DISCRIMINATIVE TRAINING OF LANGUAGE MODEL CLASSIFIERS

Uwe Ohler, Stefan Harbeck and Heinrich Niemann

Chair for Pattern Recognition
University of Erlangen-Nuremberg
Martensstrasse 3
D-91058 Erlangen, Germany
e-mail: {*ohler,snharbec,niemann*}@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de/Persons/oh

ABSTRACT

We show how discriminative training methods, namely the *Maximum Mutual Information* and *Maximum Discrimination* approach, can be adopted for the training of N-gram language models used as classifiers working on symbol strings. By estimating the model parameters according to a discriminative objective function instead of Maximum Likelihood, the emphasis is not put on the exact modeling of each class, but on the right classification of the samples. The methods are shown to be suited for a variety of applications, such as the recognition of regulatory DNA sequences and language identification. Using phonotactic information, we achieve an error reduction of 10.7% (phoneme sequences) or 41.9% (codebook classes) with respect to the standard ML estimation on a corpus of English and German sentences.

1. INTRODUCTION

N-gram language models [3] are best known as linguistic component in an automatic speech recognition system; usually, they are applied to estimate the probabilities of word chains in order to find the best chain in a word hypothesis graph or lattice. Thereby, the training goal is the minimization of the model perplexity which is equivalent to an optimization of the parameters according to Maximum Likelihood (ML). We can regard an N-gram language model as a stochastic regular grammar. Thus, several language models used in parallel are well suited as a syntactic classifier working on any kind of symbol string drawing its characters from a finite alphabet. Our group could already show the applicability of this approach in areas such as the classification of dialog acts, languages, and DNA sequences [7, 6].

A widespread way to estimate N-gram language model parameters is the following: First we perform an estimation of the symbol probabilities depending on up to $N-1$ preceding symbols, then we apply an interpolation technique to yield a more robust estimation of the parameters. Usually, both steps are performed according to the Maximum Likelihood criterion, and this is well motivated for the estimation of a single language model. But if several language models for classification purposes are to be estimated, this approach loses its eligibility: Our main focus is not the maximization of the production probability, where each class is considered independently of the others, but the maximum number of correct decisions. This justifies the application of a discriminative estimation method like Maximum Mutual Information (MMI) and its special case, Maximum Discrimination (MD), which take into account all classes simultaneously and optimize an objective function based on the *a posteriori* probability of the right decision. MMI estimation techniques were mainly used to estimate HMM parameters for speech recognition [1, 5]; the application of MD for protein sequence modeling is described in [2]. Here we will show how to apply these techniques within the first step of the construction of language model classifiers.

The paper is organized as follows: After a short review of the basic concepts of language models and ML estimation, the main part describes estimation techniques for MMI and MD parameter training. Afterwards we will show how MMI and MD compare with standard ML estimated models on different data sets. We close with a discussion of the perspectives and limitations of the discriminative estimation of language model classifiers.

2. TRAINING OF LANGUAGE MODELS

Let us assume that we have K classes $\Omega_1 \ldots \Omega_K$ and wish to classify the symbol sequence $\boldsymbol{w} = w_1 \ldots w_T$ with symbols w_i taken from a finite vocabulary $\mathcal{V}$. The likelihood of a sequence $\boldsymbol{w}$ for each class Ω_k can be computed with the chain rule:

$$P_k(\boldsymbol{w}) := P(\boldsymbol{w}|\Omega_k) = \prod_{t=1}^{T} P_k(w_t | \underbrace{w_1 \ldots w_{t-1}}_{\text{context}}). \quad (1)$$

With given *a priori* probabilities p_k, we are then able to classify the sequence into class $\hat{k}$ according to the largest *a posteriori* probability with the Bayes rule:

$$\hat{k} = \underset{k}{\operatorname{argmax}} P(\Omega_k|\boldsymbol{w}) = \underset{k}{\operatorname{argmax}} (p_k \cdot P_k(\boldsymbol{w})) \quad (2)$$

The right hand side of equation (1) contains a context of arbitrary length which cannot be handled. A possible approximation of the probability $P_k(\boldsymbol{w})$ is made by limiting the context length to $N-1$ which is the basic idea of N-gram language models:

[*] Part of this work was supported by a grant of the Boehringer Ingelheim Fonds to U. O.

$$P_k(\boldsymbol{w}) \approx \prod_{i=1}^{T} P_k(w_i|w_{i-N+1}\dots w_{i-1}) \qquad (3)$$

