Optimization Algorithms and Applications for Speech and Language Processing

Stephen J. Wright, Dimitri Kanevsky, *Senior Member, IEEE*, Li Deng, *Fellow, IEEE*, Xiaodong He, *Senior Member, IEEE*, Georg Heigold, *Member, IEEE*, and Haizhou Li, *Senior Member, IEEE*

***Abstract*—Optimization techniques have been used for many years in the formulation and solution of computational problems arising in speech and language processing. Such techniques are found in the Baum-Welch, extended Baum-Welch (EBW), Rprop, and GIS algorithms, for example. Additionally, the use of regularization terms has been seen in other applications of sparse optimization. This paper outlines a range of problems in which optimization formulations and algorithms play a role, giving some additional details on certain application problems in machine translation, speaker/language recognition, and automatic speech recognition. Several approaches developed in the speech and language processing communities are described in a way that makes them more recognizable as optimization procedures. Our survey is not exhaustive and is complemented by other papers in this volume.**

***Index Terms*— Natural language processing, optimization methods, speech processing.**

I. Introduction

ALGORITHMS for processing speech and language data as a form of machine learning have made extensive use of optimization techniques over many years [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. In common with other applications of machine learning, mathematical formulations of the learning problem in speech and language processing depend on models constructed from statistical principles, loss functions that quantify misfit between observed data and predictions, parameters for the models, "priors" for these parameters, regularization functions that quantify the complexity of the parameters (and deviation from priors), and large sets of data. The data in these speech and language processing applications often consist of labeled instances of phonemes, words, phrases, or sentences of text; or samples of speech from a particular speaker. The overall objective function for a data set is thus "partially separable," consisting of a sum of distinct terms, all depending on the parameters of the model, but each pertaining to just a single item of data, which may be a full paragraph, sentence, word, segment of speech, or frame of speech. Specifically, our objective function often has the general form

$$\max_{\Lambda \in \mathcal{V}} O(\Lambda) := \sum_{t=1}^{T} O_t(\Lambda),$$

where Λ denotes the parameters, $O(\cdot)$ denotes the overall objective, O_t denotes the partial objective corresponding to a single item of data, and $\mathcal{V}$ denotes the constraint set. This separability property places speech and language processing formulations into the same framework as many other machine learning applications.

Construction of models and loss functions for these applications often make use of classification functions (such as hinge-loss or logistic functions [15]), support-vector-machine formulations [16], and Bayesian statistics [17]–[19]. In some applications, the optimization formulations that result from these techniques are convex, a feature that allows for strong convergence guarantees to be made for the algorithms applied to these formulations in some relatively simple speech and language processing problems [20], [21]. However, many optimization problems in speech and language processing are *nonconvex* in nature, due to the complexity of the models (e.g., having latent variables) in use and to the fact that loss functions are designed to measure actual recognition errors [10], [22].

The purpose of this paper is to review a number of key application areas from the speech and language processing literature, focusing in particular on the way in which the important data processing problems in these areas are formulated and solved. We mean to explain the formulations and algorithms in a common optimization framework, identifying clearly the variables, objectives, and constraints. We also explore the relationship between the algorithms derived in the application communities and the model-based framework that is the basis of many important optimization algorithms. By placing past work in these areas more firmly in the context of optimization, and thus more easily comprehensible to those outside the communities in which this work was originally performed, we hope to make it easier to identify possible enhancements to the algorithms, and thus to stimulate new research at the intersection of optimization and speech/language processing.

A. Outline of the Paper

In Section II we provide the background material on optimization, via a general statement of the convex optimization problem—maximization of a concave function over a convex set—with smooth objective functions. We derive a model-based

Manuscript received December 08, 2012; revised June 18, 2013; accepted September 15, 2013. Date of current version October 16, 2013. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Mari Ostendorf.

S. J. Wright is with the Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI 53706 USA (e-mail: swright@cs.wisc.edu).

D. Kanevsky is with IBM Research, Yorktown Heights, NY, 10598 USA (e-mail: krmrmn@yahoo.com).

L. Deng and X. He are with Microsoft Research, Redmond, WA 98052-6399 USA (e-mail: deng@microsoft.com; xiaohe@microsoft.com).

G. Heigold is with Google Research, Mountain View, CA 94043 USA (e-mail: heigold@google.com).

Haizhou Li is with the Institute for Infocomm Research, Singapore, 138632 (e-mail: hli@i2r.a-star.edu.sg).

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.2013.2283777

framework for iterative methods that make use of gradient information. Most algorithms discussed later in the paper make use of such a framework.

The material following this background section can be divided into two parts. First, we discuss several key problems in speech and language processing, focusing on their formulation as optimization problems. Section III discusses the general problem of constructing utility-based objectives, while Section IV discusses machine translation and Section V discusses speaker and language recognition, two important areas of speech and language processing where optimization methods have played important roles. Second, we discuss key algorithmic approaches that have been developed in the applications communities, and relate them both to the applications described earlier and to the model-based optimization framework described in Section II. The extended Baum-Welch and line-search $\mathcal{A}$-function approaches are derived in Section VI, while Section VII discusses lower-bounding methods (which includes the class of methods known as GIS), while Section VIII discusses the Rprop method. We cover these selected methods in detail because they either have found successful uses in speech and language processing applications or are equipped with strong theoretical properties which serve well for unified treatment of a range of optimization methods popular in the applications.

