DISCRIMINATIVE TRAINING OF STOCHASTIC MARKOV GRAPHS FOR SPEECH RECOGNITION

F. Wolfertstetter and G. Ruske
Institute for Human-Machine-Communication, Munich University of Technology
Arcisstr. 21, D-80290 München, Germany
E-mail: wol,rus@mmk.e-technik.tu-muenchen.de

ABSTRACT

This paper proposes the application of discriminative training techniques based on the Generalized Probabilistic Descent (GPD) approach to Stochastic Markov Graphs (SMGs), a generalization of mixture-state Hidden Markov Models (HMMs), describing the constraints in the acoustic structure of speech as a graph consisting of nodes, each containing a base function, and a transition network between the nodes. State-specific weights modeling the classification relevance of the corresponding states and a transition weight representing the ratio between transitions and emissions are trained in addition to the standard parameters of the models. The experiments show, that discriminatively trained SMGs outperform discriminatively trained mixture-state HMMs with approximately the same number of parameters.

1. INTRODUCTION AND MOTIVATION

The improved speech recognition performance of discriminatively trained Hidden Markov Models (HMMs) has been shown in numerous publications in the last few years (for example [1],[2],[3]). On the other hand, a more detailed modeling structure based on Stochastic Markov Graphs (SMGs) representing the constraints in the temporal course of the feature vectors was shown to outperform mixture-state HMMs with an equal number of parameters [4], when trained by the maximum-likelihood (ML) approach [5].

In this work, the recognition performance of these two modeling approaches, which will be called Markov-models in general, is compared, when discriminative training is applied to adjust the parameters. We also investigate the effects of a sequential and a joint training of the different types of parameters in the models.
Furthermore, the difference between discriminative training within a given phoneme segmentation (determined automatically) and discriminative training with implicit segmentation is evaluated. A complete mathematical formulation of both approaches is given.

2. METHODS

2.1. Discriminative training within a given phoneme segmentation

For the discriminative training we apply the objective function of the Generalized Probabilistic Descent (GPD) approach [7],[1],[2]. The total cost **y** with the parameters γ and η is given in equations (1) and (2).

$$y = \frac{1}{\sum_{m=1}^{M} N_m} \sum_{m=1}^{M} \sum_{n=1}^{N_m} (1 + e^{\gamma \cdot d(m,n)})^{-1} \tag{1}$$

$$\begin{aligned} d(m,n) &= s(\mathbf{X}_{m,n}, m) - \frac{1}{\eta} \cdot \log\left(\frac{1}{M-1} \cdot \sum_{\tilde{m}=1, \tilde{m} \neq m}^{M} e^{s(\mathbf{X}_{m,n}, \tilde{m}) \cdot \eta}\right) \\ &= -\frac{1}{\eta} \cdot \log\left(\frac{1}{M-1} \cdot \sum_{\tilde{m}=1, \tilde{m} \neq m}^{M} e^{(s(\mathbf{X}_{m,n}, \tilde{m}) - s(\mathbf{X}_{m,n}, m)) \cdot \eta}\right) \end{aligned} \tag{2}$$

Thereby, $\mathbf{X}_{m,n}$ represents the feature vector sequence of the n-th utterance of the phoneme belonging to model number m, d(m,n) is the distance function for this vector sequence, M is the total number of models and N_m is the number of utterances of phoneme number m. The second form of (2) avoids number overflows, when high values for η are used to focus the training on the most dangerous (i.e. best scoring) false models.
Equation (3) gives the general definition of the the log-emission s(**X**,m) of Markov-model number m for vector sequence **X** :

$$s(\mathbf{X}, m) = \sum_{t=1}^{T_X} [w_{m,z(t)} \cdot \log(\sum_{p=1}^{P_{m,z(t)}} (\mathrm{mix}(p|m, z(t)) \cdot \underbrace{\frac{e^{-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_{t,d} - \mu_{m,z(t),p,d})^2}{\sigma^2_{m,z(t),p,d}}}}{\sqrt{(2\pi)^D \prod_{d=1}^{D} \sigma^2_{m,z(t),p.d}}}}_{\mathrm{gauss}(\vec{x}_t | m, z(t), p)}) + \log(a(m, z(t)|m, z(t-1)))] \tag{3}$$