2.1. Maximum Likelihood Training

Our goal is to obtain parameters, i. e. values for the conditional probabilities $P_k(w_i|w_{i-N+1}\dots w_{i-1})$, which lead to the best possible recognition rate on the K classes under consideration. As we cannot optimize the recognition rate directly, we have to use objective functions which show the desired behaviour and for which a solution can be found. One well-known objective function is *Maximum Likelihood* (ML). If Θ_k denotes the set of the parameters of model M_k for class Ω_k, we optimize the following function:

$$R^{\mathrm{ML}}_{\Theta_k}(\mathcal{W}^k) = \prod_{i=1}^{n_k} P(\boldsymbol{w}_i^k|M_k), \qquad (4)$$

where n_k is the number of sequences $\boldsymbol{w}_i^k$ in the training set $\mathcal{W}^k$ for class k. The ML estimation of the conditional probabilities $\tilde{P}_k(v|\hat{\boldsymbol{v}})$ for all elements $v \in \mathcal{V}$ and all possible contexts $\hat{\boldsymbol{v}} = v_1 \dots v_{N-1}$ of length $N-1$ can be carried out simply by counting the N- and $N-1$-grams in a set of training sequences:

$$\tilde{P}_k(v|\hat{\boldsymbol{v}}) = \frac{\#(\hat{\boldsymbol{v}}v)}{\#(\hat{\boldsymbol{v}})}, \qquad (5)$$

where # denotes the frequency of its argument in the training sample for the respective class.

2.2. Maximum Mutual Information Training

The ML objective function regards each class as independent of the others and aims at the maximization of the probability that the given training sample was generated, knowing to which class each sequence belongs. In contrast, the MMI objective function,

$$\begin{aligned} R^{\mathrm{MMI}}_{\Theta}(\mathcal{W}) &= \prod_{i=1}^{n} P(M_{q_i}|\boldsymbol{w}_i) \\ &= \prod_{i=1}^{n} \frac{P(\boldsymbol{w}_i|M_{q_i})P(M_{q_i})}{\sum_j P(\boldsymbol{w}_i|M_j)P(M_j)}, \qquad (6) \end{aligned}$$

maximizes the *a posteriori* probability of a model under the assumption that a pattern associated with this model was observed. Here, q_i gives the number of the correct model for sequence $\boldsymbol{w}_i$, and n is the total number of training sequences for all classes. Assuming that we have one training sequence $\boldsymbol{w}_i$, the partial derivation of the logarithm of the MMI objective function with respect to parameter $P_k(v|\hat{\boldsymbol{v}})$ leads us to

$$\begin{aligned} \frac{\partial \log R^{\mathrm{MMI}}_{\Theta}(\boldsymbol{w}_i)}{\partial P_k(v|\hat{\boldsymbol{v}})} &= \frac{\partial}{\partial P_k(v|\hat{\boldsymbol{v}})}(\log P(\boldsymbol{w}_i|M_{q_i})P(M_{q_i}) \\ &\quad - \log \sum_j P(\boldsymbol{w}_i|M_j)P(M_j)) \\ &= \frac{\#(\hat{\boldsymbol{v}}v)}{P_k(v|\hat{\boldsymbol{v}})}\delta_{k,q_i} - \frac{\#(\hat{\boldsymbol{v}}v)}{P_k(v|\hat{\boldsymbol{v}})} \cdot \frac{P(\boldsymbol{w}_i|M_k)P(M_k)}{\sum_j P(\boldsymbol{w}_i|M_j)P(M_j)} \\ &=: \frac{1}{P_k(v|\hat{\boldsymbol{v}})}(\#_{k,q_i}(\hat{\boldsymbol{v}}v) - \#'(\hat{\boldsymbol{v}}v)) \qquad (7) \end{aligned}$$

where δ_{k,q_i} is equal to one if $q_i = k$ and zero otherwise. $\#'$ is a weighted counting function, and $\#_{k,q_i}$ is a function which counts only if $q_i = k$.

