

Automatic Speech Recognition via N-Best Rescoring using Logistic Regression

Øystein Birkenes¹, Tomoko Matsui²,

Kunio Tanabe³ and Tor André Myrvoll¹

¹*Norwegian University of Science and Technology (NTNU), Trondheim,*

²*The Institute of Statistical Mathematics, Tokyo,*

³*Waseda University, Tokyo,*

¹*Norway*

^{2,3}*Japan*

1. Introduction

Automatic speech recognition is often formulated as a statistical pattern classification problem. Based on the optimal Bayes rule, two general approaches to classification exist; the generative approach and the discriminative approach. For more than two decades, generative classification with hidden Markov models (HMMs) has been the dominating approach for speech recognition (Rabiner, 1989). At the same time, powerful discriminative classifiers like support vector machines (Vapnik, 1995) and artificial neural networks (Bishop, 1995) have been introduced in the statistics and the machine learning literature. Despite immediate success in many pattern classification tasks, discriminative classifiers have only achieved limited success in speech recognition (Zahorian et al., 1997; Clarkson & Moreno, 1999). Two of the difficulties encountered are 1) speech signals have varying durations, whereas the majority of discriminative classifiers operate on fixed-dimensional vectors, and 2) the goal in speech recognition is to predict a sequence of labels (e.g., a digit string or a phoneme string) from a sequence of feature vectors without knowing the segment boundaries for the labels. On the contrary, most discriminative classifiers are designed to predict only a single class label for a given feature.

In this chapter, we present a discriminative approach to speech recognition that can cope with both of the abovementioned difficulties. Prediction of a class label from a given speech segment (speech classification) is done using logistic regression incorporating a mapping from varying length speech segments into a vector of regressors. The mapping is general in that it can include any kind of segment-based information. In particular, mappings involving HMM log-likelihoods have been found to be powerful.

Continuous speech recognition, where the goal is to predict a sequence of labels, is done with N-best rescoring as follows. For a given spoken utterance, a set of HMMs is used to generate an N-best list of competing sentence hypotheses. For each sentence hypothesis, the probability of each segment is found with logistic regression as outlined above. The segment probabilities for a sentence hypothesis are then combined along with a language model score in order to get a new score for the sentence hypothesis. Finally, the N-best list is reordered based on the new scores.

The chapter is organized as follows. In the next section, we introduce some notation and present logistic regression in a general pattern classification framework. Then, we show how logistic regression can be used for speech classification, followed by the use of logistic regression for continuous speech recognition with N-best rescoring. Finally, we present experimental results on a connected digit recognition task before we give a short summary and state the conclusions.

2. Pattern classification and logistic regression

In pattern classification, we are interested in finding a decision rule h , which is a mapping from the set of observations \mathcal{X} to the set of labels \mathcal{Y} . Depending on the application, an observation $x \in \mathcal{X}$ can be a vector of features, or it can have a more complex form like a sequence of feature vectors. The latter is the most common way of representing a speech segment (Rabiner, 1989). A label y is usually denoted as a natural number in the finite set $\mathcal{Y} \in \{1, \dots, K\}$ of class labels. In speech classification, for example, there are typically $K = 39$ class labels representing phonemes.

If the joint probability distribution $p(x, y)$ of observations and labels were known, the optimal decision rule would be the Bayes decision rule (Berger, 1985), which is

$$\begin{aligned} \hat{y} &= \arg \max_{y \in \mathcal{Y}} p(y | x) \\ &= \arg \max_{y \in \mathcal{Y}} p(x | y) p(y). \end{aligned} \quad (1)$$

In practical applications, however, we usually do not know any of the above probability distributions. One way to proceed is to estimate the distributions from a set $\mathcal{D} = \{(x_1, y_1), \dots, (x_L, y_L)\}$ of samples referred to as training data. Bayes decision rule can then be approximated in two ways. The first way is to estimate the two distributions $p(x | y)$ and $p(y)$, and substitute these into the second line in (1), an approach called the generative approach. The second way is to estimate $p(y | x)$, and substitute this into the first line in (1), an approach called the discriminative approach.

Logistic regression is a statistically well-founded discriminative approach to classification. The conditional probability of a class label given an observation is modeled with the multivariate logistic transform, or softmax function, defined as (Tanabe, 2001a,b)

$$\hat{p}(y = k | x, W, \Lambda) = \frac{e^{f_k(x, W, \Lambda)}}{\sum_{i=1}^K e^{f_i(x, W, \Lambda)}}. \quad (2)$$

In the above equation, f_i is a linear combination (plus a bias term) of M regressors $\phi_1(x, \lambda_1), \dots, \phi_M(x, \lambda_M)$, with hyperparameters $\Lambda = \{\lambda_1, \dots, \lambda_M\}$, i.e.,

$$\begin{aligned} f_i(x, W, \Lambda) &= w_{0i} + w_{1i} \phi_1(x, \lambda_1) + \dots + w_{Mi} \phi_M(x, \lambda_M) \\ &= w_i^T \phi(x, \Lambda), \end{aligned} \quad (3)$$

with $\phi(x, \Lambda) = [1, \phi_1(x, \lambda_1), \dots, \phi_M(x, \lambda_M)]^T$ and $w_i = [w_{0i}, \dots, w_{Mi}]^T$. The parameters of the model are the elements of the $(M + 1) \times K$ dimensional weight matrix