T_X is the length of vector sequence **X** and z(t) yields, for each time step t, the state number of the best Viterbi path through model m producing **X**. $P_{m,z(t)}$ is the number of base functions in state z(t) of model m, mix(p|m,z(t)) is the mixture weight of base function p. gauss($\vec{x}_t$|m,z(t),p) yields the value of base function number p in state z(t) of model m for feature vector $\vec{x}_t$ and a(m,z(t)|m,z(t-1)) is the transition probability from state z(t-1) to z(t) in model m. $w_{m,z(t)}$ is the weight of state z(t) in model m, representing the importance of this state for classification [6].
For the application of discriminative training, the derivative of the total cost with respect to each model parameter λ_m of each model m is calculated according to (4) and (5).

$$\frac{\partial y}{\partial \lambda_m} = \frac{1}{\sum_{\tilde{m}=1}^{M} N_{\tilde{m}}} \sum_{\tilde{m}=1}^{M} \sum_{n=1}^{N_{\tilde{m}}} \frac{-\gamma \cdot e^{\gamma \cdot d(\tilde{m},n)}}{(1 + e^{\gamma \cdot d(\tilde{m},n)})^2} \cdot \frac{\partial d(\tilde{m}, n)}{\partial \lambda_m} \tag{4}$$

$$\frac{\partial d(\tilde{m},n)}{\partial \lambda_m} = \begin{cases} \dfrac{\partial s(\mathbf{X}_{\tilde{m},n},m)}{\partial \lambda_m}, & \text{if } \tilde{m} = m \\ -\dfrac{e^{s(\mathbf{X}_{\tilde{m},n},m)\cdot\eta}}{\sum_{\hat{m}=1,\hat{m}\neq\tilde{m}}^{M} e^{s(\mathbf{X}_{\tilde{m},n},\hat{m})\cdot\eta}} \cdot \dfrac{\partial s(\mathbf{X}_{\tilde{m},n},m)}{\partial \lambda_m}, & \text{if } \tilde{m} \neq m \end{cases} \quad (5)$$

For the derivative of the log-emission s(**X**,m) of model m for vector sequence **X** with respect to the different model parameters we obtain (7),(8),(9),(10) using (6). Thereby, the set V(**X**,m,j) contains all time steps, where the best Viterbi-path for vector sequence **X** in model m runs in state j.

$$\text{apost}(p|\vec{x}_t,m,j) = \frac{\text{mix}(p|m,j)\cdot\text{gauss}(\vec{x}_t|m,j,p)}{\sum_{\hat{p}=1}^{P_{m,j}} \text{mix}(\hat{p}|m,j)\cdot\text{gauss}(\vec{x}_t|m,j,\hat{p})} \quad (6)$$

$$\frac{\partial s(\mathbf{X},m)}{\partial \mu_{m,j,p,d}} = w_{m,j} \cdot \sum_{t\in V(\mathbf{X},m,j)} \text{apost}(p|\vec{x}_t,m,j)\cdot\frac{x_{t,d}-\mu_{m,j,p,d}}{\sigma^2_{m,j,p,d}} \quad (7)$$

$$\frac{\partial s(\mathbf{X},m)}{\partial \sigma^2_{m,j,p,d}} = w_{m,j} \cdot \sum_{t\in V(\mathbf{X},m,j)} \text{apost}(p|\vec{x}_t,m,j)\cdot \cdot\left(\frac{(x_{t,d}-\mu_{m,j,p,d})^2}{2\cdot\sigma^4_{m,j,p,d}} - \frac{1}{2\cdot\sigma^2_{m,j,p,d}}\right) \quad (8)$$