We follow the approach described by Normandin *et al.* (see [5] and references therein) who carry out the parameter optimization with a re-estimation formula for rational objective functions such as MMI:

$$\tilde{P}_k(v|\hat{\boldsymbol{v}}) = \frac{P_k(v|\hat{\boldsymbol{v}})\left(\frac{\partial \log R^{\mathrm{MMI}}_{\Theta}(\mathcal{W})}{\partial P_k(v|\hat{\boldsymbol{v}})} + D\right)}{\sum\limits_{v_j \in \mathcal{V}} P_k(v_j|\hat{\boldsymbol{v}})\left(\frac{\partial \log R^{\mathrm{MMI}}_{\Theta}(\mathcal{W})}{\partial P_k(v_j|\hat{\boldsymbol{v}})} + D\right)} \qquad (8)$$

For a sufficiently large constant D, the convergence to a local optimum was proven. In practice, we choose D to be equal to

$$D = \max_{v_j \in \mathcal{V}} \left\{ -\frac{\partial \log R^{\mathrm{MMI}}_{\Theta}(\mathcal{W})}{\partial P_k(v_j|\hat{\boldsymbol{v}})}, 0 \right\} + \epsilon \qquad (9)$$

which then guarantees that the new parameters fulfill the conditions of a probability distribution. The original value of the partial derivation is replaced by

$$\frac{\partial \log R^{\mathrm{MMI}}_{\Theta}(\mathcal{W})}{\partial P_k(v|\hat{\boldsymbol{v}})} \approx \frac{\#_{k,q_i}(\hat{\boldsymbol{v}}v)}{\sum\limits_{v_j \in \mathcal{V}} \#_{k,q_i}(\hat{\boldsymbol{v}}v_j)} - \frac{\#'(\hat{\boldsymbol{v}}v)}{\sum\limits_{v_j \in \mathcal{V}} \#'(\hat{\boldsymbol{v}}v_j)} \qquad (10)$$

to remove emphasis from low-valued parameters and achieve a more stable convergence.

2.3. Maximum Discrimination Estimation

In [2] a variant of MMI was proposed under the name *Maximum Discrimination* (MD). Each class is trained according to MMI, but using only positive samples. The derivation of the objective function for class k is then equal to

$$\frac{\partial \log R^{\mathrm{MD}}_{\Theta}(\boldsymbol{w}_i)}{\partial P(v|\hat{\boldsymbol{v}})} = \frac{\#(\hat{\boldsymbol{v}}v)}{P(v|\hat{\boldsymbol{v}})}\left(1 - R^{\mathrm{MMI}}_{\Theta}(\boldsymbol{w}_i)\right), \qquad (11)$$

because $\delta_{k,q}$ (equation 7) is always equal to one, and the negative weight term is equal to the MMI objective function. We can introduce the condition that all parameters belonging to the same distribution must sum up to one with the help of Lagrange multipliers. This leads us to an expectation-maximization-style re-estimation formula for the parameters:

$$\tilde{P}_k(v|\hat{\boldsymbol{v}}) = \frac{\#(\hat{\boldsymbol{v}}v)(1 - R^{\mathrm{MMI}}_{\Theta}(\boldsymbol{w}_i))}{\sum\limits_{v_j \in \mathcal{V}} \#(\hat{\boldsymbol{v}}v_j)(1 - R^{\mathrm{MMI}}_{\Theta}(\boldsymbol{w}_i))} \qquad (12)$$

The values on the right side are calculated using the parameters of the last iteration. If we have n training sequences, the numerator and denominator sum up over all of them. Once initialized with values greater than zero, the parameters will always be greater or equal than zero, thus fulfilling all characteristics of a probability distribution.

To ensure that no models parameters are set to zero during the iterations, the counts on the right hand side are modified by Dirichlet priors on the parameters. If we have no *a priori* information on the parameters, this leads to a discounting of $1/n_k$.

A closer look at equation (12) shows that MD can be regarded as nothing else than a weighted version of ML estimation where the training sequences have weights dependent on how bad they are actually recognized by the correct model.

2.4. Corrective Training and Model Interpolation

To avoid oscillatory effects during the course of training, it was necessary for both approaches to perform an interpolation between the model before and after an estimation iteration. In the case of MD, we assign a class-dependent weight to the updated parameters which declines logarithmically with the number of iterations and is additionally dependent on the classification performance of the old model.

In the case of MMI, a uniform weight of 0.98 assigned to the old model performed well in all cases. The small weight for the new model is partly due to the fact that we performed a corrective training as proposed in [5], i. e. a training where only the misclassified sequences of the last iteration are part of the actual training set. This is justified by the observation that well recognized sequences do not contribute much to the derivation (eq. 7) and can thus be left away without much harm. This improves drastically on the speed of an iteration, as only a fraction of the sequences has to be taken into account.