II. Optimization Background

Here we outline a general formulation for the smooth constrained optimization problem, together with a model-based algorithmic framework for solving it, and discuss alternative approaches based on line search. This material provides a foundation for later sections, and we refer back to it frequently. The formulation is as follows:

$$\max_{\Lambda\in\mathcal{V}} O(\Lambda), \tag{1}$$

where

- Λ is the variable or parameter, which is generally a collection of real numbers, real vectors, and symmetric positive semidefinite matrices;
- $O(\cdot)$ is a continuous function mapping $\mathcal{U}$ to the real numbers $\mathbb{R}$, where $\mathcal{U}$ is an open set of the variables of the form Λ;
- $\mathcal{V}$ is a closed constraint set for the parameters Λ, with $\mathcal{V}\subset\mathcal{U}$.

We assume that the sets $\mathcal{U}$ and $\mathcal{V}$ are convex, as is true in the formulations discussed in this paper. The constraint sets $\mathcal{V}$ can incorporate bounds on the real and vector variables, and positive semidefiniteness constraints on the symmetric matrices in the formulation, or bounds on their eigenvalues. Another commonly occurring constraint set is the unit simplex:

$$\mathcal{V} := \left\{\Lambda\in\mathbb{R}^d \middle| \Lambda\geq 0, \sum_{i=1}^{d}\Lambda_i = 1\right\}, \tag{2}$$

which arises when the parameters represent probabilities in a distribution over a discrete number n of possible values. When $\mathcal{V}$ is the full space (that is, when we do not impose constraints on Λ), we say that the problem (1) is *unconstrained*.

We do not consider discrete-valued variables in this paper. Such variables appear in formulations of some important speech processing problems, but there are also many interesting and fundamental formulations that require only continuous variables, so we restrict ourselves to this more tractable class of applications.

Iterative algorithms for (1) frequently take the approach of forming a simplified, approximation to the function O around the current iterate Λ. Denoting this model by $\Phi(\tilde{\Lambda};\Lambda)$, we can then obtain the new iterate Λ^+ by solving

$$\hat{\Lambda} := \arg\max_{\tilde{\Lambda}\in\mathcal{V}} \Phi(\tilde{\Lambda};\Lambda), \tag{3}$$

and then searching along the line from Λ to $\hat{\Lambda}$ to find a new point Λ^+ with an improved value of O. We can thus generate a sequence of iterates $\{\Lambda^k\}_{k=0,1,2\ldots}$ by choosing Λ^0 and then solving (3), with $\Lambda := \Lambda^k$. Specifically, we set

$$\hat{\Lambda}^k := \arg\max_{\tilde{\Lambda}\in\mathcal{V}} \Phi(\tilde{\Lambda};\Lambda^k), \quad \Lambda^{k+1} := \Lambda^k + \alpha_k(\hat{\Lambda}^k - \Lambda^k), \tag{4}$$

where α_k is a line search parameter, usually restricted to the interval (0,1]. The line search may not be necessary in some circumstances, for example, when $\Phi(\cdot;\Lambda)$ is an underestimate of the function O. In such situations we can set $\alpha_k \equiv 1$ and $\Lambda_{k+1} = \hat{\Lambda}_{k+1}$.

Usually, we require the approximate model function Φ to match the function value and gradient of the true objective O at the current iterate Λ, that is,

$$\Phi(\Lambda;\Lambda) = O(\Lambda), \quad \nabla_{\tilde{\Lambda}}\Phi(\tilde{\Lambda};\Lambda)|_{\tilde{\Lambda}=\Lambda} = \nabla O(\Lambda). \tag{5}$$

For practicality, we require the subproblem (3) to be significantly easier to solve than the original problem (1)—otherwise there would be no point in using the approximation Φ in place of the true objective O.

Specific choices of the model function Φ include the concave quadratic

$$\begin{aligned}\Phi(\tilde{\Lambda};\Lambda) :&= O(\Lambda) + \nabla O(\Lambda)^T(\tilde{\Lambda}-\Lambda)\\ &\quad - \frac{1}{2\gamma}\|\tilde{\Lambda}-\Lambda\|_2^2,\end{aligned} \tag{6}$$

where $\gamma > 0$ is some constant; and

$$\begin{aligned}\Phi(\tilde{\Lambda};\Lambda) :&= O(\Lambda) + \nabla O(\Lambda)\\ &\quad + \frac{1}{2}(\tilde{\Lambda}-\Lambda)^T\nabla^2 O(\Lambda)(\tilde{\Lambda}-\Lambda),\end{aligned} \tag{7}$$

in which Φ is a second-order Taylor-series approximation to O around the current point Λ. When used in conjunction with (4), the model (6) leads to the steepest-descent algorithm (with line search), while (7) leads to the line-search Newton method. Another variant is to replace the Hessian matrix $\nabla^2 O(\Lambda)$ in (7) by a symmetric negative-definite approximation H_k, which is constructed using information gathered at previous iterates. This approach yields the class of *quasi-Newton methods*, which includes L-BFGS. Reference [23] describes algorithms for continuous optimization, while [24] focuses on convex optimization.