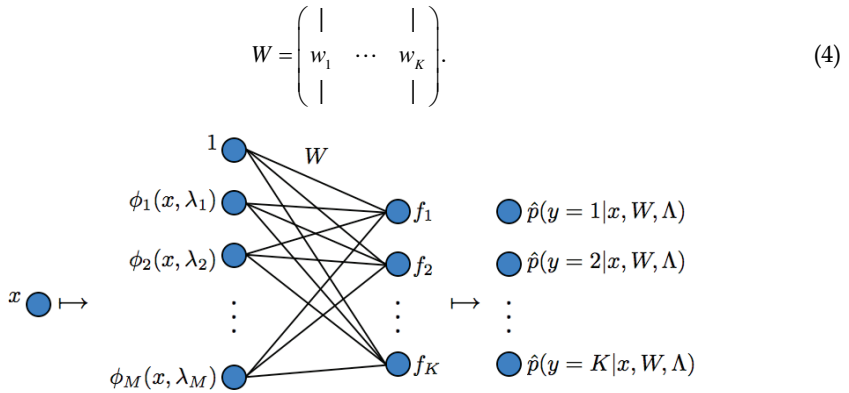


Fig. 1. The logistic regression model.

Due to the probability constraint $\sum_{k=1}^K \hat{p}(y = k | x, W, \Lambda) = 1$, the weight vector for one of the classes, say w_k , need not be estimated and can be set to all zeros. Here however, we follow the convention in (Tanabe, 2001a,b) and keep the redundant representation with K non-zero weight vectors. As explained in (Tanabe, 2001a,b), this is done for numerical stability reasons, and in order to treat all the classes equally.

We can think of the model for the conditional probability of each class k given an observation x as a series of transforms of x as illustrated in Fig. 1. First, x is transformed into a vector $\phi(x, \Lambda)$ of M regressors augmented with a "1". Then a linear transform $f = W^T \phi(x, \Lambda)$ gives the elements of the K -dimensional vector f , which are subsequently used in the multivariate logistic transform in order to obtain the conditional probabilities $\hat{p}(y = k | x, W, \Lambda)$.

The classical way to estimate W from a set of training data \mathcal{D} is to maximize the likelihood, or equivalently, minimize the negative log-likelihood

$$l(W; \mathcal{D}) = - \sum_{l=1}^L \log \hat{p}(y = y_l | x_l, W, \Lambda). \quad (5)$$

However, the maximum likelihood estimate does not always exist (Albert & Anderson, 1984). This happens, for example, when the mapped data set $\{(\phi(x_1; \Lambda), y_1), \dots, (\phi(x_L; \Lambda), y_L)\}$ is linearly separable. Moreover, even though the maximum likelihood estimate exists, overfitting to the training data may occur, which in turn leads to poor generalization performance. For this reason, we introduce a penalty on the weights and find an estimate \hat{W} by minimizing the penalized negative log-likelihood (Tanabe, 2001a,b)

$$pl_\delta(W; \mathcal{D}) = - \sum_{l=1}^L \log \hat{p}(y = y_l | x_l, W, \Lambda) + \frac{\delta}{2} \text{trace} \Gamma W^T \Sigma W, \quad (6)$$

where $\delta \geq 0$ is a hyperparameter used to balance the likelihood and the penalty factor. The $K \times K$ diagonal matrix Γ compensates for differences in the number of training examples from each class, as well as include prior probabilities for the various classes. If we let L_k

denote the number of training examples from class k , and $\hat{p}(y = k)$ denote our belief in the prior probability for class k , we let the k th element of Γ be

$$\gamma_k = \frac{L_k}{L\hat{p}(y = k)}. \quad (7)$$

The $(M + 1) \times (M + 1)$ matrix Σ is the sample moment matrix of the transformed observations $\phi(x_i; \Lambda)$ for $l = 1, \dots, L$, that is,

$$\Sigma = \Sigma(\Lambda) = \frac{1}{L} \sum_{l=1}^L \phi(x_l; \Lambda) \phi^T(x_l; \Lambda). \quad (8)$$

It can be shown (Tanabe, 2001a) that $pl_\delta(W; \mathcal{D})$ is a matrix convex function with a unique minimizer W^* . There is no closed-form expression for W^* , but an efficient numerical method of obtaining an estimate was introduced in (Tanabe, 2001a,b, 2003). In this algorithm, which is called the penalized logistic regression machine (PLRM), the weight matrix is updated iteratively using a modified Newton's method with stepsize α_i , where each step is

$$W_{i+1} = W_i - \alpha_i \Delta W_i, \quad (9)$$

where ΔW_i is computed using conjugate gradient (CG) methods (Hestenes & Stiefel, 1952; Tanabe, 1977) by solving the equation (Tanabe, 2001a,b)

$$\sum_{l=1}^L \phi_l \phi_l^T \Delta W_i (\text{diag } p_l - p_l p_l^T) + \delta \Sigma \Delta W_i \Gamma = \Phi (P^T(W_i) - Y^T) + \delta \Sigma W_i \Gamma. \quad (10)$$

In the above equation, Φ is the $(M + 1) \times L$ matrix whose l th column is $\phi_l = \phi(x_l; \Lambda)$, $P(W)$ is a $K \times L$ matrix whose l th column is $p_l = [\hat{p}(y = 1 | x_l, W, \Lambda), \dots, \hat{p}(y = K | x_l, W, \Lambda)]^T$, and Y is a $K \times L$ matrix where the l th column is a unit vector with all zeros except y_l which is 1.

