Investigations on an EM-Style Optimization Algorithm for Discriminative Training of HMMs

Georg Heigold, *Member, IEEE*, Hermann Ney, *Fellow, IEEE*, and Ralf Schlüter, *Member, IEEE*

***Abstract*—Today's speech recognition systems are based on hidden Markov models (HMMs) with Gaussian mixture models whose parameters are estimated using a discriminative training criterion such as Maximum Mutual Information (MMI) or Minimum Phone Error (MPE). Currently, the optimization is almost always done with (empirical variants of) Extended Baum-Welch (EBW). This type of optimization requires sophisticated update schemes for the step sizes and a considerable amount of parameter tuning, and only little is known about its convergence behavior. In this paper, we derive an EM-style algorithm for discriminative training of HMMs. Like Expectation-Maximization (EM) for the generative training of HMMs, the proposed algorithm improves the training criterion on each iteration, converges to a local optimum, and is completely parameter-free. We investigate the feasibility of the proposed EM-style algorithm for discriminative training of two tasks, namely grapheme-to-phoneme conversion and spoken digit string recognition.**

***Index Terms*—Expectation-maximization, generalized iterative scaling, hidden Markov model, discriminative training.**

I. Introduction

THE acoustic models in state-of-the-art speech recognition systems are based on hidden Markov models (HMM) with Gaussian mixture models. HMMs include many free model parameters that need to be reliably estimated. Traditionally, the parameters have been optimized using the generative Maximum Likelihood (ML) training criterion [1]. More recently, discriminative training criteria for HMMs have been established in speech recognition. Several criteria including Maximum Mutual Information (MMI) [2]–[4], Minimum Classification Error (MCE) [5], [6], and Minimum Phone Error (MPE) [7] have been shown to outperform the conventional ML criterion.

Optimizing discriminative training criteria for HMMs is challenging in many respects. In the last decade, much effort has been put in finding more efficient and reliable algorithms [3], [4], [8]–[11].

Manuscript received October 22, 2012; revised March 11, 2013, July 01, 2013; accepted August 19, 2013. Date of publication August 30, 2013; date of current version October 24, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark J. F. Gales.

G. Heigold was with the is with Chair of Computer Science 6, Computer Science Department, RWTH Aachen University, D-52056 Aachen, Germany. He is now with Google, Inc., Mountain View, CA 94043 USA (e-mail: heigold@google.com).

H. Ney and R. Schlüter are with the Chair of Computer Science 6, Computer Science Department, RWTH Aachen University, D-52056 Aachen, Germany (e-mail: ney@cs.rwth-aachen.de; schlueter@cs.rwth-aachen.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2013.2280234

The de-facto standard to optimize discriminative HMMs in speech recognition is Extended Baum Welch (EBW) [12], [13] or more precisely empirical variants thereof [7], [10], [14]. The latter are a highly specialized optimization algorithm that have been shown to perform well in practice, although little has been proved concerning their convergence properties. Recently, general-purpose optimization algorithms such as Rprop [5], [15], [16] and L-BFGS [5], [17] have also been successfully used to train discriminative HMMs. The convergence of Rprop [18] and L-BFGS [17] was proved for smooth, non-convex functions. The key ingredient for these proofs is a numerical line search to satisfy certain conditions, in particular enforcing monotonic convergence. However, the complexity of the line search can be high and hard to predict. Newton's method would be attractive due to its quadratic convergence rate but is prohibitive for high-dimensional optimization problems [17].

In this paper, an EM-style algorithm is an iterative optimization algorithm that guarantees to improve the training criterion on each iteration and to converge to a local optimum. In particular, no tuning of step sizes (e.g., gradient descent), no sophisticated update schemes (*cf.* EBW), no numerical line search with hardly predictable complexity (*cf.* L-BFGS), and prohibitive storage requirement (*cf.* Newton) are required. In summary, EM-style optimization stands for a safe, simple and easy to implement optimization algorithm such as EM for the generative training of HMMs.

Unlike Expectation-Maximization (EM) [19] for generative HMMs, there is no similar optimization algorithm known for discriminative training of HMMs. Generalized Iterative Scaling (GIS) [20], [21], for example, is an EM-style optimization algorithm for log-linear models. However, GIS does not directly apply to discriminative HMMs due to the different parameterization and because it does not allow for hidden variables.

This defines the scope of this paper. In particular, the main contributions include:

- Step-by-step derivation of a constructive[1], EM-style algorithm for discriminative training of Gaussian Mixture Models (GMMs) in Section V and Hidden Markov Models (HMMs) in Section VI. We shall call the proposed algorithm Generalized GIS (G-GIS).
- Experimental evaluation of the proposed algorithm without heuristics for an optical character recognition task using discrete-valued features and mixture models (Section VII-A), a grapheme-to-phoneme task with short sequences using discrete-valued features and HMMs (Section VII-B), and spoken digit string recognition with

[1]Constructive means that all quantities are explicitly known and can be efficiently computed.

longer sequences using continuous-valued features and HMMs (Section VII-C).

It was proved that EBW is an EM-style algorithm for sufficiently large iteration constants, see [22] (discrete-valued features) and [13], [14] (continuous-valued features). Both [13] and [14] do not make an explicit statement on what "sufficiently large" means in this context. The empirical iteration constants used in practice are set to guarantee well-defined Gaussian mixture models, including non-negative mixture weights and positive variances [7], [10], [14]. However, these do not imply an increase of the training criterion in general. Experiments based on the reverse Jensen inequality [23] were shown in [24] using many approximations and heuristics. In particular, all HMM dependency is ignored by approximating the sentence-level training criterion with a frame-level training criterion. An algorithm similar to ours was proposed for log-linear models with hidden variables in [25], [26], reporting on experiments for small tagging tasks using discrete-valued features. The current paper is based on the two previous conference papers [27], [28] proposing a similar algorithm for log-linear models and conditional random fields including hidden variables. There is also some overlap with our publication on the equivalence of Gaussian and log-linear models [29] although the derivation for the transition probabilities is different.

The remainder of this paper is organized as follows. Section II gives a brief review on bounds and EM-style optimization. Section III discusses generative models and the EM algorithm. Section IV summarizes log-linear models and the GIS algorithm. Section V and Section VI derive bounds for discriminative Gaussian mixture models and HMMs, based on the bounds from the previous sections. Section VII compares the proposed with existing optimization algorithms on different tasks. The paper is concluded in Section VIII.

II. Concept of Lower Bounds

In this section, we briefly review the concept of EM-style optimization, based on the notion of auxiliary functions in the strong sense [7] or lower bounds ([30], Chapter 9). Two well-known examples are Expectation-Maximization (EM) (Section III) and Generalized Iterative Scaling (GIS) (Section IV).

Assume a training criterion $G(\theta)$ bounded from above to be maximized. A lower bound $L_{\theta'}(\theta)$ of the training criterion $G(\theta)$ is a smooth function in θ that bounds G below with contact in θ'

$$G(\theta) - G(\theta') \geq L_{\theta'}(\theta) \quad \text{and} \quad L_{\theta'}(\theta') = 0. \tag{1}$$

Fig. 1 depicts an example.

EM-style optimization comprises two steps: find the lower bound based on the old parameters θ' (E-step) and maximize this lower bound to obtain the updated parameters θ (M-step). The lower bound is used as an approximation to the training criterion to make the original, hard optimization tractable. These two steps ("an iteration") are repeated until convergence. It can be shown [31] that this algorithm monotonically increases the training criterion and converges to a critical point (e.g., local maximum) of the training criterion.

In practice, lower bounds should be tight and simple, for example, should decouple the parameters and are analytically tractable. This leads to "fast" convergence at low cost per iteration, without the need to tune parameters.

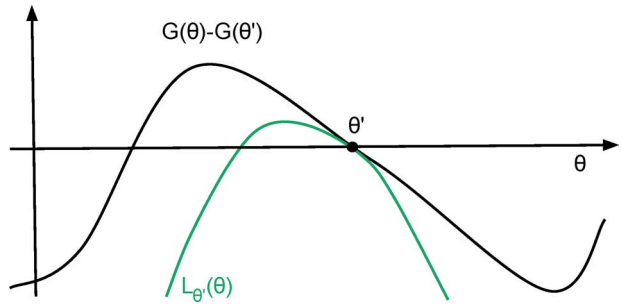

Fig. 1. Illustration of a lower bound $L_{\theta'}(\theta)$ for the training criterion $G(\theta)$ at θ': the lower bound is in the hypograph of the training criterion with contact in θ'.