3. INTERPOLATION OF LANGUAGE MODELS

The choice of the context length is a crucial point in the training process of a language model. If its value is too small, the resulting N-grams are not distinctive enough; if its value is too large, the model runs into the danger of over-fitting to the training material, as the number of parameters increases exponentially with the context length.

A compromise to this dilemma can be found by introducing an interpolation of models with different context length. The interpolation parameters are optimized using a disjoint part of the training sample. For instance, we can perform a linear interpolation:

$$\hat{P}(v|\hat{\boldsymbol{v}}) := \rho_0 \frac{1}{L} + \rho_1 \tilde{P}(v) + \ldots + \rho_N \tilde{P}(v|\hat{\boldsymbol{v}}) \qquad (13)$$

The weights can again be calculated using different objective functions such as ML or MMI. For detailed information on this topic, the reader is referred to [7, 8].

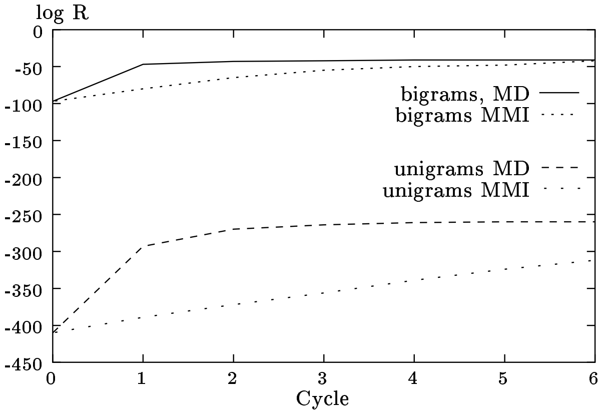

Figure 1. Convergence of the MMI objective function during the iterations of MMI and MD training.

4. EXPERIMENTS AND RESULTS

We examined the impact of discriminative estimation on different data sets to show its general usefulness. The models are initialized with the standard ML counts, discounted by one to assure that no parameter value is set to zero. This initialization is then followed by several cycles of MMI or MD estimation until a preset number of cycles is fulfilled, both recognition rate and value of the objective function do not increase any further, or all sequences of one class are completely recognized.

4.1. Language Identification

To examine the discriminative training, we took the 478 German and 464 English sentences of the OGI training and validation material as training set and tested the performance on the official NIST database test sentences cut in 10 seconds long disjoint pieces, thus totaling in 100 German and 100 English samples. The baseline system for our language identification system is described in the accompanying paper on *multigrams for language identification* [4]. We use the language models as a classifier for symbol strings which are obtained in a first step. Here, we show the classification of sequences of phonemes (80 symbols) and codebook classes (180 symbols). Figure 1 depicts the convergence of the objective function on the training set of phonemes for both MMI and MD estimation during the first six cycles. Due to the different approach of the model interpolation (sec. 2.4), the application of MMI leads to a slower convergence, but in all cases under consideration, less than 20 cycles were needed to obtain stable parameters. For the codebook class experiments, we stopped the training after 5 (MMI) resp. 2 (MD) cycles to prevent over-adaptation to the small sample.

The recognition results are given in table 1. We compare the results to (1) the models obtained by ML estimation and (2) the models obtained by ML with a follow-up linear interpolation. For both phonemes and codebook classes, a considerable improvement could be achieved: without interpolation, the error rate on phoneme sequences is reduced by 10.7 %, and on codebook class sequences by 41.9 % for both MD and MMI. The results with interpolation were obtained with uniform weights for equation 13; an opti-

Method	**Results (%)**	
	phonemes	*codebook classes*
ML	86.0	84.5
ML interp.	87.0	—
MD	87.5	91.0
MD interp.	88.0	—
MMI	87.5	91.0
MMI interp.	89.0	—

Table 1. Recognition of German and English sentences of 10 seconds. Shown is the average recognition rate on sequences of phonemes or codebook classes which were obtained in a previous step. The results on phonemes were achieved by bigrams, the results on codebook classes on unigrams; therefore, in the latter case no results for interpolated models are shown.

mization of the weights according to the ML criterion does not lead to an improvement in any of the considered cases.