2.1 Adaptive regressor parameters

Additional discriminative power can be obtained by treating Λ as a set of free parameters of the logistic regression model instead of a preset fixed set of hyperparameters (Birkenes et al., 2006a). In this setting, the criterion function can be written

$$pl_\delta(W, \Lambda; \mathcal{D}) = - \sum_{l=1}^L \log \hat{p}(y = y_l | x_l, W, \Lambda) + \frac{\delta}{2} \text{trace} \Gamma W^T \Sigma(\Lambda) W, \quad (11)$$

which is the same as the criterion in (6), but with the dependency on Λ shown explicitly. The goal of parameter estimation is now to find the pair (W^*, Λ^*) that minimizes the criterion in (11). This can be written mathematically as

$$(W^*, \Lambda^*) = \arg \min_{(W, \Lambda)} pl_\delta(W, \Lambda; \mathcal{D}). \quad (12)$$

As already mentioned, the function in (11) is convex with respect to W if Λ is held fixed. It is not guaranteed, however, that it is convex with respect to Λ if W is held fixed. Therefore, the best we can hope for is to find a local minimum that gives good classification performance.

A local minimum can be obtained by using a coordinate descent approach with coordinates W and Λ . The algorithm is initialized with Λ_0 . Then the initial weight matrix is found as

$$W_0 = \arg \min_W p l_\delta(W, \Lambda_0; \mathcal{D}). \quad (13)$$

The iteration step is as follows:

$$\begin{aligned} \Lambda_{i+1} &= \arg \min_\Lambda p l_\delta(W_i, \Lambda; \mathcal{D}) \\ W_{i+1} &= \arg \min_W p l_\delta(W, \Lambda_{i+1}; \mathcal{D}). \end{aligned} \quad (14)$$

The coordinate descent method is illustrated in Fig. 2.

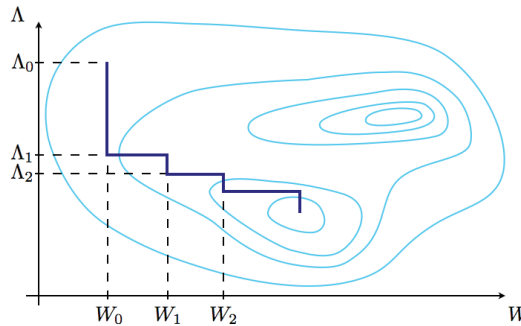


Fig. 2. The coordinate descent method used to find the pair (W^*, Λ^*) that minimizes the criterion function $p l_\delta(W, \Lambda; \mathcal{D})$.

For the convex minimization with respect to W , we can use the penalized logistic regression machine (Tanabe, 2001a,b). As for the minimization with respect to Λ , there are many possibilities, one of which is the RProp method (Riedmiller and Braun, 1993). In this method, the partial derivatives of the criterion with respect to the elements of Λ are needed. These calculations are straightforward, but tedious. The interested reader is referred to (Birkenes, 2007) for further details.

When the criterion function in (11) is optimized with respect to both W and Λ , overfitting of Λ to the training data may occur. This typically happens when the number of free parameters in the regressor functions is large compared to the available training data. By keeping the number of free parameters in accordance with the number of training examples, the effect of overfitting may be reduced.

2.2 Garbage class

In some applications, the classifier will be presented with observations x that do not correspond to any of the classes in the label set \mathcal{Y} . In this situation, the classifier should return a small probability for every class in \mathcal{Y} . However, this is made impossible by the fact

that the total probability should sum to 1, that is, $\sum_{y \in \mathcal{Y}} p(y|x) = 1$. The solution to this problem is to introduce a new class $y = K + 1 \in \mathcal{Y}_0 = \mathcal{Y} \cup \{K + 1\}$, called a garbage class, that should get high conditional probability given observations that are unlikely for the classes in \mathcal{Y} , and small probability otherwise (Birkenes et al., 2007).

In order to train the parameters of the logistic regression model with such a garbage class, a set of observations labeled with a garbage label, or garbage observations, are needed. For applications with a low-dimensional observation set \mathcal{X} , these garbage observations can be drawn from a uniform distribution over \mathcal{X} . For many practical applications however, \mathcal{X} has a very high dimensionality, so an unreasonably high number of samples must be drawn from the uniform distribution in order to achieve good performance. In such cases, prior knowledge of the nature or the generation of the possible garbage observations that the classifier will see during prediction is of great value. We will soon see how we can use N-best lists to generate garbage observations for continuous speech recognition.

3. Classification of speech segments with logistic regression

In this section we will be concerned with the modeling of the conditional distribution $p(y|x)$ using the logistic regression model, where each observation $x = (o_1, \dots, o_{T_x})$ is a sequence of feature vectors extracted from a speech segment and y is a word label. Since the observation x is here a sequence of feature vectors that can vary in length, the logistic regression mapping $\phi: \mathcal{X} \rightarrow \mathbb{R}^{M+1}$ is a map from the set \mathcal{X} of all such observations x into the Euclidean space \mathbb{R}^{M+1} containing all regressor vectors $\phi(x; \Lambda)$. The mapping should be able to map observations of varying lengths into fixed dimensional vectors while preserving the discriminative information embedded in the observations.