$$\frac{\partial s(\mathbf{X},m)}{\partial \text{mix}(p|m,j)} = w_{m,j} \cdot \sum_{t\in V(\mathbf{X},m,j)} \frac{\text{gauss}(\vec{x}_t|m,j,p)}{\sum_{\hat{p}=1}^{P_{m,j}} \text{mix}(\hat{p}|m,j)\cdot\text{gauss}(\vec{x}_t|m,j,\hat{p})} \quad (9)$$

$$\frac{\partial s(\mathbf{X},m)}{\partial w_{m,j}} = \sum_{t\in V(\mathbf{X},m,j)} \log\left(\sum_{p=1}^{P_{m,j}} \text{mix}(p|m,j)\cdot\text{gauss}(\vec{x}_t|m,j,p)\right) \quad (10)$$

The update of the parameters at the end of iteration k is based on the reestimation formula in [3] for all positive parameters with the constraint to sum up to one and on the gradient descent formula for the other parameters:

$$\mu^{(k+1)}_{m,j,p,d} = \mu^{(k)}_{m,j,p,d} - \varepsilon_\mu \cdot \left(\frac{\partial y}{\partial \mu_{m,j,p,d}}\right)^{(k)} \quad (11)$$

$$\sigma^{2\ (k+1)}_{m,j,p,d} = \sigma^{2\ (k)}_{m,j,p,d} - \varepsilon_{\sigma^2} \cdot \left(\frac{\partial y}{\partial \sigma^2_{m,j,p,d}}\right)^{(k)} \quad (12)$$

$$\text{mix}(p|m,j)^{(k+1)} = \frac{\text{mix}(p|m,j)^{(k)} - \varepsilon_m \cdot \text{mix}(p|m,j)^{(k)} \cdot \left(\frac{\partial y}{\partial \text{mix}(p|m,j)}\right)^{(k)}}{\sum_{\hat{p}=1}^{P_{m,j}} \text{mix}(\hat{p}|m,j)^{(k)} - \varepsilon_m \cdot \text{mix}(\hat{p}|m,j)^{(k)} \cdot \left(\frac{\partial y}{\partial \text{mix}(\hat{p}|m,j)}\right)^{(k)}} \quad (13)$$

$$w^{(k+1)}_{m,j} = \frac{w^{(k)}_{m,j} - \varepsilon_{w\cdot} w^{(k)}_{m,j} \cdot \left(\frac{\partial y}{\partial w_{m,j}}\right)^{(k)}}{\sum_{i=1}^{S_m} w^{(k)}_{m,i} - \varepsilon_{w\cdot} w^{(k)}_{m,i} \cdot \left(\frac{\partial y}{\partial w_{m,i}}\right)^{(k)}} \cdot S_m \quad (14)$$

2.2. Discriminative training with implicit segmentation

In the approach with implicit segmentation, the training can not only optimize the recognition rate, but also improve the segmentation properties, i.e. decrease the number of insertions made by the recognizer.

The total score s(**X**) of a sentence with the acoustic feature vector sequence **X** of length T_X is defined in equation (15).

$$s(\mathbf{X}) = \sum_{t=1}^{T_X} [w_{h(t),z(t)} \cdot \log\left(\sum_{p=1}^{P_{h(t),z(t)}} (\text{mix}(p|h(t),z(t))\cdot \text{gauss}(\vec{x}_t|h(t),z(t),p))\right) + R\cdot\log(a(h(t),z(t)|h(t-1),z(t-1)))] \quad (15)$$

h(t) and z(t) yield the model and state number, respectively, of the best Viterbi path of an unconstrained model concatenation for vector sequence **X** at time step t (recognition task). It is also possible to use a lexicon- or word-bigram-constrained recognizer. In our experiments we only use the top-1 sentence hypothesis and not an n-best sentence approach as it was proposed in [8],[9].
The model independent parameter R weights the transition scores against the emission scores to compensate for the different dynamic properties of emissions and transitions.