The model-based approach is by no means the only possible way to construct useful methods for optimization of nonlinear functions. It is often more useful to think of the search direction $\hat{\Lambda}^k - \Lambda^k$ in (4) being defined directly, rather than being derived from the solution of a model problem. A class of methods that could be termed "momentum methods" defines the search direction to be a linear combination of the previous search step and the latest gradient, that is,

$$\hat{\Lambda}^k - \Lambda^k := \delta_k \nabla O(\Lambda^k) + \epsilon_k (\hat{\Lambda}^{k-1} - \Lambda^{k-1}), \tag{8}$$

for some scalars δ_k and ϵ_k. In unconstrained optimization, the *conjugate gradient* [23, Chapter 5] and *heavy-ball* [25] methods are algorithms of this type. Such methods require only function and gradient information about the function $O(\cdot)$, but have superior convergence rates to the steepest descent method. Another approach is to use an approximation Ξ^k to the steepest descent direction $\nabla O(\Lambda)$, and define

$$\hat{\Lambda}^k := \Lambda^k + \Xi^k. \tag{9}$$

In unconstrained optimization, this choice of $\hat{\Lambda}^k$ will yield a direction of ascent for O provided that

$$\nabla O(\Lambda^k)^T \Xi^k > 0. \tag{10}$$

In other words, for all sufficiently small choices of α_k in (4), we will identify a new iterate Λ^{k+1} such that $O(\Lambda^{k+1}) > O(\Lambda^k)$.

III. Utility-Driven Objectives

Objective functions in our application space share many common features. Here we discuss a general framework that encompasses a variety of such applications in speech and language processing, giving references to earlier survey works that contain additional details.

In [10], a general parameter learning criterion was presented for speech recognition which unifies the commonly used discriminative training criteria, including maximum mutual information (MMI), minimum classification error (MCE), and minimum phone/word error (MPE/MWE). Using R to denote the number of sentences in a training set, and X_i and F_i, $i = 1, 2, \ldots, R$ to denote the speech-feature sequence and the recognition-symbol sequence of ith utterance, respectively, we consider the following unified objective for discriminative training:

$$O(\Lambda) = \sum_{F_1,\ldots,F_R} p(F_1, \ldots, F_R | X_1, \ldots, X_R, \Lambda) \times C_{DT}(F_1, F_2, \ldots, F_R). \tag{11}$$

Here, Λ is the set of parameters in the model while $p(F_1, \ldots, F_R | X_1, \ldots, X_R, \Lambda)$ denotes the a posteriori probabilities for recognition symbols F_i, given utterances X_i and model parameters Λ. The function $C_{DT}(F_1, F_2, \ldots, F_R)$ is a classification quality measure that is independent of Λ. By taking different forms of C_{DT}, we recover a number of familiar discriminative training criteria:

- maximum mutual information (MMI): $\prod_r \delta(F_r, F_r^*)$ (a product of 0-1 loss functions);
- minimum classification error (MCE): $\sum_r \delta(F_r, F_r^*)$ (a sum of 0-1 loss fucntions);
- minimum phone/word error (MPE/MWE): $\sum_r A(F_r, F_r^*)$, where $A(F_r, F_r^*)$ represents the raw phone/word accuracy count of F_r given the transcription reference F_r^*.

More detection-based alternatives to error rate are discussed in Section V.

From this point of view, the MMI, MCE, and MPE/MWE criteria can all be viewed as model-based expectations of a classification quality measure defined on the entire training corpus. For MMI, this measure is a 0-1 loss of the structure of the training corpus. For MCE, it is a sum of 0-1 losses, defined at each sentence, while MPE is defined at the level of phones. These have all been well studied by speech recognition researchers [26], [27], and the same general training criterion has been extended to large scale discriminative training of translation models for statistical machine translation, to be discussed in Section IV. It can be extended further to a set of speech-centric information processing tasks; see [22].

In order to avoid overfitting and to improve on generalization, a prior on the model parameters can be added to the training criterion defined in Equation (11) (cf. I-smoothing for Gaussian models [5] or ℓ_1- and ℓ_2-regularization for log-linear models [28]). Further, we can extend the training criterion defined in (11) by replacing posterior $p(F_1, \ldots, F_R | X_1, \ldots, X_R, \Lambda)$ with the equation shown at the bottom of the page, where $C_{MT}(F_1, \ldots, F_R)$ is another (possibly scaled) cost function. This effectively adds a boosting [29] or margin [30] term, which allows for margin-based training within the framework of utility-driven objectives.

IV. Machine Translation

A. Background

Machine Translation (MT) is the process of converting text in one language (source) to another language (target). The history of machine translation can be traced back to the 1950s [31]. In the early 1990s, Brown *et al.* [32] of IBM conducted the pioneering work in statistical modeling approaches, establishing a range of IBM models, known prosaically as "Model 1" through "Model 5." Since that time, important progress has been made in a wide span of MT component technologies, including word alignment, phrase-based MT methods [33], hierarchical phrase-based methods, syntax-based methods, discriminative training methods [34], [35], model adaptation methods, and system combination methods. Modern statistical MT commonly makes use

$$\frac{p(F_1, \ldots, F_R, X_1, \ldots, X_R, \Lambda) \exp(C_{MT}(F_1, \ldots, F_R))}{\sum_{\tilde{F}_1,\ldots,\tilde{F}_R} p(\tilde{F}_1, \ldots, \tilde{F}_R, X_1, \ldots, X_R, \Lambda) \exp(C_{MT}(\tilde{F}_1, \ldots, \tilde{F}_R))}$$

of a log-linear model. Using the standard MT terminology, the optimal translation $\hat{E}$ given the input sentence F is obtained via the decoding process according to

$$\hat{E} = \arg\max_{E} p(E|F, \Lambda),$$

where the posterior probability above of the output sentence E given the text F is computed through a log-linear model:

$$P(E|F, \Lambda) = \frac{1}{Z} \exp\left(\sum_i (w_i \log h_i(E, F)) \right)$$

where $Z = \sum_E \exp(\sum_i (w_i \log h_i(E, F))$ is the normalization denominator which ensures that the probabilities sum to one. Feature functions $h_i(E, F)$ (also known as component models) are constructed from E and F, and the w_i are known as feature weights.

