training sample and the hyperplane. This optimal margin hyperplane can be found by solving the following constrained quadratic optimization problem [12] (here we consider only one binary classifier and drop the index k from our notation for simplicity),

$$\begin{array}{rl} \min\limits_{\mathbf{w},b,\xi} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t\in R^g}\xi_t \\ \text{subject to} & y_t(\langle \mathbf{w}, \mathbf{h_t}\rangle + b) + \xi_t - 1 \geq 0 \ \text{ and } \ \xi_t \geq 0, \end{array} \tag{2}$$

where R^g denotes the training set, ξ_t are slack variables introduced for non-separable data, and C penalizes those samples across the boundary. By introducing Lagrange multipliers α_t, we have the solution $\mathbf{w} = \sum_{t\in R^g}\alpha_t y_t \mathbf{h_t}$. The resulting hyperplane is determined by those samples with nonzero α_t values, known as support vectors (SV).

3. Adaptation Strategy

As mentioned in the introduction, nearly all the existing MLP adaptation algorithms achieve the tradeoff between the SI model and the adaptation data by partially retraining the original network or by training additional neurons [4, 5, 7, 6, 8, 9]. This strategy, however, has the following limitations: (a) Similiar to MLP learning, the optimization is not guaranteed to reach a global optimum; (b) The number of free parameters to estimate at the adaptation stage often relies on input, hidden and output layer dimensions; (c) The interpolation between the SI model and the adaptation data is not always intuitive and flexible.

In this work, we propose to update only the hidden-to-output layer at adaptation time in the maximum margin sense, while keeping the input-to-hidden layer intact. As mentioned in Section 2, the input-to-hidden layer acts as a nonlinear mapping from the original $D\cdot W$-dimensional input space to a new N-dimensional feature space, while the hidden-to-output layer simply acts as K binary linear classifiers in this transformed feature space. Fixing the input-to-hidden layer is akin to fixing the kernel, while retraining the hidden-to-output layer is equivalent to updating the SVs for a specific speaker.

Since only the SVs contribute to the decision boundary, training on the SVs only would give exactly the same hyperplane as training on the whole data set. This makes a SVM amenable to incremental learning [13] where only the SVs are preserved and are combined with new data in the next training epoch. The user adaptation problem has been tackled in the same fashion, where the SVs trained using user-independent data are combined with a subset of user-dependent data for adaptation. Examples of this can be found in the field of handwritten character recognition [11, 14]. The adaptation problem, however, is not entirely the same as the problem of incremental learning. The former aims to reduce the test error of user-dependent data, whereas the latter aims to reduce the test error of all data. Furthermore, the number of SVs obtained from the training set is often larger than that of the adaptation data. Therefore, it is not always effective to treat all the old SVs and the adaptation data equally or even to discard a portion of the adaptation data.

To solve the adaptation problem, we propose to weight the slack variables to make an error on the training data less costly than one on the adaptation data. Again we only consider one binary classifier. We define SV^g and $\mathbf{w}^g$ as the SVs and the corresponding hyperplane obtained from the SI data R^g; and SV^a and $\mathbf{w}^a$ as those obtained from the adaptation data R^a. Note that $SV^g \bigcap R^a = \emptyset$. We then modify the objective function in Equation (2) as

$$\begin{array}{rl} \min\limits_{\mathbf{w},b,\xi} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t\in SV^g} p_t\xi_t + C\sum_{t\in R^a}\xi_t \\ \text{subject to} & y_t(\langle \mathbf{w}, \mathbf{h_t}\rangle + b) + \xi_t - 1 \geq 0 \ \text{ and} \\ & \xi_t \geq 0, \ \ t \in SV^g \bigcup R^a \end{array} \tag{3}$$

In this way, we can adjust how important the role that the SI data plays in the adapted classifier. In an extreme case, where $p_t = 1, \forall t \in SV^g$, the above objective is equivalent to training a SVM using all old SVs and all adaptation data. At the other extreme, where $p_t = 0, \forall t \in SV^g$, the adaptation leads to a completely new SVM trained using only the adaptation data. Between these two ex-

tremes, we would like to weight each sample in SV^g by how likely it is to be generated from the adaptation data distribution. Specifically

$$p_t = g(\frac{1}{s}(\langle \mathbf{w}^{\mathbf{a}}, \mathbf{h}_t \rangle + b^a)), \quad (4)$$

where $s = \sum_{t \in R^a} |\alpha_t|$ is a scalar for normalization, and g can be a monotonically increasing function converting a real number to a probability. In this work, we use an indicator function $g(x) = \delta(x > d)$ for efficiency, where d is a constant controlling the amount of SI data information incorporated in adaptation. All SV^g are selected when $d = -\infty$, while none of them are selected when $d = +\infty$.