A mapping that has been found to be effective for speech classification makes use of $M = K$ hidden Markov models (HMMs), one for each word in the vocabulary, and is defined as (Birkenes et al., 2006a)

$$\phi(x; \Lambda) = \begin{pmatrix} 1 \\ \frac{1}{T_x} \log \hat{p}(x; \lambda_1) \\ \vdots \\ \frac{1}{T_x} \log \hat{p}(x; \lambda_M) \end{pmatrix}, \quad (15)$$

where $\hat{p}(x; \lambda_m)$ is the Viterbi-approximated likelihood (i.e., the likelihood computed along the Viterbi path) of the m th HMM with parameter vector λ_m . Specifically, if we let $\lambda = (\pi, A, \eta)$ be the set of parameters for an HMM, where π denotes the initial state probabilities, A is the transition matrix, and η is the set of parameters of the state-conditional probability density functions, then

$$\begin{aligned} \hat{p}(x; \lambda) &= \max_q \hat{p}(x, q; \lambda) \\ &= \max_q \pi_{q_1} \prod_{t=2}^{T_x} a_{q_{t-1}, q_t} \prod_{t=1}^{T_x} \hat{p}(o_t | q_t; \eta_{q_t}), \end{aligned} \quad (16)$$

where $q = (q_1, \dots, q_{T_x})$ denotes a state sequence. Each state-conditional probability density function is a Gaussian mixture model (GMM) with a diagonal covariance matrix, i.e.,

$$\begin{aligned} \hat{p}(o | q; \eta_q) &= \sum_{h=1}^H c_{qh} \mathcal{N}(\mu_{qh}, \Sigma_{qh}) \\ &= \sum_{h=1}^H c_{qh} (2\pi)^{-D/2} \left(\prod_{d=1}^D \sigma_{qhd} \right)^{-1} e^{-\frac{1}{2} \sum_{d=1}^D \left(\frac{o_d - \mu_{qhd}}{\sigma_{qhd}} \right)^2}, \end{aligned} \quad (17)$$

where H is the number of mixture components, c_{qh} is the mixture component weight for state q and mixture h , D is the vector dimension, and $\mathcal{N}(\mu, \Sigma)$ denotes a multivariate Gaussian distribution with mean vector μ and diagonal covariance matrix Σ with elements σ_d . The hyperparameter vector of the mapping in (15) consists of all the parameters of all the HMMs, i.e., $\Lambda = (\lambda_1, \dots, \lambda_M)$.

We have chosen to normalize the log-likelihood values with respect to the length T_x of the sequence $x = (o_1, \dots, o_{T_x})$. The elements of the vector $\phi(x; \Lambda)$ defined in (15) are thus the average log-likelihood per frame for each model. The reason for performing this normalization is that we want utterances of the same word spoken at different speaking rates to map into the same region of space. Moreover, the reason that we use the Viterbi-approximated likelihood instead of the true likelihood is to make it easier to compute its derivatives with respect to the various HMM parameters. These derivatives are needed when we allow the parameters to adapt during training of the logistic regression model.

With the logistic regression mapping ϕ specified, the logistic regression model can be trained and classification can be performed as explained in the previous section. In particular, classification of an observation x is accomplished by selecting the word $\hat{y} \in \mathcal{Y}$ having the largest conditional probability, that is,

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \hat{p}(y | x, W, \Lambda), \quad (18)$$

where

$$\hat{p}(y = k | x, W, \Lambda) = \frac{e^{w_k^\top \phi(x, \Lambda)}}{\sum_{i=1}^K e^{w_i^\top \phi(x, \Lambda)}}. \quad (19)$$

Although in this section we only considered probabilistic prediction of words given a speech segment, the theory is directly applicable to subword units such as phones.

4. N-best rescoring using logistic regression

In this section, we consider the continuous speech recognition problem, which amounts to finding the best sequence of subwords, or sentence hypothesis, given a whole utterance of a sentence. A problem we have to deal with in this context is that the segment boundaries are not known. We propose a two step approach: 1) generate an N-best list using a set of HMMs and the Viterbi algorithm (Viterbi, 1983), and 2) rescore the N-best list and select the

sentence hypothesis with the highest score. Rescoring of a sentence hypothesis is done by obtaining probabilities of each subword using logistic regression, and combining the subword probabilities into a new sentence score using a geometric mean. These sentence scores are used to reorder the sentence hypotheses in the N-best list. The recognized sentence hypothesis of an utterance is then taken to be the first one in the N-best list, i.e., the sentence hypothesis with the highest score.

In the following, let us assume that we have a set of HMMs, one for each subword (e.g., a digit in a spoken digit string, or a phone). We will refer to these HMMs as the baseline models and they will play an important role in both the training phase and the recognition phase of our proposed approach for continuous speech recognition using logistic regression. For convenience, we let $z = (o_1, \dots, o_{T_s})$ denote a sequence of feature vectors extracted from a spoken utterance of a sentence $s = (y_1, \dots, y_{L_s})$ with L_s subwords. Each subword label y_i is one of $(1, \dots, K)$, where K denotes the number of different subwords. Given a feature vector sequence z extracted from a spoken utterance s , the baseline models can be used in conjunction with the Viterbi algorithm in order to generate a sentence hypothesis $\hat{s} = (\hat{y}_1, \dots, \hat{y}_{L_s})$, which is a hypothesized sequence of subwords. Additional information provided by the Viterbi algorithm is the maximum likelihood (ML) segmentation on the subword level, and approximations to the subword likelihoods. We write the ML