In a conventional MT system, there are a handful of features, including language and translation models. The parameter set Λ of an MT system includes the feature weights and all the parameters in the component models. In training, the choice of features is usually made by referring to a small hold-out development set. In this paper, we focus on optimization of the parameters of the component models. The remainder of this section focuses on phrase and lexicon translation models.

Phrase-based SMT consists of three steps: segmentation of source sentences into a sequence of phrases [36]; translation of each source phrase to a target phrase; and reordering of target phrases into target sentences [33], [37]. Elements of a phrase-based system include the language model, reordering model, word and phrase counts, and phrase and lexicon translation models. The language model here is the same as that used in speech recognition, while the reordering model described in [33] is a simple penalty proportional to the jump distance. After segmentation of the input sentence into K phrases, the source-to-target forward phrase translation feature is scored by assuming independence among individual phrases in the input sentence. The target-to-source (backward) phrase translation model is defined similarly.

The lexicon translation model defines the translation relationship between source and target at the word level. Assuming that the input sentence is segmented into K phrases, the source-to-target forward phrase translation component model can be written as follows:

$$h_{\text{F2E-ph}}(E, F) = \prod_k p(\tilde{e}_k | \tilde{f}_k),$$

where $\tilde{e}_k$ and $\tilde{f}_k$ are the kth phrase in E and F, respectively. Conventionally, phrase translation probabilities are computed as relative frequencies of phrases over the training dataset, that is,

$$p(\tilde{e}|\tilde{f}) = \frac{C(\tilde{e}, \tilde{f})}{C(\tilde{f})}$$

where $C(\tilde{e}, \tilde{f})$ is the joint count of $\tilde{e}$ and $\tilde{f}$, while $C(\tilde{f})$ is the marginal counts of $\tilde{f}$. The target-to-source (backward) phrase translation model is defined similarly.

One popular lexicon translation model is based on the IBM Model 1 [32], in which the source-to-target forward lexicon translation feature is hand-engineered using independence assumption and given by

$$h_{\text{F2E-wd}}(E, F) = \prod_k \prod_m \sum_r p(e_{k,m} | f_{k,r}),$$

where $e_{k,m}$ is the mth word of the kth target phrase $\tilde{e}_k$, $f_{k,r}$ is the rth word in the kth source phrase $\tilde{f}_k$, and $p(e_{k,m}|f_{k,r})$ is the probability of translating word $f_{k,r}$ to word $e_{k,m}$. The target-to-source (backward) lexicon translation model is defined similarly.

B. Using EBW to Optimize Translation Models

The Extended Baum-Welch (EBW) algorithm for optimization, which will be covered in detail in Section VI, has been applied successfully to training translation models for statistical MT [38]. A utility function is defined as the expected BLEU score over the entire training set, where BLEU (BiLingual Evaluation Understudy) [39] is the most popular automatic, inexpensive metric for evaluating the quality of text that has been machine-translated from one natural language to another.

We write the BLEU-score-based utility function for machine translation over the entire training set as follows:

$$U(\Lambda) = \sum_{E_1, E_2, \ldots, E_R} P(E_1, E_2, \ldots, E_R | F_1, F_2, \ldots, F_R, \Lambda) \times \left(\sum_{r=1}^{R} \text{BLEU}(E_r, E_r^*) \right),$$

where R is the number of sentences in the training set, E_r^* is the reference translation of the rth source sentence, F_r and $E_r \in \text{Hyp}(F_r)$ denotes the list of translation hypotheses for F_r. We further denote by $\Lambda = \{p_{ij}\}$ the set of model parameters to be optimized, including all the phrase translation probabilities and the lexical translation probabilities in the translation models. All translation models are discrete probability distributions, and are thus subject to simplex-type constraints of the form $\sum_j p_{ij} = 1$ for all i, and $p_{ij} \geq 0$, for all i, j (see (2)). Note that the function $U(\Lambda)$ has the form of our general objective $O(\Lambda)$ for discriminative training defined in (11), where the classification quality measured C_{DT} is defined to be the BLEU criterion.

To prevent overfitting to the training set, a regularization scheme based on KL-divergence is applied, which is computed as the sum of KL divergence over the entire parameter space:

$$\text{KL}(\Lambda^0 \| \Lambda) = \sum_i \sum_j p_{ij}^0 \log \frac{(p_{ij}^0)}{p_{ij}}$$

where Λ^0 denotes the set of prior model parameters. The objective function to be optimized thus becomes

$$O(\Lambda) = \log U(\Lambda) - \tau \text{KL}(\Lambda^0 \| \Lambda)$$

Optimization of $O(\Lambda)$ can be achieved by the classical EBW algorithm as detailed in [22], [38], and in Section VI below.