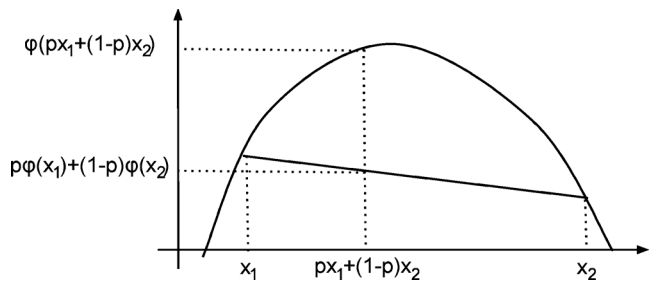

Fig. 2. Illustration of Jensen's inequality for two points x_1, x_2 and distribution $\pi_1 = p, \pi_2 = 1 - p$.

EM [19] (see also Section III for a brief review) probably is the most prominent example for an algorithm with these properties. This is why we call this type of optimization EM-style optimization. Growth transformations [12], [13] generalize this concept.

The main tool to derive lower bounds for discriminative GMMs and HMMs will be Jensen's inequality. In the context of probability theory, it is generally stated in the following form: if $\{x_i\}_i$ is a set of variables, $\{\pi_i\}_i$ a distribution, and ϕ a concave function, then

$$\phi\left(\sum_i \pi_i x_i\right) \quad \geq \sum_i \pi_i \phi(x_i) \tag{2}$$

([30], p.56). Fig. 2 illustrates this inequality for two points $x_1, x_2 \in \mathbb{R}$.

III. Gaussian Mixture Models (Generative)

The Gaussian Mixture Model (GMM) is a generative model which assigns the following likelihood to the feature vector x given the class index c

$$p_\theta(x|c) = \sum_l p_\theta(x, l|c) = \sum_l p(l|c)\mathcal{N}(x|\mu_{cl}, \Sigma_{cl}). \tag{3}$$

This is a superposition of Gaussian densities $\mathcal{N}(\cdot|\cdot)$ with weights $p(l|c)$. The model parameters θ include the means μ_{cl}, the covariance matrices Σ_{cl}, and the weights $p(l|c)$ where l denotes the mixture index. These parameters are subject to the constraints such as the normalization of the conditional probabilities and the positivity of the variances.

Traditionally, EM is used for ML training of Gaussian mixture models

$$G(\theta) = \log p_\theta(x|c) = \log \sum_l p_\theta(x, l|c). \tag{4}$$

For simplicity, the training criterion $G(\theta)$ is defined per observation (x, c). The training criterion over labeled training data $\{x_n, c_n\}_{n=1}^N$ is obtained by accumulation. The lower bound for the ML training criterion associated with EM, $L_{\theta'}(\theta)$, reads

$$G(\theta) - G(\theta') \geq \sum_l p_{\theta'}(l|x, c) \log \frac{p_\theta(x, l|c)}{p_{\theta'}(x, l|c)} =: L_{\theta'}(\theta). \tag{5}$$

Here, $p_{\theta'}(l|x, c)$ denotes the conditional probability of the mixture index l conditioned on the feature vectors x and the class index c. This quantity is derived from the joint probability by Bayes rule, $p_{\theta'}(l|x, c) = \frac{p_{\theta'}(x,l|c)}{\sum_{l'} p_{\theta'}(x,l'|c)}$. For GMMs, the lower bound can be analytically optimized ([30] Chapter 9).

EM is a general-purpose optimization technique for generative training of generative models. Generalized Iterative Scaling (GIS) is a similar technique for discriminative training of log-linear models but not generative models such as GMMs.

IV. Log-Linear Models

Log-linear models are discriminative models that directly parameterize the class-posterior

$$p_\lambda(c|x) = \exp\left(\sum_i \lambda_i f_i(x, c)\right) / Z(\lambda). \tag{6}$$

Each feature function $f_i(x, c)$ is weighted by some λ_i. The model parameters are $\lambda = \{\lambda_i\}$. The normalization constant is denoted by $Z(\lambda)$. Log-linear models are optimized using the Maximum Conditional Likelihood criterion, which is traditionally known in the speech community as Maximum Mutual Information (MMI)

$$G(\lambda) = \log p_\lambda(c|x). \tag{7}$$

GIS was originally introduced by [20]. It applies to log-linear models that are optimized according to (7). A lower bound for this configuration is given in [21]

$$G(\lambda) - G(\lambda') \geq \sum_i \Delta\lambda_i f_i(x, c) - \frac{1}{F} \exp(\Delta\lambda_i F) \sum_{c'} p_{\lambda'}(c'|x) f_i(x, c') + \text{const}(\lambda) =: L_{\lambda'}(\lambda)$$

with $\Delta\lambda_i = \lambda_i - \lambda'_i$. For this bound, we assume without loss of generality[2] that the feature functions $f_i(x, c)$ are non-negative and sum up to the feature constant

$$F = \sum_i f_i(x, c), \quad \forall x, c,$$

which does not depend on the class index c and the observation x. Additive terms that do not depend on λ, $\text{const}(\lambda)$, are not explicitly shown for simplicity.

[2]These assumptions can always be imposed by a suitable affine feature transform and the introduction of a dummy feature, without changing the model.

For labeled training data $\{x_n, c_n\}_{n=1}^N$, the optimum of the accumulated bound, $\nabla L_{\lambda'}(\lambda) = 0$, is attained for

$$\lambda_i = \lambda'_i + \frac{1}{F} \log \frac{N_i}{Q_i(\lambda')}.$$

Here, the numerator and denominator statistics are defined as

$$N_i = \sum_{n=1}^N f_i(x_n, c_n)$$

$$Q_i(\lambda') = \sum_{n=1}^N \sum_c p_{\lambda'}(c|x_n) f_i(x_n, c).$$

These equations are referred to as update rules.

GIS cannot be directly applied to discriminative GMMs (Section V) or HMMs (Section VI) because GIS is only for log-linear models, *i.e.*, for models with a different parameterization and without hidden variables.

V. Gaussian Mixture Models (Discriminative)

In this section, we derive a lower bound for GMMs in (3) that are estimated with MMI in (7). The class-posterior is determined from the joint probability by Bayes rule, including the class-prior $p(c)$

$$p_\theta(c|x) = \sum_l p_\theta(c, l|x) = \frac{\sum_l p(c) p_\theta(x, l|c)}{\sum_{c',l'} p(c') p_\theta(x, l'|c')}. \tag{8}$$

This will be accomplished in three steps: 1) EM is applied to "eliminate" the hidden variables, 2) the GMM is rewritten in the log-linear parameterization, and 3) GIS is applied to the re-parameterized model to obtain the final bound. This material is based on our previous work in [27], [28].

A. Step 1 (EM)

In the first step, the lower bound from Section III is used with the variables x and c swapped to give a bound for the class-posteriors ((8)) and not the likelihoods ((3)). The resulting intermediate bound $L_{\theta'}(\theta)$ does not have any hidden variables. The difference of the training criterion (see (1)) is re-arranged and bounded from below as follows:

$$\begin{aligned}
G(\theta) - G(\theta') &\overset{\text{Eq.}(8)}{=} \log \sum_l \frac{p_\theta(c, l|x)}{\sum_{l'} p_{\theta'}(c, l'|x)} \\
&= \log \sum_l \frac{p_{\theta'}(c, l|x)}{\sum_{l'} p_{\theta'}(c, l'|x)} \frac{p_\theta(c, l|x)}{p_{\theta'}(c, l|x)} \\
&= \log \sum_l p_{\theta'}(l|x, c) \frac{p_\theta(c, l|x)}{p_{\theta'}(c, l|x)} \\
&\overset{\text{Eq.}(2)}{\geq} \sum_l p_{\theta'}(l|x, c) \log \frac{p_\theta(c, l|x)}{p_{\theta'}(c, l|x)} \\
&= \sum_l p_{\theta'}(l|x, c) \log p_\theta(c, l|x) + \text{const}(\theta) \\
&=: L_{\theta'}(\theta).
\end{aligned} \tag{9}$$

Additive terms that do not depend on θ are not explicitly shown to highlight the relationship of this bound with the training criterion in (7). This bound is an example for Generalized EM

([30], Chapter 9.4), [32]. In general, it implies a non-trivial optimization problem that requires numerical optimization (what we want to avoid in our EM-style optimization algorithm). This bound holds true independent of the model and also for non-negative, unnormalized quantities such as $p_\lambda(c,l|x) = \exp(\sum_i \lambda_i f_i(x,c,l))$ [19], see the next step.