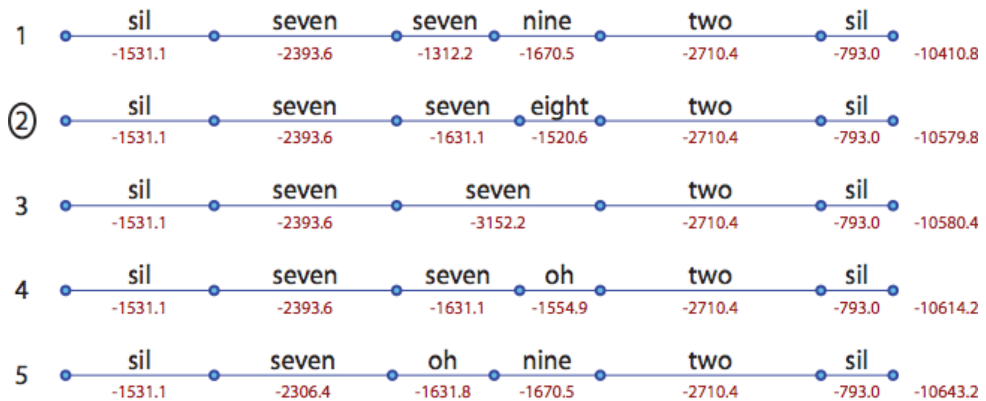


Fig. 3. A 5-best list where the numbers below the arcs are HMM log-likelihood values corresponding to the segments. The total log-likelihood for each sentence hypothesis is shown at the right. The list is sorted after decreasing log-likelihood values for the sentences. The circle around sentence number 2 indicates that this is the correct sentence.

segmentation as $z = (x_1, \dots, x_{L_s})$, where x_l denotes the subsequence of feature vectors associated with the l th subword \hat{y}_l of the sentence hypothesis.

For a given utterance, we can use the baseline models to generate an N-best list of the N most likely sentence hypotheses (Schwartz and Chow, 1990). An example of a 5-best list is shown in Fig. 3. The list is generated for an utterance of the sentence "seven, seven, eight, two", with leading and trailing silence. The most likely sentence hypothesis according to the

HMMs appears at the top of the list and is the sentence “seven, seven, nine, two”. This sentence differs from the correct sentence, which is the second most likely sentence hypothesis, by one subword. The segmentation of each sentence hypothesis in the list is the most likely segmentation given the sentence hypothesis. Each segment is accompanied with the HMM log-likelihood.

The reason for generating N-best lists is to obtain a set of likely sentence hypotheses with different labeling and segmentation, from which the best sentence hypothesis can be chosen based on additional knowledge. In the following we will first consider how we can obtain reliable subword probabilities given speech segments appearing in N-best lists. We suggest using a garbage class for this purpose. Then, we introduce a method for rescoring N-best lists using these estimated subword probabilities.

4.1 Logistic regression on segments in N-best lists

Provided that the baseline models are reasonably good, many of the segments in the N-best lists are good in the sense that they correspond to a complete utterance of exactly one subword. However, it is inherent that N-best lists frequently contain segments that do not correspond to a complete utterance of exactly one subword. Some segments, for example, correspond to only a part of an utterance of a subword, or even an utterance of several subwords together. Consider again the 5-best list in Fig. 3, where the correct sentence hypothesis appears in position 2. Let us assume that the correct unknown segmentation coincides with the ML segmentation in position 2. Then, the third segment in sentence hypothesis 3 actually corresponds to an utterance of the two connected digits “seven” and “eight” spoken in a sequence. Moreover, for hypotheses 1 and 5, the third segment may not correspond to a complete utterance of “seven”, whereas the fourth segment corresponds to an utterance of the last part of “seven” and the whole of “eight”. Thus, the segments of an N-best list can be roughly divided into two: good segments and garbage segments.

The role of logistic regression in our N-best rescoring approach is to provide conditional probabilities of subword labels given a segment. Obviously, we want a correct subword label to get high conditional probability given a good segment. This implies that incorrect subword labels will get low probabilities for good segments since the total probability should sum to one. Furthermore, garbage segments should result in low probabilities for all subword labels. For this reason we introduce a garbage class, whose role is to aggregate large probability for garbage segments and low probability otherwise. In the training of the model, we need two sets of training examples; 1) a set of good segments each labeled with the correct subword label, and 2) a set of garbage segments labeled with the garbage label.

Let us first discuss how we can obtain segments from the former set. If the training utterances were segmented on the subword level, i.e., if we knew the segment boundaries of each subword, we could simply use these subword-labeled segments as the training set for the logistic regression model. In most training databases for speech however, the segment boundaries are not known, only the orthographic transcription, i.e., the correct subword sequence. Then, the most straightforward thing to do would be to estimate the segment boundaries. For this, we will make use of the baseline models to perform Viterbi forced alignment (FA) segmentation on the training data. From a pair (z, s) in the training

database, FA segmentation gives us a set $\{(x_1, y_1), \dots, (x_{L_s}, y_{L_s})\}$ of subword labeled segments. Doing this for all the pairs (z, s) in the training database yields a set

$$\mathcal{D}_{\text{FA}} = \{(x_l, y_l)\}_{l=1, \dots, L_{\text{FA}}} \quad (20)$$

of all FA-labeled segments.