V. Speaker and Language Recognition

In this section, we present another significant application area in speech and language processing where optimization plays important roles. The goal in speaker recognition is to establish or verify the identity of a speaker using a sample of the speaker's voice [40], [41], [16], [42], [43], [44]. Similarly, spoken language recognition (or, simply, language recognition) aims to identify or confirm the language, dialect, or accent that is spoken in a speech sample [45], [46], [47]. Both are typical pattern recognition tasks that also make use of prior knowledge.

Speaker or language recognition is commonly formulated as a detection problem rather than an identification problem. We take language recognition as an example. If all possible languages are known to the system (a situation known as a "closed set"), language recognition could be viewed as a multi-way classification task. However, the recognition task often involves unknown languages—the so-called "open-set" scenario. Thus, a hypothesis test is required to accept or reject an identification result—a process referred to as language verification or detection in the literature [45]. In contrast to identification, one essential element of a detection task is a proper calibration of the output scores from the classifier. Score calibration is related to the ability to properly set a threshold for the hypothesis test so as to minimize the detection cost. This can be achieved with a linear transformation function, the parameters of which are obtained via optimization of logistic loss [48], [49], [14].

Speaker and language detection systems have a common foundation in statistics and optimization, so they share common tools, utilities [50], and performance evaluation metrics. In this section, we highlight the performance metrics typically used for detection tasks and the related optimization techniques. We explain the issues in formulating the so-called detection cost as the objective function, the elements used in the construction of this function, and the use of optimization in model training, score calibration and fusion. Specifically, we describe the formulation of two optimization problems of the form (1). Sections V-A and V-B culminate in an objective $O(\Lambda)$ derived from (14), (15), and (16), whose gradient is given in (17). Section V-C describes a score fusion and calibration model that results in the objective (19).

A. Detection Cost Function

In a detection task, system performance is evaluated by presenting the system with a set of trials, each consisting of a test sample and a hypothesized identity. The system has to decide, for each trial, whether to accept or reject the hypothesized identity for the given test sample, by comparing the confidence score with a given confidence threshold θ. The confidence score of a test sample is defined in terms of the discrepancy between the target model and a reference model, as shown in Fig. 1. There are four possible outcomes for this comparison:

- true positives (TPs) denoting true samples correctly labeled as positives;
- false positives (FPs) denoting negative samples incorrectly labeled as positives;
- true negatives (TNs) denoting samples correctly labeled as negatives;
- false negatives (FNs) denoting positive samples incorrectly labeled as negatives.

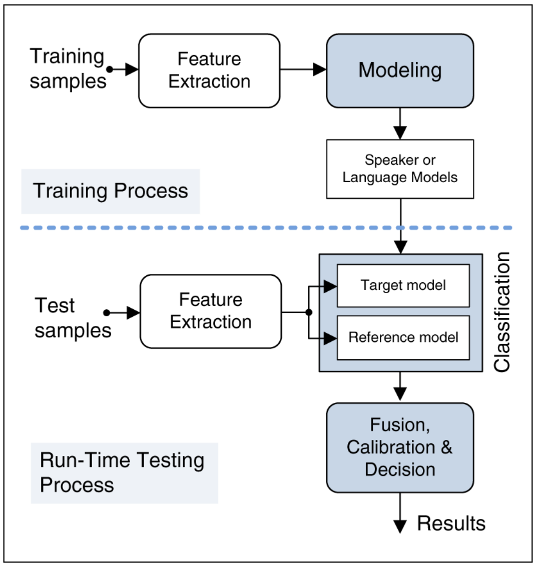

Fig. 1. Block diagram showing a general approach to speaker or language recognition. Two processes are involved. The upper panel shows the training process in which we model the extracted features to represent the intended patterns. In the lower panel, we compare the extracted features with the hypothesized patterns to make a classification decision at run time.

FNs and FBs represent two type of errors with error rates, *miss probability* (P_{miss}) and *false-alarm probability* (P_{fa}), defined as follows:

$$P_{\text{miss}} = \frac{\text{FN}}{\text{TP}+\text{FN}}, \quad P_{\text{fa}} = \frac{\text{FP}}{\text{FP}+\text{TN}}. \tag{12}$$

For P_{miss}, the denominator is the total number of positive (or target) trials, while the denominator of P_{fa} is the total number of negative (non-target) trials.

Depending on the application, the cost of miss detection (C_{miss}) and the cost of false alarm (C_{fa}) may vary significantly. For instance, one might aim for a higher value of C_{miss} but lower value of C_{fa}, to impose a higher security level on an access control system. Taking these into consideration, we define a detection cost function (DCF) [51] as follows:

$$C_{\text{DET}} = C_{\text{miss}} P_{\text{tar}} P_{\text{miss}} + C_{\text{fa}}(1-P_{\text{tar}})P_{\text{fa}} \tag{13}$$

where P_{tar} is the a priori probability that a trial is a target trial. The prior P_{tar} does not correspond to the actual percentage of target trial in the dataset; it is generally set to a value that is envisaged in the applications. Notice that the DCF in (13) could be evaluated by pooling the scores from sufficient number of trials. Alternatively, one might use the *average DCF* C_{avg} [52]–[54], in which replace P_{miss} and P_{fa} by sample averages and define