B. Step 2 (Log-Linear Parameterization)

Next, the parameterization of the posterior of Gaussians in (8) is changed to the log-linear, see (6) in Section IV

$$p_\lambda(c,l|x) = \exp\left(\sum_i \lambda_i f_i(x,c,l)\right)/Z(\lambda) = \exp\left(\sum_{d',d} \lambda^{(2)}_{cld'd} x_{d'} x_d + \sum_d \lambda^{(1)}_{cld} x_d + \lambda^{(0)}_{cl}\right)/Z(\lambda). \tag{10}$$

The first line describes a log-linear model with general feature functions $f_i(x,c,l)$. The second line uses polynomial features of the type $f^{(2)}_{c'l',d'd}(x,c,l) = \delta(c',c)\delta(l',l)x_{d'}x_d$ (from variances), $f^{(1)}_{c'l',d}(x,c,l) = \delta(c',c)\delta(l',l)x_d$ (from means), and $f^{(0)}_{c',l'}(x,c,l) = \delta(c',c)\delta(l',l)$ (from priors). In this example, the index i stands for index tuples, $(c'l',d'd)$, $(c'l',d)$, (c',l'). Furthermore, most of the feature functions are filtered out by the Kronecker delta and are discarded. The log-linear parameters $\lambda = \{\lambda^{(2)},\lambda^{(1)},\lambda^{(0)}\}$ can be obtained from the Gaussian parameters θ by comparison of terms quadratic, linear, and constant in x, see for example [33].

For our purpose, however, representing a GMM as a log-linear model is not sufficient because this may add flexibility to the model (for example, positive variances or normalized priors are no longer guaranteed) such that we no longer optimize GMMs. It was shown in [29], [34] that (3) and (10) represent equivalent models and thus, the two parameterizations can be used interchangeably for the same underlying model. The back transformation is shown and further discussed in Section V-D.

C. Step 3 (GIS)

The intermediate lower bound from the first step in (9) using the log-linear parameterization from the second step in (10) is a superposition of training criteria of the type discussed in Section IV with non-negative weights. Hence, the lower bound for GIS applies

$$G(\lambda)-G(\lambda') \geq \sum_i \sum_l p_{\lambda'}(l|x,c)\Delta\lambda_i f_i(x,c,l) - \sum_{c',l} p_{\lambda'}(c',l|x)\frac{\exp(F\Delta\lambda_i)}{F} f_i(x,c',l) + \text{const}(\lambda) =: L_{\lambda'}(\lambda)\,. \tag{11}$$

with $\Delta\lambda_i = \lambda_i - \lambda'_i$. Similar to GIS, it is assumed without loss of generality that the feature functions are non-negative and sum up to the feature constant, $\sum_i f_i(x,c,l) = F,\ \forall x,c,l$. The lower bound in (11) is proved by observing that the first line corresponds with the numerator of the posterior (see (10)) in (9) and the second line is an upper bound for the denominator of the posterior, *i.e.*, the normalization constant $Z(\lambda)$.

$$\begin{aligned}
Z(\lambda)-Z(\lambda') &\overset{\text{Eq.(9)}}{=} \underbrace{\sum_l p_{\lambda'}(l|x,c)}_{=1} \log \frac{\sum\limits_{c',l'} \exp(\sum\limits_i \lambda_i f_i(x,c',l'))}{\sum\limits_{c',l'} \exp(\sum\limits_i \lambda'_i f_i(x,c',l'))} \\
&\overset{\text{Eq.(10)}}{=} \log \sum_{c,l} p_{\lambda'}(c,l|x) \exp\left(\sum_i \Delta\lambda_i f_i(x,c,l)\right) \\
&\overset{\log x \leq x-1}{\leq} \sum_{c,l} p_{\lambda'}(c,l|x) \exp\left(\sum_i \Delta\lambda_i F \frac{f_i(x,c,l)}{F}\right) + \text{const}(\lambda) \\
&\overset{\text{Eq.(2)}}{\leq} \sum_{c,l} p_{\lambda'}(c,l|x) \sum_i \frac{f_i(x,c,l)}{F} \exp(\Delta\lambda_i F) + \text{const}(\lambda)
\end{aligned}$$

Finally, we plug in the specific feature functions for discriminative GMMs (see Section V-B) into (11). The parameters are decoupled in the lower bound such that $L_{\lambda'}(\lambda) = L_{\lambda'}(\lambda^{(0)}) + L_{\lambda'}(\lambda^{(1)}) + L_{\lambda'}(\lambda^{(2)})$ with, for example,

$$L_{\lambda'}(\lambda^{(1)}) = \sum_{l,d} p_{\lambda'}(l|x,c)\Delta\lambda^{(1)}_{cld} x_d - \sum_{c'} p_{\lambda'}(c',l|x)\frac{\exp(F\Delta\lambda^{(1)}_{c'ld})}{F} x_d + \text{const}(\lambda)$$

where $\Delta\lambda^{(1)}_{cld} = \lambda^{(1)}_{cld} - \lambda'^{(1)}_{cld}$. Accumulating over labeled training data $\{x_n,c_n\}_{n=1}^N$ and solving $\nabla L_{\lambda'}(\lambda) = 0$ for λ provides the update rules, for example,

$$\lambda^{(1)}_{cld} = \lambda'^{(1)}_{cld} + \frac{1}{F}\log\frac{N^{(1)}_{cld}(\lambda')}{Q^{(1)}_{cld}(\lambda')}. \tag{12}$$

These are functions of the numerator and denominator statistics

$$N^{(1)}_{cld}(\lambda') = \sum_{n=1}^N \delta(c,c_n) p_{\lambda'}(l|x_n,c) x_{nd} \tag{13}$$

$$Q^{(1)}_{cld}(\lambda') = \sum_{n=1}^N p_{\lambda'}(c,l|x_n) x_{nd} \tag{14}$$

etc., and the feature constant

$$F = 1 + \sum_d x_{nd} + \sum_{d,d'} x_{nd}x_{nd'}. \tag{15}$$

D. Back Transformation

The back transformation from the log-linear to the Gaussian parameterization is given by [29], [34]:

$$\begin{aligned}
&1.\ \tilde{\lambda}_{cl}^{(2)} \leftarrow \lambda_{cl}^{(2)} + \Delta\lambda^{(2)} \\
&2.\ \Sigma_{cl} = -\frac{1}{2}(\tilde{\lambda}_{cl}^{(2)})^{-1} \\
&3.\ \mu_{cl} = \Sigma_{cl}\lambda_{cl}^{(1)} \\
&4.\ \tilde{\lambda}_{cl}^{(0)} \leftarrow \lambda_{cl}^{(0)} + \frac{1}{2}\left(\mu_{cl}^{\top}\Sigma_{cl}^{-1}\mu_{cl} + \log|2\pi\Sigma_{cl}|\right) \\
&5.\ p(l|c) = \exp(\tilde{\lambda}_{cl}^{(0)}) / \sum_{l'} \exp(\tilde{\lambda}_{cl'}^{(0)}) \\
&6.\ p(c) = \sum_{l'} \exp(\tilde{\lambda}_{cl'}^{(0)}) / \sum_{c',l'} \exp(\tilde{\lambda}_{c'l'}^{(0)})
\end{aligned} \tag{16}$$

In Step 1, a matrix $\Delta\lambda^{(2)}$ that does not depend on c, l is added to make $\lambda_{cl}^{(2)}$ negative-definite and thus, the covariance matrix Σ_{cl} positive-definite (see Step 2). This operation leaves the posterior unchanged as it adds the same constant factor in the numerator and denominator of the posterior in (10). In Step 4, the normalization constant of the Gaussian density is added to $\lambda_{cl}^{(0)}$. Another constant factor is added in Step 5 to normalize the mixture weights, which in turn is propagated to Step 6. The normalization constant in Step 6 cancels in the posterior.