Extracting garbage segments to be used in the training of the logistic regression model is more difficult. In the rescoring phase, segments that differ somehow from the true unknown segments should give small probability to any class in the vocabulary, and therefore high probability to the garbage class. In order to achieve this, we generate an N-best list for each training utterance, and compare all segments within the list with the corresponding forced alignment generated segments, or the true segments if they are known. The segments from the N-best list that have at least ε number of frames not in common with any of the forced alignment segments, are labeled with the garbage label $K+1$ and used as garbage segments for training. This gives us a set

$$\mathcal{D}_{\text{gar}} = \{(x_l, K+1)\}_{l=1, \dots, L_{\text{gar}}} \quad (21)$$

of all garbage-labeled segments. The full training data used to train the logistic regression model is therefore

$$\mathcal{D} = \mathcal{D}_{\text{FA}} \cup \mathcal{D}_{\text{gar}}. \quad (22)$$

4.2 The rescoring procedure

Now that we have seen how logistic regression can be used to obtain the conditional probability of a subword given a segment, we will see how we can use these probability estimates to rescore and reorder sentence hypotheses of an N-best list.

For a given sentence hypothesis $\hat{s} = (\hat{y}_1, \dots, \hat{y}_{L_s})$ in an N-best list with corresponding segmentation $z = (x_1, \dots, x_{L_s})$, we can use logistic regression to compute the conditional probabilities $\hat{p}_{\hat{y}_l} = \hat{p}(y = \hat{y}_l | x_l, W, \Lambda)$. A score for the sentence hypothesis can then be taken as the geometric mean of these probabilities multiplied by a weighted language model score $\hat{p}(\hat{s})$ as in

$$v_{\hat{s}} = \left(\prod_{l=1}^{L_s} \hat{p}_{\hat{y}_l} \right)^{1/L_s} (\hat{p}(\hat{s}))^{\beta}, \quad (23)$$

where β is a positive weight needed to compensate for large differences in magnitude between the two factors. In order to avoid underflow errors caused by multiplying a large number of small values, the score can be computed as

$$v_{\hat{s}} = \exp \left\{ \frac{1}{L_s} \sum_{l=1}^{L_s} \log \hat{p}_{\hat{y}_l} + \beta \log \hat{p}(\hat{s}) \right\}. \quad (24)$$

When all hypotheses in the N-best list have been rescored, they can be reordered in descending order based on their new score. Fig. 4 shows the 5-best list in Fig. 3 after rescoring and reordering. Now, the correct sentence hypothesis "seven, seven, eight, two" has the highest score and is on top of the list.

Additional performance may be obtained by making use of the log-likelihood score for the sentence hypothesis already provided to us by the Viterbi algorithm. For example, if $\hat{p}(z|\hat{s})$ denotes the sentence HMM likelihood, we can define an interpolated logarithmic score as

$$\tilde{v}_s = (1 - \alpha) \frac{1}{L_s} \sum_{l=1}^{L_s} \log \hat{p}_{y_l} + \alpha \log \hat{p}(z|\hat{s}) + \beta \log \hat{p}(\hat{s}), \quad (25)$$

where $0 \leq \alpha \leq 1$.

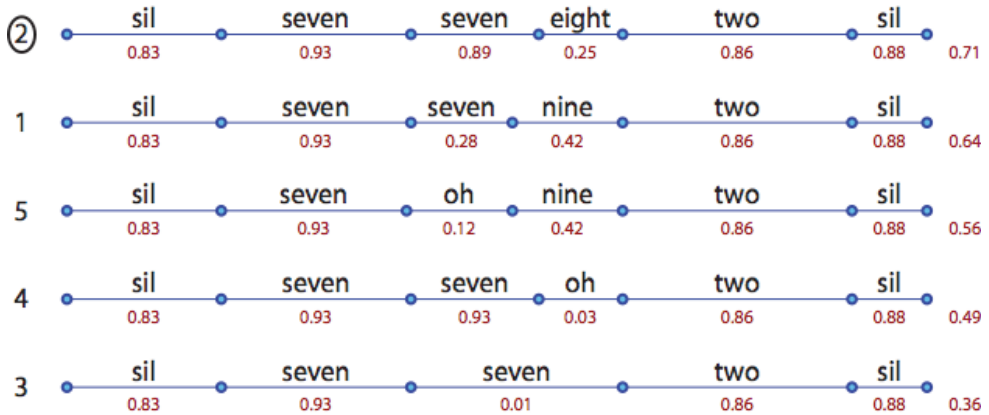


Fig. 4. The 5-best list in Fig. 3 after rescoring using penalized logistic regression with HMM log-likelihood regressors. The hypotheses have been re-ordered according to sentence scores computed from geometric means of the segment probabilities. Sentence number 2, which is the correct one, is now at the top of the list.

5. Experimental results

We performed rescoring of 5-best lists generated by an HMM baseline speech recognizer on the Aurora2 database (Pearce and Hirsch, 2000). We tried both rescoring without a garbage class, and with a garbage class. In the latter experiment, we also interpolated the logistic regression score and the HMM score. In all experiments, a flat language model was used.

5.1 The Aurora2 database and the baseline system

The Aurora2 connected digits database (Pearce & Hirsch, 2000) contains utterances, from different speakers, of digit strings with lengths 1–7 digits. We used only the clean data in both training and testing. The clean data corresponds to the data in the TI-digits database (Leonard, 1984) downsampled to 8 kHz and filtered with a G712 characteristic.