In addition to the recognizer score s(**X**), the score $\hat{s}(\mathbf{X})$ of a forced Viterbi segmentation process, i.e. the score of the best Viterbi path when the correct model sequence is given, is defined. The calculation of $\hat{s}(\mathbf{X})$ corresponds to that of s(**X**). However, the functions $\hat{h}(t)$ and $\hat{z}(t)$ are used, yielding the model and state number of the best path at time step t for the forced Viterbi processing.

The distance function d(**X**) is defined in equation (16):

$$d(\mathbf{X}) = \frac{1}{T_X}(\hat{s}(\mathbf{X}) - s(\mathbf{X})) \quad (16)$$

This means, that scores from correct time segments, where the "recognizer Viterbi path" and the "forced Viterbi path" are identical, do not influence the distance function. In contrast to the discriminative training within given phoneme boundaries, where all models are trained in each (correctly or wrongly classified) phoneme segment, the training procedure is focussed only on falsely classified segments of the sentences.

The derivative of the distance function with respect to parameter λ_m of model m is given by (17).

$$\frac{\partial d(\mathbf{X})}{\partial \lambda_m} = \frac{1}{T_X}\left(\frac{\partial \hat{s}(\mathbf{X})}{\partial \lambda_m} - \frac{\partial s(\mathbf{X})}{\partial \lambda_m}\right) \quad (17)$$

The cost function y and its derivative with respect to model parameter λ_m are given by equations (18) and (19).

$$y = \frac{1}{N}\sum_{n=1}^{N}(1+e^{\gamma\cdot d(\mathbf{X}_n)})^{-1} \quad (18)$$

$$\frac{\partial y}{\partial \lambda_m} = \frac{1}{N}\sum_{n=1}^{N}(-\gamma\cdot(1+e^{\gamma\cdot d(\mathbf{X}_n)})^{-2}\cdot e^{\gamma\cdot d(\mathbf{X}_n)})\cdot\frac{\partial d(\mathbf{X}_n)}{\partial \lambda_m} \quad (19)$$

Thereby, N ist the total number of training sentences and $\mathbf{X_n}$ is the feature vector sequence of sentence number n.

The derivatives of the scores $s(\mathbf{X})$ and $\hat{s}(\mathbf{X})$ with respect to means, variances, mixture weights and state weights correspond to equations (7),(8),(9),(10), the update formulae are identical to equations (11),(12),(13),(14). The derivative of the scores with respect to the transition weight R is given by equation (20), the update formula for R is defined in equation (21).

$$\frac{\partial s(\mathbf{X})}{\partial R} = \sum_{t=1}^{T_X}\log(a(h(t),z(t)|\,h(t-1),z(t-1))) \quad (20)$$

$$R^{(k+1)} = R^{(k)} - \varepsilon_R\cdot(\frac{\partial y}{\partial R})^{(k)} \quad (21)$$

The equations for the transition probabilities are not given here, because no improvement was achieved by transition training in the experiments.

3. EXPERIMENTS AND INTERPRETATION

In all experiments, maximum-likelihood trained models are used as initial models for the discriminative training. After 16kHz-sampling, windowing in 10ms steps with a 16ms Hamming window and Fast Fourier Transformation, 20 spectral channels, zero crossing rate, total energy and two loudness features are extracted within each window. All experiments use the German "Diphon" database containing all possible diphon combinations and a natural distribution of the phoneme a priori probabilities. 7771 sentences from 67 speakers are used for training and 3301 sentences from 33 other speakers are used for evaluation.
Since discriminative training requires a lot of computing resources, we use an HMM and an SMG having only about 500 base functions. The base functions are state-specific in both approaches.