$$\begin{aligned} C_{\text{avg}} :&= C_{\text{miss}} P_{\text{tar}} \underbrace{\frac{1}{M}\sum_{i=1}^{M} P_{\text{miss}}(i)}_{P_{\text{miss}}} \\ &\quad + C_{\text{fa}}(1-P_{\text{tar}}) \underbrace{\frac{1}{M}\sum_{i=1}^{M}\left[\frac{1}{N-1}\sum_{j\neq i} P_{\text{fa}}(i,j)\right]}_{P_{\text{fa}}}, \end{aligned} \tag{14}$$

where N is the number of classes, which may be larger than the number M of target classes in the open-set scenario. In (14), the miss probabilities $P_{\text{miss}}(i)$ are computed separately for each target class, and for each i the false alarm probabilities $P_{\text{fa}}(i,j)$ are computed for each target/non-target pair (i,j). By using the averaged detection cost C_{avg} in place of the pooled detection cost C_{DET} as a performance metric, the number of target trials does not influence the resulting detection cost.

B. Minimizing DCF of a Base System

Optimizing the detection costs is not straightforward. We present more details about formulation of the functions $P_{\text{miss}}(i)$ and $P_{\text{fa}}(i,j)$ that appear in (14). Consider the case of the language recognition, where the indexes i and j indicate the target and non-target languages, respectively. One approach [55] is to use parametrized continuous and differentiable functions, as follows

$$\begin{aligned} P_{\text{miss}}(i) &= \frac{1}{|\Omega_i|}\sum_{\mathbf{s}\in\Omega_i} l_i(\mathbf{s}|\Lambda), \\ P_{\text{fa}}(i,j) &= \frac{1}{|\Omega_j|}\sum_{\mathbf{s}\in\Omega_j} [1 - l_i(\mathbf{s}|\Lambda)], \end{aligned} \tag{15}$$

where Ω_i is the set of training data belonging to the ith language. Assume that a base system is built upon L language detectors. The loss function $l_i(\mathbf{s}|\Lambda)$ is defined for the score vector $\mathbf{s}\in\mathbb{R}^L$ of a test segment given a set of parameters Λ. An element in the vector $\mathbf{s}$ is a score from one of the L language detectors, while the set of parameters Λ describes the target and reference language models. In particular, the loss function l_i is defined as

$$l_i(\mathbf{s}|\Lambda) := [1 + \exp(-\gamma d_i(\mathbf{s}|\Lambda) + \beta)]^{-1} \tag{16}$$

where γ is a positive constant that controls the learning rate, and β is a constant measuring the offset of $d_i(\mathbf{s}|\Lambda)$ from 0. The function $d_i(\mathbf{s}|\Lambda)$ is a misclassification measure [56], defined so that $d_i(\mathbf{s}|\Lambda) > 0$ implies a misclassification and $d_i(\mathbf{s}|\Lambda) \leq 0$ means a correct decision. In [55], the authors consider the case in which Λ is the set of parameters of two Gaussian mixture models (GMMs), Λ_i for the target language i, and $\Lambda_{\bar{i}}$ for the reference model of target language i, both trained on score vectors $\mathbf{s}$. With these definitions, the misclassification measure is given by $d_i(\mathbf{s}|\Lambda) = -\log p(\mathbf{s}|\Lambda_i) + \log p(\mathbf{s}|\Lambda_{\bar{i}})$.

By substituting (15) and (16) into (14), the DCF function (14) becomes a continuous objective function $O(\Lambda)$, depending on the set Ω of training data from all languages. The derivative of this function, which is utilized in model-based optimization approaches and in the generalized probabilistic descent (GPD) algorithm described in [56], [55], is defined as follows:

$$\begin{aligned} \nabla O(\Lambda) = \frac{1}{M}\Bigg[& C_{\text{miss}}P_{\text{tar}}\sum_{\mathbf{s}\in\Omega_i}\frac{1}{|\Omega_i|}\nabla_\Lambda l_i(\mathbf{s}|\Lambda) \\ & - \frac{C_{\text{fa}}(1-P_{\text{tar}})}{(N-1)}\sum_{j\neq i}\sum_{\mathbf{s}\in\Omega_j}\frac{1}{|\Omega_j|}\nabla_\Lambda l_i(\mathbf{s}|\Lambda)\Bigg]. \end{aligned} \tag{17}$$

C. Optimization for Score Fusion and Calibration

A state-of-the-art speaker verification system is typically a fusion of multiple base systems that are developed using different acoustic features and classifiers to serve as a mixture of experts. Aside from the base systems optimization discussed above, a related issue is score fusion, calibration and the setting of detection threshold for the fused score. An approach now accepted widely is to postprocess the scores via the so-called application-independent calibration, then set the threshold in a straightforward way based on the application parameters $\{C_{\text{miss}}, C_{\text{fa}}, P_{\text{tar}}\}$, giving minimum risk decisions. In the following, we show how the score fusion and calibration can be accomplished jointly [48], [49], [14] for the case of speaker recognition.