There are 8440 training utterances and 4004 test utterances in the training set and the test set, respectively. The speakers in the test set are different from the speakers in the training set.

From each speech signal, a sequence of feature vectors were extracted using a 25 ms Hamming window and a window shift of 10 ms. Each feature vector consisted of 12 Mel-frequency cepstral coefficients (MFCC) and the frame energy, augmented with their delta and acceleration coefficients. This resulted in 39-dimensional vectors.

Each of the digits 1–9 was associated with one class, while 0 was associated with two classes reflecting the pronunciations “zero” and “oh”. The number of digit classes was thus $C = 11$. For each of the 11 digit classes, we used an HMM with 16 states and 3 mixtures per state. In addition, we used a silence (sil) model with 3 states and 6 mixtures per state, and a short pause (sp) model with 1 state and 6 mixtures. These HMM topologies are the same as the ones defined in the training script distributed with the database. We refer to these models as the baseline models, or collectively as the baseline recognition system. The sentence accuracy on the test set using the baseline system was 96.85%.

5.2 Rescoring 5-best lists without a garbage class

Before training the logistic regression model, the training data was segmented using the baseline models with forced alignment. We updated only the means of the HMMs while keeping the other HMM parameters fixed. For each of the coordinate descent iterations we used the Rprop method (Riedmiller & Braun, 1993) with 100 iterations to update the HMM means Λ and the Newton method with 4 iterations to update W . After 30 coordinate descent iterations, the optimization was stopped.

We used the trained logistic regression model to rescore 5-best lists that were generated on the test set by the baseline recognition system. The upper bound on the sentence accuracy inherent in the 5-best lists, i.e., the sentence accuracy obtainable with a perfect rescoring method, was 99.18%. We chose to rescore only those sentence hypotheses in each 5-best list that had the same number of digits as the first hypothesis in the list (Birkenes et al., 2006b). The resulting sentence accuracy was 97.20%.

5.3 Rescoring 5-best lists with a garbage class

We now present results that we achieved with 5-best rescoring with the use of a garbage class in the logistic regression model. The 5-best lists used in the rescoring phase were the same as above. This time the training was done using two sets of segments; correct segments with the correct class label, and garbage segments with the garbage label. The former set was generated using the baseline recognition system with forced alignment on the training data. The garbage segments were generated from 5-best lists on the training data, with $\varepsilon = 10$. Again, we updated only the mean values of the HMMs while keeping the other HMM parameters fixed. The training with the coordinate descent approach was done in the same way as above. Also this time we stopped the optimization after 30 iterations.

The sentence accuracies for $\delta \in \{10^3, 10^4, 10^5, 10^6\}$ are shown in Fig. 5. The baseline accuracy and the accuracy of the 5-best rescoring approach without a garbage class are also shown. We see that our approach with a garbage class gives the best accuracy for the four values of the regularization parameter δ we used in our experiments. For lower values of δ , we

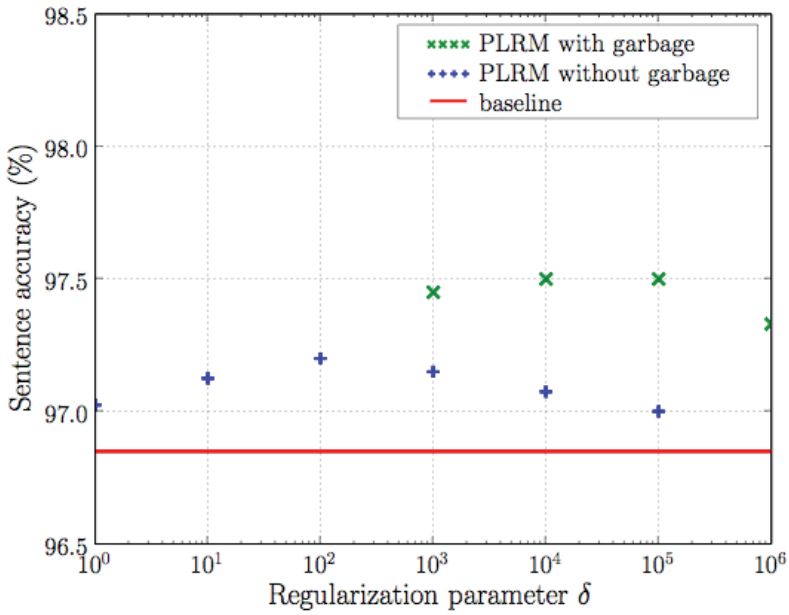


Fig. 5. Sentence accuracy on the test set for various δ .