3.1 Training within a given phoneme segmentation

The training parameters, η=1.0, γ=0.1, ε_μ=0.2, ε_{σ^2} =2.0*10^{-6}, ε_m=200 and ε_w =0.4, have been optimized by try and error for a subset of the training database. The phoneme recognition results on the test data shown for the HMMs in table 1 and for the SMGs in table 2 have been evaluated within the automatically generated phoneme segmentation, i.e. only the recognition performance and not the segmentation properties of the models are measured here. No mixtures exist in the SMG-approach, since those models consist of mono-modal states [4].

ML-trained: 67.82%		
↓	GPD-μ: 71.44%	GPD-w: 69.54%
↓	GPD-σ^2: 72.00%	
↓	GPD-m: 72.09%	
GPD-μσ^2mw: 72.21%	GPD-w: 72.47%	

Tab. 1: HMM-phoneme recognition rates on test data for given segmentation; left: joint training of the four types of parameters, middle: sequential training of the four types of parameters, right: only state weights discriminatively trained

ML-trained: 69.79%		
↓	GPD-μ: 73.46%	GPD-w: 72.55%
↓	GPD-σ^2: 74.16%	
GPD-μσ^2w: 76.23%	GPD-w: 74.83%	

Tab. 2: SMG-phoneme recognition rates on test data for given segmentation; left: joint training of the three types of parameters, middle: sequential training of the three types of parameters, right: only state weights discriminatively trained

Tables 1 and 2 show the phoneme recognition rate of the maximum-likelihood trained models in the top row. The result of the joint training of all types of parameters, starting from the ML-trained models, is depicted in the left column. The middle column shows the recognition rates of the sequential training of the different types of parameters, i.e. the discriminative training of the distribution means μ is performed using the ML-trained models, then the variances σ^2 of the distributions are trained keeping the means fixed, and so on. In the right column, the recognition rate of the models is depicted, when only the state weights w are trained discriminatively, starting from the ML-trained models.

The results show, that the distance in the recognition rate between HMMs and SMGs is increased by discriminative training and that the (computing time saving) joint training of all types of parameters yields approximately the same (HMM-approach) or even better results (SMG-approach) than sequential training.

The application of state weights modeling the classification relevance of the states [6] results in a significant improvement, when the state weights are the only discriminatively trained parameters (right column in tables 1 and 2). The improvement for SMGs is slightly larger than for HMMs, since SMGs have much more states, giving more discrimination power to the state weights. However, if the other parameters are trained discriminatively, too, the improvement by state weights becomes quite small (middle column, last row). The same conclusion can be drawn for variances and mixtures. This shows, that for models with state specific base functions, the means are the most important parameters to be trained discriminatively.

To evaluate the recognition *and* segmentation properties of the models trained within a given segmentation, the phoneme recognition and insertion rates of an unconstrained phoneme recognition experiment on the test data were determined. The recognition rate of the **HMM**s increased from 56.63% (ML-trained) to 59.24% (GPD, joint training of the parameters), but the insertion rate also increased from 13.40% to 14.20%. Thus,

the total improvement by discriminative training is about 1.8 points.
The recognition rate for the **SMG**s increased from 58.70% (ML-trained) to 62.35% (GPD, joint training), and the insertion rate decreased from 14.53% to 13.46%. Thus, the total improvement by discriminative training is about 4.7 points.

3.2 Discriminative training with implicit segmentation

The training parameters, γ=1.0, ε_R=100, ε_μ=0.05, ε_{σ^2} =5.0*10^{-7}, ε_m=50.0 and ε_w =0.4, have been optimized by try and error for a subset of the training database. The phoneme recognition results on the test data shown for the HMMs in table 3 and for the SMGs in table 4 have been evaluated by unconstrained phoneme recognition.