Assume that we have K base systems, each producing a score for a given test segment. A linear fusion of these scores, using weight vector $\mathbf{w}\in\mathbb{R}^K$ and bias β, is $\mathbf{w}^T\mathbf{s}+\beta$, where $\mathbf{s}\in\mathbb{R}^K$ is now a vector holding the scores from the K base systems. The parameters $\Lambda = \{\mathbf{w},\beta\}$ are trained by embedding the fused score in a sigmoid function, leading to the following logistic regression model:

$$P(y|\mathbf{s}) = [1 + \exp(-y(\mathbf{w}^T\mathbf{s}+\beta))]^{-1}, \tag{18}$$

where $y\in\{-1,1\}$ indicates whether the test segment originates from a target speaker ($y=1$) or from a non-target ($y=-1$). Using a development dataset, we find parameters $\Lambda = (\mathbf{w},\beta)$ that maximize the log-posterior using the following as the objective function:

$$\begin{aligned} O(\Lambda) = O(\mathbf{w},\beta) := & -\frac{1}{|\Omega_{\text{tar}}|}\sum_{\mathbf{s}\in\Omega_{\text{tar}}}\log[1+\exp(-(\mathbf{w}^T\mathbf{s}+\beta))] \\ & -\frac{1}{|\Omega_{\text{non}}|}\sum_{\mathbf{s}\in\Omega_{\text{non}}}\log[1+\exp(\mathbf{w}^T\mathbf{s}+\beta)]. \end{aligned} \tag{19}$$

Here, Ω_{tar} and Ω_{non} denote the set of score vectors belonging to the of target and non-target trials, respectively. There is no closed-form solution to this formulation, in general. For minimizing this objective, a conjugate gradient descent optimization technique was found to be the most efficient amongst several other gradient-based numerical optimization techniques [15], [57], [58].

In the above, score fusion is achieved by optimizing (19), where the scores from K base systems are combined linearly using the optimized weights $\mathbf{w}$ and bias β. It was further reported in [57], [58] that, another benefit of minimizing the cost function (19) is that the fused score could be interpreted as a well-behaving (i.e., properly calibrated) log-likelihood ratio. With this, a threshold could then be set directly on the fused score depending on the desirable costs (C_{miss}, C_{fa}) and priors (P_{tar}), that are specific to the applications, as follows:

$$\theta = \log\left(\frac{C_{\text{fa}}}{C_{\text{miss}}}\times\frac{1-P_{\text{tar}}}{P_{\text{tar}}}\right). \tag{20}$$

Notice that θ is derived according to the Bayes decision rule [3]. When $K = 1$, $\mathbf{w}^T\mathbf{s} + \beta$ reduces to a score calibration device. Similar techniques have been applied for language detection [59], [45].

VI. Extended Baum-Welch and Model-Based Optimization

In Section IV, we introduced the Extended Baum-Welch (EBW) technique, popular in the speech and language processing community but little known in optimization, in the context of machine translation applications. In this section, we describe EBW and show how it can be viewed as a model-based algorithm for optimization.

Historically, EBW was introduced for estimating the discrete probability parameters of multinomial distribution functions of HMM speech recognition problems under the Maximum Mutual Information (MMI) discriminative objective function [60]. Later, in [61] and [62], EBW was extended to estimating the parameters of Gaussian Mixture Models (GMMs) of HMMs under the MMI discriminative function for speech recognition problems. EBW's popularity, like that of its namesake Baum-Welch algorithm, is due to the simplicity of its recursion formula for updating the model parameters in "Expectation-Maximization" (EM) fashion and to its impressive numerical performance.

In [60] it was shown that the value of the objective function improves at every iteration of EBW. This property was first noted for arbitrary functions of GMMs in [63] and in [64], [65] for rational functions. In [66], the EBW recursion formula for GMMs was recast in a new form using the notion of "associated functions," along with an optimization process based on these functions. In [67], the EBW technique was generalized beyond associated functions to a class called "$\mathcal{A}$-functions." The relationship between these classes was explored further in [66], allowing the EBW approach to be extended beyond GMMs to a variety of probability density and distribution functions such as exponential, Poisson, and gamma functions. In [67], this extension of EBW was named the *Line Search $\mathcal{A}$-functions* (LSAF) optimization technique.

In this section, we show that the LSAF technique can be placed firmly in the model-based optimization framework of Section II, that is, algorithms that obtain each new iterate by solving a subproblem in which the original objective is replaced by a model function which is simpler than the true objective O, but locally similar to it. We then show how, for the particular case of multivariate Gaussian densities, the EBW update formula is derived from the LSAF approach, and thus that EBW can be viewed as a model-based optimization approach with line search.

A. The Line-Search $\mathcal{A}$-Function Technique

Given the real-valued function O on an open set $\mathcal{U}$, we say that A_O is an $\mathcal{A}$-function for O if the following properties hold:

1. $A_O(\tilde{\Lambda}, \Lambda)$ is a strictly convex or strictly concave function of $\tilde{\Lambda} \in \mathcal{U}$, for all $\Lambda \in \mathcal{U}$;
2. $\nabla_{\tilde{\Lambda}} A_O(\tilde{\Lambda}; \Lambda)|_{\tilde{\Lambda}=\Lambda} = \nabla O(\Lambda)$. That is, the first derivatives of $A_O(\cdot, \Lambda)$ and O agree at all $\Lambda \in \mathcal{U}$.