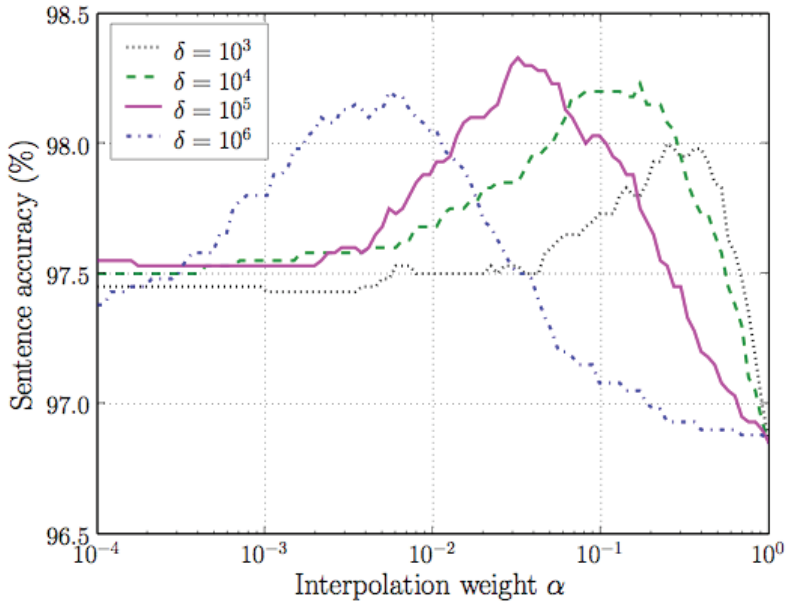


Fig. 6. Sentence accuracy using interpolated scores.

expect a somewhat lower sentence accuracy due to overfitting. Very large δ values are expected to degrade the accuracy since the regression likelihood will be gradually negligible compared to the penalty term.

Fig. 6 shows the effect of interpolating the HMM sentence likelihood with the logistic regression score. Note that with $\alpha = 0$, only the logistic regression score is used in the rescoring, and when $\alpha = 1$, only the HMM likelihood is used. The large gain in performance when taking both scores into account can be explained by the observation that the HMM score and the logistic regression score made very different sets of errors.

6. Summary

A two-step approach to continuous speech recognition using logistic regression on speech segments has been presented. In the first step, a set of hidden Markov models (HMMs) is used in conjunction with the Viterbi algorithm in order to generate an N-best list of sentence hypotheses for the utterance to be recognized. In the second step, each sentence hypothesis is rescored by interpolating the HMM sentence score with a new sentence score obtained by combining subword probabilities provided by a logistic regression model. The logistic regression model makes use of a set of HMMs in order to map variable length segments into fixed dimensional vectors of regressors. In the rescoring step, we argued that a logistic regression model with a garbage class is necessary for good performance.

We presented experimental results on the Aurora2 connected digits recognition task. The approach with a garbage class achieved a higher sentence accuracy score than the approach without a garbage class. Moreover, combining the HMM sentence score with the logistic regression score showed significant improvements in accuracy. A likely reason for the large improvement is that the HMM baseline approach and the logistic regression approach generated different sets of errors.

The improved accuracies observed with the new approach were due to a decrease in the number of substitution errors and insertion errors compared to the baseline system. The number of deletion errors, however, increased compared to the baseline system. A possible reason for this may be the difficulty of sufficiently covering the space of long garbage segments in the training phase of the logistic regression model. This needs further study.

7. References

- Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1-10
- Berger, J. (1985). *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, 2 edition
- Birkenes, Ø. (2007). *A Framework for Speech Recognition using Logistic Regression*, PhD thesis, Norwegian University of Science and Technology (NTNU)
- Birkenes, Ø.; Matsui, T. & Tanabe, K. (2006a). Isolated-word recognition with penalized logistic regression machines, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Toulouse, France

- Birkenes, Ø.; Matsui, T.; Tanabe, K. & Myrvoll, T. A. (2006b). Continuous speech recognition with penalized logistic regression machines, *Proceedings of IEEE Nordic Signal Processing Symposium*, Reykjavik, Iceland
- Birkenes, Ø.; Matsui, T.; Tanabe, K. & Myrvoll, T. A. (2007). N-best rescoring for speech recognition using penalized logistic regression machines with garbage class, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA
- Bishop, C. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press
- Clarkson, P. & Moreno, P. (1999). On the use of support vector machines for phonetic classification, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, pp. 585-588
- Hestenes, M. R. & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49: 409-436
- Leonard, R. (1984). A database for speaker independent digit recognition, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Vol. 3, pp. 42.11
- Pearce, D. & Hirsch, H.-G. (2000). The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions, *In ISCA ITRW ASR*, pp. 181-188, Paris, France
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77(2):257-286
- Riedmiller, M. & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *Proceedings of the IEEE Intl. Conf. on Neural Networks*, pp. 586-591, San Francisco, CA
- Schwartz, R. & Chow, Y. (1990). The N-best algorithm: an efficient and exact procedure for finding the N most likely sentence hypotheses, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 81-84, Albuquerque, New Mexico, USA
- Tanabe, K. (1977). Conjugate-gradient method for computing the moore-penrose inverse and rank of a matrix. *Journal of Optimization Theory and Applications*, 22(1):1-23
- Tanabe, K. (2001a). Penalized logistic regression machines: New methods for statistical prediction 1. *ISM Cooperative Research Report 143*, pp. 163-194
- Tanabe, K. (2001b). Penalized logistic regression machines: New methods for statistical prediction 2, *Proceedings of Information-based Induction Sciences (IBIS)*, pp. 71-76, Tokyo, Japan
- Tanabe, K. (2003). Penalized logistic regression machines and related linear numerical algebra, *In KOKYUROKU 1320, Institute for Mathematical Sciences*, pp. 239-250, Kyoto, Japan
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*, Springer, 2 edition
- Viterbi, A. (1983). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory*, 13(4):179-190

Zahorian, S.; Silsbee, P. & Wang, X. (1997). Phone classification with segmental features and a binary-pair partitioned neural network classifier, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Vol. 2, pp. 1011-1014