ML-trained: 56.63% (13.40%)	
↓	GPD-R: 56.75% (7.83%)
↓	GPD-μ: 57.21% (7.23%)
↓	GPD-σ^2: 57.13% (7.25%)
↓	GPD-m: 56.96% (7.15%)
GPD-R$\mu\sigma^2$mw: 56.88% (6.70%)	GPD-w: 57.30% (7.45%)

Tab. 3: Unconstrained phoneme recognition rates (phoneme insertion rates) for HMMs on test data; left: joint training of the five types of parameters, right: sequential training of the five types of parameters

ML-trained: 58.70% (14.53%)	
↓	GPD-R: 59.67% (7.66%)
↓	GPD-μ: 59.93% (6.89%)
↓	GPD-σ^2: 59.99% (6.86%)
GPD-R$\mu\sigma^2$w: 60.30% (7.63%)	GPD-w: 60.38% (6.85%)

Tab. 4: Unconstrained phoneme recognition rates (phoneme insertion rates) for SMGs on test data; left: joint training of the four types of parameters, right: sequential training of the four types of parameters

The total improvement (increase in recognition rate and decrease in insertion rate) by discriminative training is about 6.5 points for the **HMM**s and about 9 points for the **SMG**s.

A surprising result is, that the major part of the improvement is achieved by the discriminative optimization of the transition weight R. In the HMM case, the training of R reduces the insertion rate from 13.40% for R=1.0 to 7.83% for R=2.62. In the SMG case, the improvement is even larger: The insertion rate drops from 14.53% (R=1.0) to 7.66% (R=3.51), while the recognition rate rises by about one point.
Apart from the training of the means μ, which results in a total improvement of about one point in both modeling approaches, discriminative optimization of the other parameters has no significant effect on the recognition performance.

In additional experiments without the transition weight R, i.e. R=1.0 (const), we observed, that discriminative training only reduces the insertion rate without increasing the recognition rate. This shows, that for example the means μ are shifted so as to reduce the dynamic range of the emission scores for the majority of the feature vectors and not to improve the class boundaries. Thus, the optimization of the ratio between the emission and transition scores is very important.

4. IMPORTANT RESULTS

- Discriminative training improves the recognition performance of Stochastic Markov Graphs more than that of mixture-state Hidden Markov Models.
- Joint training of different types of model parameters yields approximately the same results than sequential training.
- Optimization of transition to emission weighting is important for discriminative training with implicit segmentation.
- Discriminative training with implicit segmentation concentrates on reducing the phoneme insertion rate.
- State specific emission weights improve recognition performance significantly, when they are the only discriminatively trained parameters in the models. The improvement is small, when all types of model parameters are trained discriminatively.

Acknowledgements

This work was carried out within a project supported by the German Research Foundation („Deutsche Forschungsgemeinschaft“, DFG).

References

[1] Chou W., Juang B.-H., Lee C.H., Segmental GPD Training of HMM Based Speech Recognizer, Proc. ICASSP 1992, pp. 473-476
[2] Euler S., Zinke J., Experiments on the Use of the Generalized Probabilistic Descent Method in Speech Recognition, Proc. ICSLP 1992, pp. 157-160
[3] Normandin Y., Cardin R., De Mori R., High-Performance Connected Digit Recognition Using Maximum Mutual Information Estimation, IEEE Trans. on Speech and Audio Processing, vol. 2, no. 2, April 1994
[4] Wolfertstetter F., Ruske G., Structured Markov Models for Speech Recognition, Proc. ICASSP 1995, pp. 544-547
[5] Rabiner L.R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989
[6] Wolfertstetter F., Ruske G., Discriminative State-Weighting in Hidden Markov Models, Proc. ICSLP 1994, pp. 219-222
[7] Hampshire J.B. and Waibel A.H., A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks, IEEE Trans. on Neural Networks, vol. 1, no. 2, June 1990
[8] Chou W., Lee C.-H. and Juang. B.-H., Minimum Error Rate Training Based on N-Best String Models, Proc. ICASSP 1993, pp. 652-655
[9] Chen J.-K. and Soong F.K., An N-Best Candidates-Based Discriminative Training for Speech Recognition Applications, IEEE Trans. on Speech and Audio Processing, vol. 2, no. 1, part II, pp. 206 - 216, Jan. 1994