Finally, it is worth noting that this idea can be considered analogous to [9]. The adaptation scheme presented in [9] aims to minimize the relative entropy. It interpolates the SI model and the adaptation data by retraining only the most "active" hidden neurons, i.e. those with the maximum variance over the adaptation data. In our work, instead, we use the maximum margin objective for adaptation and achieve interpolation by combining part of the SI support vectors with the adaptation data.

4. Application and Database

The *Vocal Joystick* (VJ) project, conducted at the University of Washington, is intended to assist individuals with motor impairments in human-machine interaction using non-verbal vocal parameters. The VJ system offers control mechanisms for both continuous and discrete tasks. In an exemplary application of cursor control, vowel quality is utilized to control the direction of cursor movement, while voice intensity and pitch are used to determine speed and acceleration. Using such an interface, the computer does not need to wait for a complete command to actuate an action, but rather continuously listens to a user and maps his voice into cursor movement.

The VJ system performs frame-by-frame vowel classification in order to capture the vocal tract shift in real-time. Since the vowels pronounced in the VJ framework may have a huge dynamic range in both intensity and pitch values, a reliable frame-level vowel classifier robust to energy and pitch variations is indispensable. Furthermore, this classifier should be amenable to adaptation to further improve accuracy, since a vowel class articulated from one speaker might overlap, in acoustic space, with a different vowel class from another speaker. Therefore, our proposed MLP-SVM classifier and its adaptation algorithm can be well applied to this problem.

We have collected a data set of constant-vowel utterances, consisting of 8 vowels whose IPA symbols and articulatory gestures are depicted in Figure 1. This data set so far contains utterances from 15 speakers, but is expected to have many more speakers eventually. Each speaker articulated each vowel with all combinations of the following configurations: (a) long/short (duration); (b) falling/level/rising (pitch); (c) loud/normal/quiet (intensity). There are $2 \times 3 \times 3 = 18$ utterances for each vowel from a single speaker.

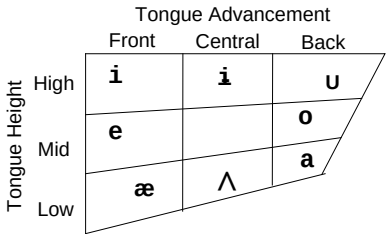

Figure 1: Vowel set

The recordings of 10 speakers were allocated to the training set, while those of the remaining 5 speakers were used for adaptation and evaluation. There were 180 utterances (approximately 22,000 sample frames) for *each* vowel class used for SI training. For a particular test speaker, the 18 utterances for each vowel class were further divided into 6 subsets with 3 utterances each. Each subset was used for adaptation and the remaining subsets for evaluation. There were 3/15 utterances (approximately 360/1,840 sample frames) for *each* vowel in the adaptation/evaluation subsets respectively. We calculated the mean of the error rates over these 6 subsets, and we repeated this for each speaker. The final classification error rate was an average over the 5 test speakers, and hence essentially an average of 30 subsets.

5. Experiments and Results

Using these data, we conducted two sets of experiments. The first one only included 4 vowel classes, /æ/, /a/, /u/ and /i/, while the second added the other 4 vowels leading to 8 classes. For both experiments, we evaluated SI and adapted classifiers on the same evaluation subsets. We varied the amount of adaptation data by choosing either 1, 2 or 3 utterances in each adaptation subset, corresponding to 1.2s, 1.8 and 3.6 seconds, on average.

To construct the speaker-independent hybrid MLP-SVM classifier, we first built a two-layer perceptron. The input layer consists of W=7 frames of MFCCs and their deltas (with mean subtraction and variance normalization), leading to 182 dimensions. The hidden layer has N=50 nodes. The W and N values were empirically found the best for our task. The output layer has either 4 or 8 nodes, corresponding to the 4 or 8 vowel classes. As proposed, we replaced the hidden-to-output layer by optimal margin hyperplanes trained by SVMTorch [15], where C=10 in the 4-class case and C=1 for the 8-class case. For comparison, we also built up a GMM classifier with 16 mixtures (empirically the best) per vowel class. Table 1 summarizes the average error rates using these SI classifiers. Our proposed hybrid classifier obtained the lowest error rate for both experiments. It improved over

	4-class	8-class
GMM	14.87%	44.12%
MLP	10.81%	39.95%
MLP-SVM	**9.30**%	**37.05**%

Table 1: Avg.error rates of SI classifiers

4-class	1.2s	1.8s	3.6s
GMM+MLLR	12.10%	10.35%	9.27%
MLP	10.25%	9.33%	8.34%
MLP-SVM	**8.59%**	**8.21%**	**7.37%**

Table 2: Avg. error rates of adapted 4-class classifiers

the pure MLP by a relative 13.9% error rate reduction in the 4-class case and 7.3% in the 8-class case.