It may be an interesting exercise to translate the G-GIS update rules into Gaussian space. This is not necessary due to the equivalence. Furthermore, it will not be straightforward because the back transformation is not unique due to $\Delta\lambda^{(2)}$ in Step 1.

Finally, we extend this result to HMMs and discuss possible refinements and the limitations.

VI. Hidden Markov Models (Discriminative)

In this section, we present a constructive, lower bound for discriminative training of HMMs.

Let x_1^T be a sequence of feature vectors $x_t \in \mathbb{R}^D$ and s_1^T a sequence of states $s_t \in \{0, \ldots, S-1\}$. The joint probability of x_1^T and s_1^T of a first-order Markov model is decomposed as follows

$$p\left(x_1^T, s_1^T\right) = \prod_{t=1}^{T} p(s_t|s_{t-1})p(x_t|s_t). \tag{17}$$

Discrete conditional distributions are used for the transition probabilities $p(s|s')$. For discrete-valued features, discrete conditional distributions can be used for the emission probabilities $p(x|s)$ as well. For continuous-valued features, the Gaussian mixture model is the common choice to represent the emission probabilities, $p(x|s) = \sum_l p(l|s)\mathcal{N}(x|\mu_{sl}, \Sigma_{sl})$.

The joint probability of a word or sentence is obtained by marginalization over the state sequences s_1^T representing W

$$p(x_1^T, W) = \sum_{s_1^T \in W} p(x_1^T, s_1^T).$$

Note that the symbol W is interchangeably used to denote the sentence or the set of state sequences that represent the sentence. This model is referred to as hidden Markov model (HMM). The model parameters are the Gaussian means, covariance matrices, the mixture weights, and the transition probabilities, $\theta = \{\mu_{sl}, \Sigma_{sl}, p(l|s), p(s|s')\}$. Traditionally, HMMs are generatively trained using ML in (4) [1]. More recently, discriminative training (e.g., MMI in (7)) for HMMs has been established with good success in speech recognition.

The same ideas as in Section V can be used to derive a formal, lower bound for HMMs, using the substitutions $x \rightarrow x_1^T, l \rightarrow s_1^T$, and $c \rightarrow W$.

A. Step 1 (EM)

The derivation in (9) can be used.

B. Step 2 (Log-Linear Parameterization)

Equivalence of the posteriors of HMMs in (17) and a log-linear model of the type

$$\prod_{t=1}^{T} \exp\left(\lambda_{s_{t-1}s_t}^{(0)} + \sum_d \lambda_{s_t d}^{(1)} x_{td} + \sum_{d,d'} \lambda_{s_t dd'}^{(2)} x_{td} x_{td'}\right) / Z(\lambda). \tag{18}$$

was proved in [29], [34] and thus, the two parameterizations of the HMM can be interchangeably used. Section VI-D shows the back transformation for further discussion.

The decoding is affected by the parameterization only through the feature scorer, i.e., how the emission probabilities, $p(x|s)$, are computed. In case of the log-linear parameterization, the feature scorer returns the value of the exponential function in (18).

C. Step 3 (GIS)

Applying GIS results in update rules similar to those in Section V, for example,

$$\lambda_{sd}^{(1)} = \lambda_{sd}^{\prime(1)} + \frac{1}{F}\log\frac{N_{sd}^{(1)}(\lambda')}{Q_{sd}^{(1)}(\lambda')}. \tag{19}$$

Again, these are based on the sufficient statistics

$$N_{sd}^{(1)}(\lambda') = \sum_{t=1}^{T} p_{\lambda'}\left(s_t = s|x_1^T, W\right) x_{td} \tag{20}$$

$$Q_{sd}^{(1)}(\lambda') = \sum_{t=1}^{T} p_{\lambda'}\left(s_t = s|x_1^T\right) x_{td} \tag{21}$$

and the feature constant

$$F = \sum_{t=1}^{T}\left(1 + \sum_d x_{td} + \sum_{d,d'} x_{td}x_{td'}\right). \tag{22}$$

The symbols s and x_{td} stand for the HMM state and the d-th component of the feature vector x_t, respectively. The statistics are the features weighted by the probability of state s at frame t conditioned on the feature vectors x_1^T, and the sentence W in case of the numerator statistics, $N_{sd}(\lambda')$. These posteriors can be efficiently computed using the forward/backward algorithm [3], [9].

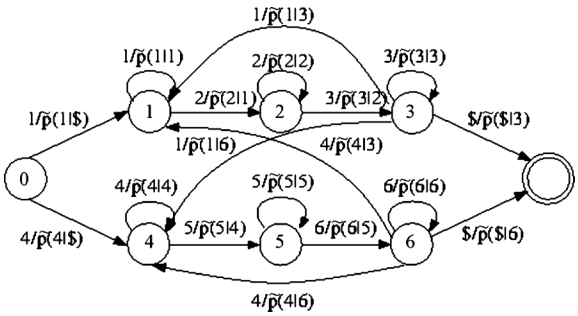

Fig. 3. Network representing a transition model with loop and forward transitions, the arc labels $s/\tilde{p} \in \{1, \ldots, 6, \$\} \times \mathbb{R}$ denote the HMM state and the (unnormalized) transition weight, respectively. $\$ \rightarrow 1/4$ and $3/6 \rightarrow \$$ implement the entry and exit transitions.

The update rules are based on the same sufficient statistics as used for other, gradient-based optimization algorithms such as EBW, L-BFGS, and Rprop. The overall training complexity per iteration is dominated by computing (linear to the number of arcs and the number of states in the network) and accumulating (linear to the number of training observations and the number of active features per observation) the sufficient statistics. Hence, the overall complexity per iteration is basically the same for G-GIS, EBW, Rprop, and L-BFGS (ignoring the line searches). The consumed wall clock time, however, can be different due to the different convergence speed (i.e., the number of iterations to converge). The storage requirement for G-GIS, EBW, Rprop is comparable, whereas L-BFGS requires considerably more storage for the previous gradients used to approximate the Hessian information.

This provides a constructive, EM-style algorithm with finite step sizes for sequences of finite length. However, the feature constant in (22) scales with the length of the sentence. Thus, the actual steps are inversely proportional to the sentence length and approach zero in the limit of infinitely long sentences. The vanishing step sizes seem to be an artifact of GIS and not a fundamental problem with HMMs. Armijo's approach [27], [35], for example, is probably the most naive EM-style algorithm whose step sizes are given by the inverse Lipschitz constant of the gradient, i.e., the maximum possible eigenvalue of the Hessian matrix. It can be shown that under some assumptions, the Hessian matrix never diverges, also not in the limit of infinite long sequences. This is similar to the fact that a one-dimensional Ising model cannot have any phase transitions [36].

D. Back Transformation

The back transformation is performed step-by-step, similar to the mixture models above [29], [34]. First, the emission probability, $p(x|s)$ in (17), is processed using the first five steps from (16). The normalization constant from Step 5 is passed to the respective transition parameter $\lambda^{(0)}_{s's}$ in (18). A transition model of the type shown in Fig. 3 is assumed. Here, we explicitly construct analytical expressions for the conditional probabilities, avoiding solving an eigenvalue problem as for the approach in [29], [34]. Without loss of generality (adding a constant to all transition parameters does not change the posterior), we assume that all exponentiated transition parameters are smaller than 1, $\exp(\lambda^{(0)}_{s's}) =: \tilde{p}(s|s') < 1$. Then, the transition probabilities, $p(s|s')$ in (17), are obtained by marginalization and Bayes rule ([34], Chapter 4.6.3)

$$p(s|s') = \frac{Z_s}{Z_{s'}} \tilde{p}(s|s') \tag{23}$$

where Z_s denotes the backward probability of node s in the transition network. In general, these quantities can be efficiently computed by recursion. For the sparse transition network in Fig. 3, an analytical solution for the backward probabilities exist, see Appendix. The remaining normalization constant of the transition network, Z_0, cancels in the posterior.