4.2. DNA Sequence Classification

An increasingly important application field for speech recognition methods emerges for bioinformatics problems, such as the classification of DNA sequences into several functional classes. In this case, the vocabulary consists of the four nucleotides which are the basic units of DNA sequences. An interesting problem within sequence analysis is the identification of so-called promoter sequences which have regulatory potential over neighbouring genes [6]. In figure 2 a part of the receiver operating characteristics obtained by a five-fold cross-validation experiment on a standard set for promoter vs. non-promoter classification is depicted. We compare the results of standard discounted ML estimation using 6-grams and the improvements made by successive application of MMI and MD. The current system uses a threshold set at four percent of false positives; the figure shows that in this case the recognition rate could be improved from 53.6 (ML) to 55.8 (MD) respectively 58.6 % (MMI).

5. CONCLUSIONS AND FUTURE WORK

Our results show clearly that a discriminative training outperforms consistently the usual ML parameter estimation. Also, MMI always performs equally or better than the MD approach, but at the cost of much higher computational costs, as the training sequences of all classes have to be examined by each model in the re-estimation step.

A fundamental problem of discriminative estimation with respect to ML is the fact that much more parameters have to be represented explicitly. For ML, only the N-grams which occur in the training set of a particular class are stored. For MD, all parameters which context occurs in the training set of the particular class are part of the model. For MMI the number is further increased by all parameters which context occurs in any of the training sequences, no matter which class they belong to, because each sequence has to be judged by each class. This leads to problems with large vocabularies: With increasing context length, the model quickly gets intractably large, and the obligatory discounting has a stronger impact for a large number of parameters. But as our results show, MMI is well justified for a vocabulary of small to medium size.

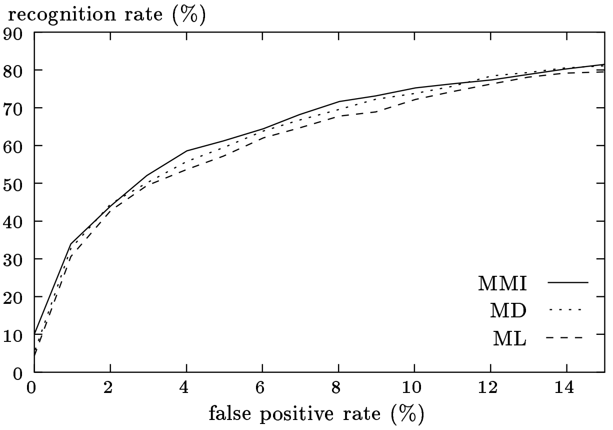

Figure 2. Receiver operating characteristics for the recognition of DNA promoter sequences with language model classifiers trained by MMI, MD and ML estimation.

In a recent paper, our group described how to perform the estimation of interpolation parameters (sec. 3) according to MMI instead of ML [8]. In future we will therefore examine language models for which both parameters and interpolation coefficients are trained with discriminative methods.

REFERENCES

[1] L. Bahl, P. Brown, P. de Souza, and R. Mercer. Maximum Mutual Information estimation of hidden Markov model parameters for speech recognition. In *Proc. ICASSP*, pages 49–52, Tokyo, 1986.

[2] S. R. Eddy, G. Mitchison and R. Durbin. Maximum discrimination hidden Markov models of sequence consensus. *J. Comp. Biol.* 2(1):9–23, 1995.

[3] F. Jelinek. Self-Organized Language Modeling for Speech Recognition. In A. Waibel and K.F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, San Mateo, CA, 1990.

[4] S. Harbeck and U. Ohler. Multigrams for language identification. To appear in *Proc. EUROSPEECH*, Budapest, 1999.

[5] Y. Normandin and S. D. Morgera. An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition. In *Proc. ICASSP*, pages 537–540, Toronto, 1991.

[6] U. Ohler, S. Harbeck, H. Niemann, E. Nöth, and M. G. Reese. Interpolated Markov chains for eukaryotic promoter recognition. To appear in *Bioinformatics*, 1999.

[7] E. G. Schukat-Talamazzini, F. Gallwitz, S. Harbeck, and V. Warnke. Rational interpolation of Maximum Likelihood predictors in stochastic language modeling. In *Proc. EUROSPEECH*, pages 2731–2734, Rhodes, 1997.

[8] V. Warnke, S. Harbeck, E. Nöth, H. Niemann, and M. Levit. Discriminative estimation of interpolation parameters for language model classifiers. In *Proc. ICASSP*, pages 525–528, Phoenix, 1999.