Finally, we conducted adaptation experiments using the method proposed in Section 3. The SVs of the training set, again denoted as SV^g, were combined with the adaptation data to update the optimal margin hyperplanes. In the 4-class case, the lowest error rate was obtained when $d = -\infty$, meaning all SV^g samples were used in adaptation. In the 8-class case, the best performance was achieved when about 50% of the SV^g samples were used. It is interesting to notice that the old SVs were not always helpful in adaptation. For comparison, we adapted the GMM classifier by maximum likelihood linear regression (MLLR), and adapted the MLP classifier by further training the hidden-to-output layer under the minimum relative entropy criterion (this was done on all hidden neurons since we only had 50 hidden nodes). Table 2 and Table 3 shows the lowest average error rates we obtained using each classifier when the adaptation data was 1.2, 1.8 and 3.6 seconds respectively. The support vector based adaptation consistently achieved the best performance when the adaptation data is in a small amount, while the simple MLP adaptation approach became superior when more adaptation data was available in the 8-class case.

6. Conclusions

The hybrid MLP-SVM classifier presented in this work combines the MLP's ability to model nonlinearity with the SVM's superior classification power. Furthermore, the idea of weighting the slack variables in learning the optimal margin hyperplanes offers an intuitive and flexible adaptation mechanism. In a preliminary experiment for an in-house application, the hybrid MLP-SVM classifier outperformed conventional MLP classifiers for SI classification problems. Its associated adaptation strategy worked remarkably well by using only a small amount of adaptation data. When more adaptation data was available, a simple MLP adaptation scheme became the best in the 8-class problem. The authors would like thank Richard Wright, Kelley Kilanski, Andrea Macleod and Scott Drellishak for VJ data colleciton.

8-class	1.2s	1.8s	3.6s
GMM+MLLR	36.64%	33.55%	32.48%
MLP	31.82%	28.37%	**25.48%**
MLP-SVM	**28.37%**	**28.20%**	27.27%

Table 3: Avg. error rates of adapted 8-class classifiers

7. References

[1] H.Leung and V.Zue, "Phonetic classification using multilayer perceptions," in *ICASSP*, 1990.

[2] S.A.Zahorian and Z.B.Nossair, "A partitioned neural network approach for vowel classification using smoothed time/frequency features," *IEEE Trans. on Speech and Audio Processing*, 1999.

[3] P.Schmid and E.Barnard, "Explicit, n-best formant features for vowel classification," in *ICASSP*, 1997.

[4] H.Bourlard and N.Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[5] A.J.Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Trans. on Neural Networks*, 1994.

[6] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *eurospeech*, 1995.

[7] J.Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," in *Proc. Eurospeech*, 1995.

[8] N.Strom, "Speaker adaptation by modeling the speaker variation in a continuous speech recognition system," in *ICSLP*, 1996.

[9] J.Stadermann and G.Rigoll, "Two-stage speaker adaptation of hybrid tied-posterior acoustic models," in *ICASSP*, 2005.

[10] V.Vapnik, *The Nature of Statistical Learning Theory, Chapter 5*. Springer-Verlag, New York, 1995.

[11] N. Matic, I. Guyon, J. Denker, and V. Vapnik, "Writer adaptation for on-line handwritten character recognition," in *Proc. Intl. Conf. on Pattern Recognition and Document Analysis*, 1993.

[12] B.Scholkopf and A.J.Smola, *Learning with kernels*. The MIT Press, 2001.

[13] N.Syed, H.Liu, and K.Sung, "Incremental learning with support vector machines," in *Proc. Workshop on Support Vector Machines at the Intl. Joint Conf. on Aritifical Intelligence*, 1999.

[14] B.-B.Peng, Z.-X.Sun, and X.-G. Xu, "SVM-based incremental active learning for user adaptation for online graphics recognition system," in *Proc.Intl.Conf.on Machine Learning and Cybernetics*, 2002.

[15] R.Collobert and S.Bengio, "SVMTorch: support vector machines for large-scale regression problems," *The journal of Machine Learning Research*, 2001.