Again, it may be tempting to translate the G-GIS update rules into Gaussian space. We refrain from doing it because it is not necessary, and the back transformation is ambiguous and hardly analytically tractable (beside the loop and forward transitions, skip transitions can also be allowed, for example).

E. Extensions

This result can be extended to training criteria in the rational form

$$G(\lambda) = \log \frac{\sum_{s_1^T} a\left(s_1^T\right) p_\lambda\left(s_1^T | x_1^T\right) Z(\lambda)}{\sum_{s_1^T} b\left(s_1^T\right) p_\lambda\left(s_1^T | x_1^T\right) Z(\lambda)} \tag{24}$$

where a, b denote non-negative weights. The proof is a simple extension of the proof above for MMI. This generalized training criterion gives us the flexibility to represent many of the common training criteria investigated in speech recognition, including

- MMI: $a(s_1^T) = \delta(s_1^T \in W), b(s_1^T) = 1$,
- Minimum Phone Error (MPE) [7]: $a(s_1^T) = A(s_1^T|W)$ where A is the phone accuracy between s_1^T and the correct hypothesis W[3], $b(s_1^T) = 1$, and
- margin-based training as introduced in [34]: $a(s_1^T) \leftarrow a(s_1^T)\exp(-\rho A(s_1^T|W)), b(s_1^T) \leftarrow b(s_1^T)\exp(-\rho A(s_1^T|W))$ with ρ some scaling factor, i.e., the weights are multiplied with the margin term in addition.

In this case, the sufficient statistics for the update rules in (19) are of the form

$$N^{(1)}_{sd}(\lambda') = \frac{\sum_{s_1^T} p_{\lambda'}\left(s_1^T | x_1^T\right) a\left(s_1^T\right) \sum_{t=1}^T \delta(s_t, s) x_{td}}{\sum_{s_1^T} p_{\lambda'}\left(s_1^T | x_1^T\right) a\left(s_1^T\right)}$$

$$Q^{(1)}_{sd}(\lambda') = \frac{\sum_{s_1^T} p_{\lambda'}\left(s_1^T | x_1^T\right) b\left(s_1^T\right) \sum_{t=1}^T \delta(s_t, s) x_{td}}{\sum_{s_1^T} p_{\lambda'}\left(s_1^T | x_1^T\right) b\left(s_1^T\right)}.$$

The feature count is the same as in (22). These statistics can be efficiently computed using the forward/backward algorithm in combination with the expectation semiring [16], [37], [38].

VII. Experimental Results

The proposed algorithm in (12)–(15) for GMMs and (19)–(22) for HMMs, G-GIS for short, is applied to three tasks from optical character recognition, grapheme-to-phoneme conversion, and spoken digit string recognition. Discriminative

[3]We can always add a sufficiently large constant to the training criterion to make all accuracies non-negative.

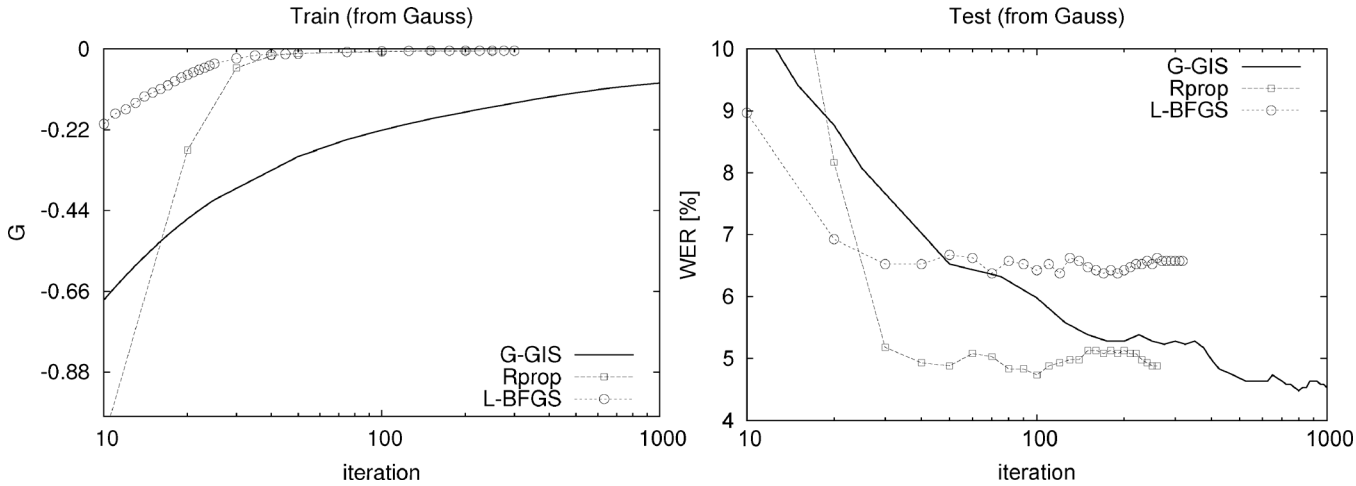

Fig. 4. Comparison of different optimization algorithms (G-GIS, Rprop, L-BFGS) for mixture models using MMI on USPS task. Left: evolution of G on training corpus. Right: evolution of character error rate (CER) on test corpus.

GMMs are used for the first task while for the latter two tasks, discriminative HMMs are used. G-GIS is compared to Rprop [15], L-BFGS [17], and EBW [10] if applicable.

Since all these optimization algorithms make use of exactly the same sufficient statistics of the data, the computing time per iteration is comparable, see Section VI-C. L-BFGS can and does recompute the gradient during the line search. This effect, however, is negligible in our experiments. Thus, a comparable number of iterations implies a comparable wall clock time.

A. Optical Character Recognition

The well-known USPS handwritten digit database consists of isolated and normalized images of handwritten digits taken from US mail envelopes scaled to 16×16 pixels. The database contains a separate training and test set, with 7,291 and 2,007 images, respectively[4]. The US Postal Service task is still one of the most widely used reference data sets for handwritten character recognition and allows fast experiments due to its small size. The test set contains a large amount of image variability and is considered to be a "hard" recognition task. Good error rates are in the range of 2–3% and use advanced modeling techniques, e.g. deformation models [39].

Here, we use GMMs with 16 components for each digit in combination with the gray-scale features augmented with Sobel-based derivatives, amounting to a total of 512 features. The model is optimized using MMI with ℓ_2-regularization. Comparative results are shown in Fig. 4 for different optimization algorithms (G-GIS, Rprop, L-BFGS). MMI training is initialized with the ML estimate.

The convergence speed (and thus the computation time) for G-GIS, Rprop, and L-BFGS is comparable, although G-GIS tends to be slower than Rprop and L-BFGS. This is not surprising because G-GIS is derived for the worst case scenario. Furthermore, G-GIS achieves the same test error rates as Rprop whereas L-BFGS seems to get stuck in a sub-optimal local maximum, see Table I.

[4]Data available from ftp://ftp.kyb.tuebingen.mpg.de/pub/bs.

TABLE I
ERROR RATES (ER) ON USPS TEST CORPUS FOR DIFFERENT OPTIMIZATION ALGORITHMS

optimization	ER [%] (MMI)
G-GIS	4.7
Rprop	4.9
L-BFGS	6.4

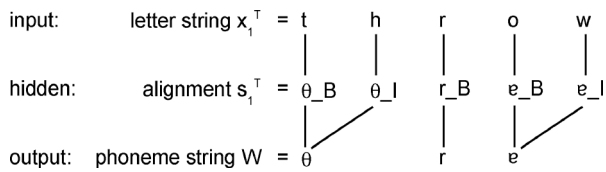

Fig. 5. Example of a word/pronunciation alignment.

B. Grapheme-to-Phoneme Conversion

Grapheme-to-phoneme conversion is an important task to build state-of-the-art speech recognition systems. It is used to enrich the pronunciation lexicon by words where the lexical representation is known, but not the phonetic representation. In this example, the letters are the (discrete-valued) features x_1^T and the phoneme string representing a word from the lexicon corresponds with W, see Section VI. In general, multiple letters are consumed to describe one phoneme. To align the two strings, we adopt the so-called Begin-Inside-Out scheme [40]. An example of a possible alignment for a word/pronunciation pair using this scheme is depicted in Fig. 5.

For the reported experiments, the English NETtalk 15 k corpus has been chosen. It consists of 26 different graphemes and 50 phonemes. The data set is partitioned into roughly 14 k training words (Train), 1 k words for development and tuning of the system (Dev), and 5 k testing words (Eva). As a common trick, we represent the letters in a symmetric window of size nine around the current letter as vectors, which are used as the features. A bigram model is used on the output side. All model parameters are updated with MMI.

Fig. 6 shows the evolution of the training criterion and the phoneme error rate over the training iterations. In this example, the training criterion increases monotonically for all three optimization algorithms. This is guaranteed for G-GIS. L-BFGS

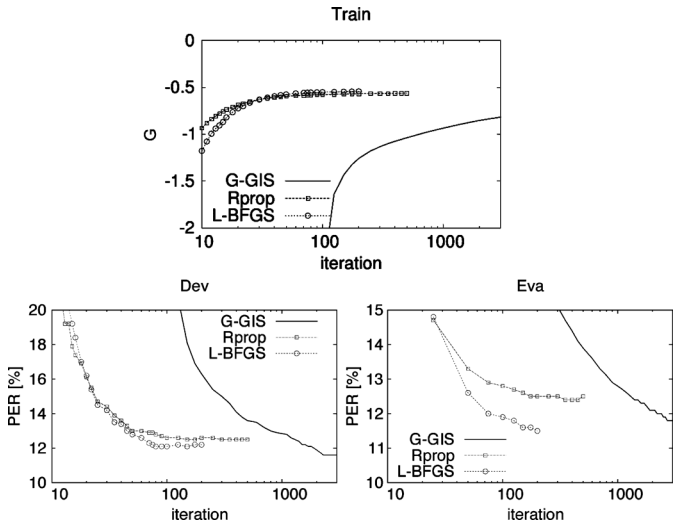

Fig. 6. Comparison of different optimization algorithms (G-GIS, Rprop, L-BFGS) for NETtalk. Top: evolution of training criterion G on training data. Bottom: evolution of Phoneme Error Rate (PER) on dev data (left) and eval data (right).

spends some additional iterations in the first few iterations on the line search to make sure that the training criterion increases. For Rprop, this just happens to be true for this example. The algorithms seem to converge to different but comparable local optima. Both L-BFGS and Rprop converge within a few hundred iterations. In contrast, G-GIS takes about ten times more iterations to converge.

What is the reason for the relatively slow convergence of G-GIS compared to L-BFGS? The update rules for G-GIS are constructed such as to always improve the training criterion, also in the worst case. In contrast, L-BFGS implements a "trial-and-error" approach: start with a reasonable step size and reduce it until the current iteration improves over the last iteration. In general, this can be costly (up to ten trials in the given example, i.e., the actual step size is 210 times smaller than the initial). In practice, however, this only occurs rarely such that the overall training time is hardly affected. This analysis suggests that the convergence rate for G-GIS is reasonable for a global step size parameter such as the feature count F.

C. Spoken Digit String Recognition

This example is used to test G-GIS for continuous-valued features. We use the SieTill task which consists of spoken digit strings [41]. The recognition system is based on whole-word HMMs with 215 distinct states in total. The vocabulary consists of the German digits, including a pronunciation variant. The feature vectors consist of twelve cepstral features without temporal derivatives. The linear discriminant analysis (LDA) is applied to five consecutive frames and projects the concatenated features to a 25-dimensional vector. The training and the test corpus consist of approximately 2.5 h audio data/10 k spoken digits each. The ML baseline system uses single Gaussians with globally pooled variances to keep the complexity low. As usual in speech recognition, the transition probabilities are not updated in MMI training. To be precise, this leads to some slight inconsistency of the Gaussian and log-linear parameterization. This is not considered critical in this context because the focus is on the convergence behavior rather than the error rate. In addition, control experiments on this task suggest that updating the transition model does not make a significant difference.

Fig. 7 shows the evolution of the training criterion and the word error rate over the training iterations. A few comments are due.

The progress of conventional MMI training using the de-facto standard EBW [10] is shown in Fig. 7 for comparison. This is the typical performance of EBW we observe for HMMs with Gaussian models using globally pooled variances [16]: relatively slow convergence compared to systems using untied variances [10] and the training criterion is not monotonically increasing. This is because the empirical iteration constants are set such as to make the Gaussian models and the HMMs well-defined [7], [10], [14] which, however, is not sufficient to improve the training criterion in general. This is only guaranteed for "sufficiently large" iteration constants [13], [14]. Even worse, the convergence behavior of empirical EBW has not been studied in the literature, including convergence to a critical point or if it converges at all.

In this example, L-BFGS converges quickly and more reliably. The training criterion increases on each iteration by construction. Repeating the iteration with a smaller step size typically only happens in the first few iterations. Thus, the line search has only a small impact on the overall training time. The refined optimization around the optimum, however, does not pay off in terms of the word error rate.

After a few iterations to adapt the step sizes, Rprop converges quickly. Although not guaranteed, the training criterion improves on each iteration in this example.

As proved, G-GIS improves the training criterion in each iteration and converges smoothly to the same word error rate as the other optimization algorithms, see Table II. However, this appears to be at the expense of a considerably slower convergence: the computing time of G-GIS is 1,000 times larger than for the other three optimization algorithms.

D. Scalability & Practicability

This section addresses the question of the application of the proposed algorithm in practice and for large-vocabulary continuous speech recognition.

Inspection of the feature constant ((15) and (22)) suggests that the convergence speed of G-GIS generally depends on the length of the longest training sequence and the type of features (for example, bounded/unbounded or sparse/dense). Also, the amount of training data will affect the convergence speed of G-GIS. This is because the maximum in the feature constant ((15) and (22)) is sensitive to "outliers", which are more likely for larger amounts of data. These properties of the feature constant make it hard to scale the current version of G-GIS beyond the examples given in this paper. However, this is most likely not a fundamental problem with bounds for HMMs but rather an artifact of the G-GIS bound not taking into account the Markov assumption. The observation that a one-dimensional Ising model cannot have phase transitions [36] suggests that a feature constant for a tight HMM bound is only over a few frames and not the complete utterance, see also end of Section VI-C.

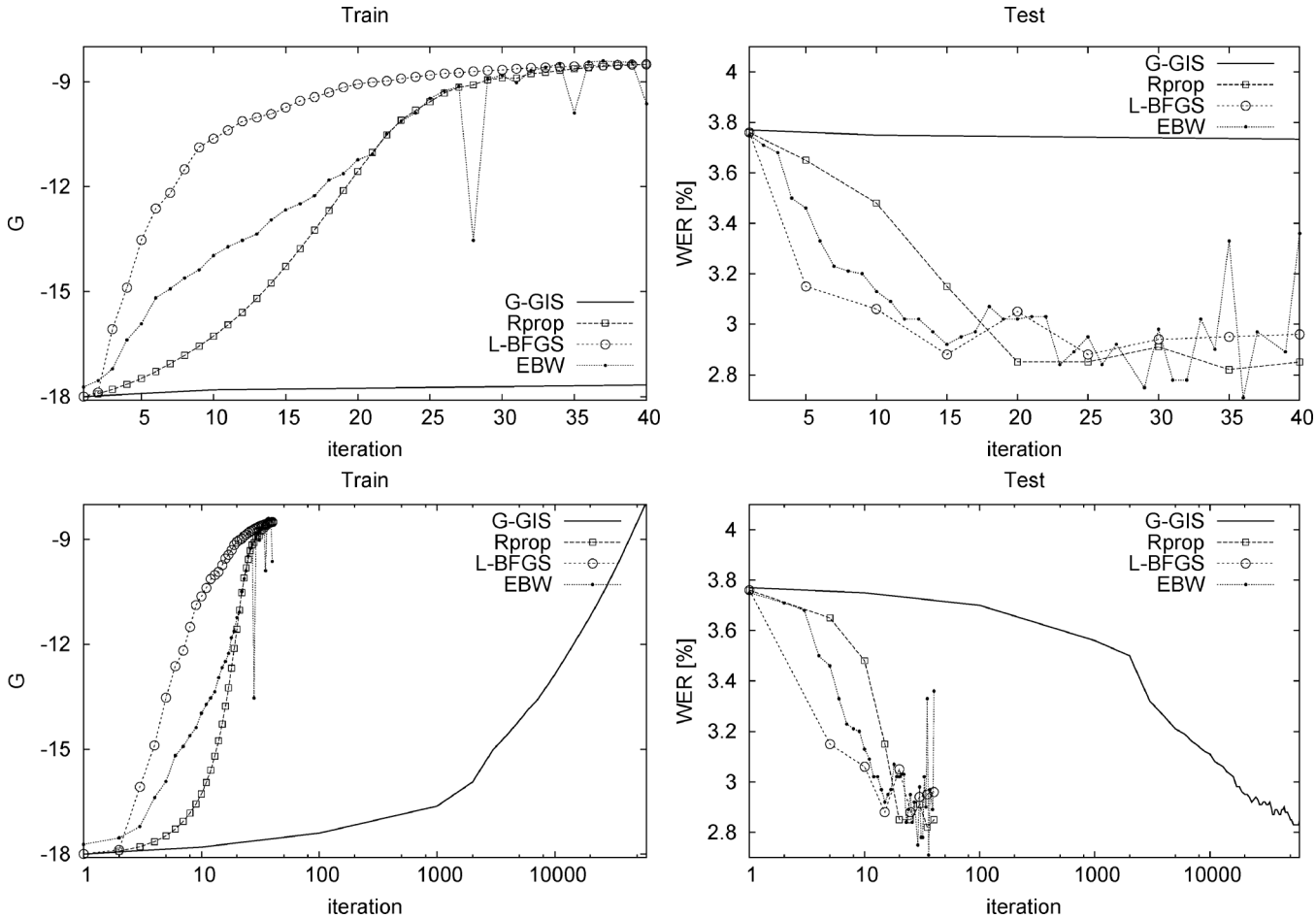

Fig. 7. Comparison of different optimization algorithms (G-GIS, Rprop, L-BFGS, EBW) for male portion of SieTill. Upper row: linear x-axis. Lower row: logarithmic x-axis. Left column: evolution of training criterion G on training corpus. Right column: evolution of word error rate (WER) on test corpus.

TABLE II
WORD ERROR RATES (WER) ON SIETILL TEST CORPUS FOR DIFFERENT OPTIMIZATION ALGORITHMS. WER IS GIVEN FOR THE BEST AND THE "LAST" ITERATION AS A ROUGH MEASURE OF VARIABILITY, FOR EXAMPLE, EBW TRAINING CRITERION VARIES A LOT TOWARDS THE END WHICH IS REFLECTED BY THE HUGE DIFFERENCE BETWEEN THE BEST AND THE LAST ITERATION

Training criterion	Optimization	WER [%]	
		best iteration	last iteration
ML	EM	3.76	
MMI	EBW	2.71	3.36
	Rprop	2.82	2.85
	L-BFGS	2.88	2.96
	G-GIS	2.83	2.85

The experimental results in this section support this scaling issue. The convergence speed decreases from single observations (Section VII-A) to string observations (Sections VII-B, and VII-C). Compared to the grapheme-to-phoneme task (Section VII-B), the digit string recognition task (Section VII-C) is expected to converge more slowly because the sequences include several hundred frames (compared to 10–20 letters), the MFCC features are basically unbounded (compared to binary features), and the training data consists of one million training frames (compared to 100,000 training letters).

Besides setting the feature constant to a reasonable empirical value, similar to [42] (Chapter 3.4.3, pp. 60) (GIS) or [7], [10], [14] (EBW), there are a several possibilities that may speed up G-GIS and will be investigated in more detail in the future. For example, replacing the unbounded MFCC features with sparse, bounded features such as posterior-based features will overcome some of the limitations. Optimal pre-processing and normalization of the features may help as well [43]. G-GIS may only be used for "fine-tuning", i.e., used to converge to a local optimum from a close initialization point. Also, a stochastic version of G-GIS with a better convergence rate may be derived. Last but not least, a bound that is aware of the Markov assumption will give considerably faster convergence.

In practice, gradient-based approaches combined with a few heuristics for setting the learning rate and tuning the free optimization parameters for a specific type of problem often gives satisfactory results at good convergence speed [7], [10], [14], [44]. That line of work is important and has its merits but should not be confused with the investigative work in this paper.

VIII. CONCLUSION

A constructive, EM-style optimization algorithm for discriminative HMMs was derived in this paper. It applies to the common discriminative training criteria such as Maximum Mutual Information (MMI) and Minimum Phone Error (MPE) for Gaussian mixture models. The theoretical properties and the feasibility of the proposed algorithm were checked for a grapheme-to-phoneme and a spoken digit string recognition

task. As predicted, the training criterion improves monotonically, at a fixed computing time per iteration, and converges to a local optimum with competitive error rates. Unlike for most other optimization algorithms, no tuning of step sizes or heuristics are required. In general, these advantages will come at the expense of the convergence speed. The reason for this is that the proposed algorithm assumes the worst case scenario to determine the global step size parameter.

APPENDIX

We sketch in this Appendix how analytical expressions for the backward probabilities in Section VI can be derived. In particular, this proves the existence of a solution. From Fig. 3, the following conditions on the backward probabilities can be formulated.

$$\begin{aligned} Z_3 &= \tilde{p}(\$|3) + Z_3\tilde{p}(3|3) + Z_1\tilde{p}(1|3) \\ Z_2 &= Z_3\tilde{p}(3|2) + Z_2\tilde{p}(2|2) \\ Z_1 &= Z_1\tilde{p}(1|1) + Z_2\tilde{p}(2|1) \\ &\dots \\ Z_0 &= Z_1\tilde{p}(1|\$) + Z_4\tilde{p}(4|\$) \end{aligned}$$

This represents a system of linear equations that can be solved by standard approaches. To explicitly solve it, the loops are resolved, i.e., the same symbols appearing on both sides of the equations are contracted.

$$\begin{aligned} Z_3 &= \frac{\tilde{p}(\$|3) + Z_1\tilde{p}(1|3)}{1-\tilde{p}(3|3)} \\ Z_2 &= \frac{\tilde{p}(3|2)}{1-\tilde{p}(2|2)}Z_3 \\ Z_1 &= \frac{\tilde{p}(2|1)}{1-\tilde{p}(1|1)}Z_2 = \frac{\tilde{p}(2|1)}{1-\tilde{p}(1|1)}\frac{\tilde{p}(3|2)}{1-\tilde{p}(2|2)}Z_3 \end{aligned}$$

Combining these results, leads to the equation for Z_3

$$Z_3 = \frac{\frac{\tilde{p}(\$|3)}{1-\tilde{p}(3|3)}}{1-\frac{\tilde{p}(2|1)}{1-\tilde{p}(1|1)}\frac{\tilde{p}(3|2)}{1-\tilde{p}(2|2)}\frac{\tilde{p}(1|3)}{1-\tilde{p}(3|3)}}.$$

Plugging these expressions into the formula in (23) gives the normalized transition probabilities.

$$\begin{aligned} p(\$|3) &= 1 - \frac{\tilde{p}(2|1)}{1-\tilde{p}(1|1)}\frac{\tilde{p}(3|2)}{1-\tilde{p}(2|2)}\frac{\tilde{p}(1|3)}{1-\tilde{p}(3|3)}(1-\tilde{p}(3|3)) \\ p(3|3) &= \tilde{p}(3|3) \\ p(1|3) &= \frac{\tilde{p}(2|1)}{1-\tilde{p}(1|1)}\frac{\tilde{p}(3|2)}{1-\tilde{p}(2|2)}\tilde{p}(1|3) \\ p(2|2) &= \tilde{p}(2|2) \\ p(3|2) &= 1-\tilde{p}(2|2) \\ p(1|1) &= \tilde{p}(1|1) \\ p(2|1) &= 1-\tilde{p}(1|1) \\ &\dots \end{aligned}$$

As a simple check, we can assume that $\tilde{p} = p$ and verify that the left-hand and right-hand sides are identical.

REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[2] Y. Normandin, "Hidden Markov models, maximum mutual information, and the speech recognition problem," Ph.D. dissertation, McGill Univ., Montreal, QC, Canada, 1991.

[3] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 25–48, 2002.

[4] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Las Vegas, NV, USA, Apr. 2008, pp. 4057–4060.

[5] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large vocabulary speech recognition using minimum classification error," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 203–223, Jan. 2007.

[6] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Proc. Interspeech*, Lisbon, Portugal, Sep. 2005, pp. 2133–2136.

[7] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge, U.K., 2004.

[8] Y. Normandin and S. Morgera, "An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Toronto, ON, Canada, May 1991, pp. 537–540.

[9] R. Schlüter, "Investigations on discriminative training criteria," Ph.D. dissertation, RWTH Aachen Univ., Aachen, Germany, Sep. 2000.

[10] W. Macherey, R. Schlüter, and H. Ney, "Discriminative training with tied covariance matrices," in *Proc. Interspeech*, Jeju Island, Korea, Oct. 2004, pp. 681–684.

[11] R. Hsiao, Y.-C. Tam, and T. Schultz, "Generalized Baum-Welch algorithm for discriminative training on large vocabulary continuous speech recognition systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3769–3772.

[12] P. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 107–113, Jan. 1991.

[13] D. Kanevsky, "Extended Baum Welch transformations for general functions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Montreal, QC, Canada, May 2004, pp. 821–824.

[14] S. Axelrod, V. Goel, R. Gopinath, P. Olsen, and K. Visweswariah, "Discriminative estimation of subspace constrained Gaussian mixture models for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 15, no. 1, pp. 172–189, Jan. 2007.

[15] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm," in *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, San Francisco, CA, USA, Mar.–Apr. 1993, pp. 586–591.

[16] G. Heigold, "A log-linear discriminative modeling framework for speech recognition," Ph.D. dissertation, RWTH Aachen Univ., Aachen, Germany, Jun. 2010.

[17] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 1999.

[18] A. D. Anastasiadis, G. D. Magoulas, and M. N. Vrahatis, "New globally convergent training scheme based on the resilient propagation algorithm," *Neurocomputing*, vol. 64, pp. 253–270, 2005.

[19] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. 39, no. B, pp. 1–38, 1977.

[20] J. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *Ann. Math. Statist.*, vol. 43, pp. 1470–1480, 1972.

[21] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 380–393, Apr. 1997.

[22] A. Gunawardana, "Maximum mutual information estimation of acoustic HMM emission densities, Center for Language and Speech Processing," Johns Hopkins Univ., Baltimore, MD, CLSP Research Note No. 40, 2001.

[23] T. Jebara, "Discriminative, generative, and imitative learning," Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, MA, USA, 2002.
[24] M. Afify, "Extended Baum-Welch reestimation of Gaussian mixture models based on reverse Jensen inequality," in *Interspeech*, Lisbon, Portugal, Sep. 2005, pp. 1113–1116.
[25] S. Riezler, "Probabilistic constraint logic programming," Ph.D. dissertation, Univ. Tübingen, Tübingen, Germany, 1998.
[26] S. Riezler, J. Kuhn, D. Prescher, and M. Johnson, "Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training," in *Proc. Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Hong Kong, Oct. 2000, pp. 480–487.
[27] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "GIS-like estimation of log-linear models with hidden variables," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Las Vegas, NV, USA, Apr. 2008, pp. 4045–4048.
[28] G. Heigold, S. Hahn, P. Lehnen, and H. Ney, "EM-style optimization of hidden conditional random fields for grapheme-to-phoneme conversion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011, pp. 4920–4923.
[29] G. Heigold, H. Ney, P. Lehnen, T. Gass, and R. Schlüter, "Equivalence of generative and log-linear models," *IEEE Trans. Speech Audio Process.*, vol. 19, no. 5, pp. 1138–1148, Jul. 2011.
[30] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[31] C. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.
[32] S. Wang, D. Schuurmans, and Y. Zhao, "The latent maximum entropy principle," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Lausanne, Switzerland, Jun.–Jul. 2002, p. 131.
[33] A. Gunawardana, M. Mahajan, A. Acero, and J. Platt, "Hidden conditional random fields for phone classification," in *Proc. Interspeech*, Lisbon, Portugal, Sep. 2005, pp. 117–120.
[34] G. Heigold, P. Dreuw, S. Hahn, R. Schlüter, and H. Ney, "Margin-based discriminative training for string recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 6, pp. 917–925, Dec. 2010.
[35] L. Armijo, "Minimization of functions having Lipschitz continuous first derivatives," *Pacific J. Math.*, vol. 16, pp. 1–3, 1966.
[36] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Z. Phys.*, vol. 31, pp. 253–258, 1925.
[37] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "Modified MMI/MPE: A direct evaluation of the margin in speech recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, Jul. 2008, pp. 384–391.
[38] G. Heigold, R. Schlüter, and H. Ney, "Modified MPE/MMI in a transducer-based framework," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3749–3752.
[39] D. Keysers, T. Deselaers, C. Gollan, and H. Ney, "Deformation models for image recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1422–1435, Aug. 2007.
[40] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Proc. 3rd Workshop Very Large Corpora*, Cambridge, MA, USA, Jun. 1995, pp. 84–94.
[41] T. Eisele, R. Haeb-Umbach, and D. Langmann, "A comparative study of linear feature transformation techniques for automatic speech recognition," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, Philadelphia, PA, USA, Oct. 1996, pp. 252–255.
[42] Y. H. Abdel-Haleem, "Conditional random fields for continuous speech recognition," Ph.D. dissertation, Faculty of Eng., Univ. of Sheffield, Sheffield, U.K., 2006.
[43] S. Wiesler and H. Ney, "A convergence analysis of log-linear training," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA, USA: MIT Press, Dec. 2011, pp. 657–665.
[44] A. Senior, G. Heigold, M. Ranzato, and K. Yang, "An empirical study of learning rates in deep neural networks for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Vancouver, BC, Canada, Apr. 2013, vol. 1, pp. 6724–6728.

Georg Heigold received the Diplom degree in physics from ETH Zurich, Switzerland, in 2000. He was a Software Engineer at De La Rue, Berne, Switzerland, from 2000 to 2003. From 2004 to 2010, he was with the Computer Science Department, RWTH Aachen University, Aachen, University. Since 2010, he has been a Research Scientist at Google, Mountain View, CA. His research interests include automatic speech recognition, discriminative training, and log-linear modeling.

Hermann Ney is a full professor of computer science at RWTH Aachen University, Germany. Before, he headed the speech recognition group at Philips Research. His main research interests lie in the area of statistical methods for pattern recognition and human language technology and their specific applications to speech recognition, machine translation and image object recognition. In particular, he has worked on dynamic programming for continuous speech recognition, language modeling and phrase-based approaches to machine translation. He has authored and co-authored more than 350 papers in journals, books, conferences and workshops. From 1997 to 2000, he was a member of the Speech Technical Committee of the IEEE Signal Processing Society. In 2006, he was the recipient of the Technical Achievement Award of the IEEE Signal Processing Society.

Ralf Schlüter studied physics at RWTH Aachen University, Germany, and Edinburgh University, UK. He received the Dipl. degree with honors in physics in 1995 and the Dr.rer.nat. degree with honors in computer science in 2000, from RWTH Aachen University. From November 1995 to April 1996 Ralf Schlüter was with the Institute for Theoretical Physics B at RWTH Aachen, where he worked on statistical physics and stochastic simulation techniques. Since May 1996 Ralf Schlüter has been with the Computer Science Department at RWTH Aachen University, where he currently is senior researcher and leads the automatic speech recognition group at the Human Language Technology and Pattern Recognition institute. His research interests cover speech recognition, discriminative training, decision theory, stochastic modeling, and signal analysis.