Let us assume for simplicity that A_O is strictly concave. (The discussion can be modified easily, with some changes of sign, when A_O is strictly convex.) In the line-search A-function (LSAF) approach, consistently with (4), we move from the current iterate Λ to a new iterate Λ^+ by first finding the maximum of $A_O(\cdot, \Lambda)$:

$$\hat{\Lambda} = \arg\max_{\tilde{\Lambda} \in \mathcal{U}} A_O(\tilde{\Lambda}, \Lambda).$$

We then perform a line search from the current iterate Λ toward $\hat{\Lambda}$, choosing a parameter $\alpha \in (0, 1]$ and setting

$$\Lambda^+ = \Lambda + \alpha(\hat{\Lambda} - \Lambda). \tag{21}$$

Often α is chosen to be a value that approximately maximizes O along the search direction $\hat{\Lambda} - \Lambda$. If can be shown, using an elementary argument based on Taylor series, that $O(\Lambda^+) > O(\Lambda)$ provided that α is sufficiently small and that Λ is not itself a solution.

Note that the $\mathcal{A}$-function defined in this way is a particular instance of the model function Φ discussed in Section II, with the additional property of strict concavity. (This property enables the guarantee that $O(\Lambda^+) > O(\Lambda)$ for all α sufficiently small.) The popular model functions that were constructed in Section II all qualify as $\mathcal{A}$-functions, given appropriate assumptions on the smoothness and concavity of the objective O and on the properties of the sets $\mathcal{U}$ and $\mathcal{V}$.

B. EBW as a Model-Based Method

We show here how the "continuous" version of EBW can be expressed as an LSAF / model-based optimization method. EBW is predicated on the objective $O(\Lambda)$ having the particular form of the composition of a scalar function $f(\cdot)$ with a vector function $\xi(\cdot)$, that is,

$$O(\Lambda) = f(\xi(\Lambda)),$$

where ξ is a vector with components ξ_t, $t = 1, 2, \ldots, T$. Given this structure, we have

$$\nabla O(\Lambda) = \sum_{t=1}^{T} \frac{\partial f}{\partial \xi_t} \nabla \xi_t(\Lambda).$$

For certain types of functions f and ξ, the following quantity plays a key role in the algorithm:

$$c_t(\Lambda) := \frac{\partial f}{\partial \xi_t} \xi_t(\Lambda).$$

We demonstrate by considering the following definition of ξ_t, which arises from the particular case of a single multivariate Gaussian density in which the mean μ and covariance matrix Σ are unknown parameters (thus $\Lambda := \{\mu, \Sigma\}$) and x_t, $t = 1, 2, \ldots, T$ are samples. The intermediate variables ξ_t are thus defined by

$$\xi_t := \frac{|\Sigma|^{-1/2}}{(2\pi)^{n/2}} e^{-(x_t-\mu)^T \Sigma^{-1} (x_t-\mu)/2}. \tag{22}$$

The EBW updates for μ and Σ have the following form:

$$\mu^+ := \frac{\sum_t c_t(\Lambda) x_t + D\mu}{\sum_t c_t(\Lambda) + D} \tag{23}$$

$$\Sigma^+ := \frac{\sum_t c_t(\Lambda) x_t x_t^T + D(\mu\mu^T + \Sigma)}{\sum_t c_t(\Lambda) + D} - (\mu^+)(\mu^+)^T. \tag{24}$$

We show how the update formula for μ (23) can be derived from the LSAF approach. Since

$$\nabla_\mu \xi_t(\Lambda) = -\xi_t(\Lambda)\left[\Sigma^{-1}(\mu - x_t)\right],$$

we have

$$\nabla_\mu O(\Lambda) = \sum_{t=1}^T \frac{\partial f}{\partial \xi_t} \nabla_\mu \xi_t(\Lambda) = -\sum_{t=1}^T \frac{\partial f}{\partial \xi_t} \xi_t(\Lambda)\left[\Sigma^{-1}(\mu - x_t)\right]$$
$$= -\Sigma^{-1} \sum_{t=1}^T c_t(\Lambda)(\mu - x_t).$$

We now define the $\mathcal{A}$-function for $O(\Lambda) = f(\xi(\Lambda))$, restricted to the μ component of Λ, to be the following quadratic:

$$A_O(\tilde{\mu}, \mu) := \nabla_\mu O(\Lambda)^T(\tilde{\mu} - \mu) - \frac{1}{2}\beta(\Lambda)(\tilde{\mu} - \mu)^T \Sigma^{-1}(\tilde{\mu} - \mu),$$

where the scaling factor $\beta(\Lambda)$ is

$$\beta(\Lambda) := \sum_{t=1}^T c_t(\Lambda).$$

It is easy to see that the maximizer $\hat{\mu}$ of $A_O(\tilde{\mu}, \mu)$ with respect to $\tilde{\mu}$ satisfies the following condition:

$$\nabla_\mu O(\Lambda) - \beta(\Lambda)\Sigma^{-1}(\tilde{\mu} - \mu) = 0.$$

By substituting for $\nabla_\mu O(\Lambda)$ from the formula above, we have

$$-\Sigma^{-1}\left(\sum_{t=1}^T c_t(\Lambda)(\tilde{\mu} - x_t)\right) - \beta(\Lambda)\Sigma^{-1}(\tilde{\mu} - \mu) = 0,$$

from which, after some manipulation, and by substituting the definition of $\beta(\Lambda)$, we obtain

$$\tilde{\mu} = \frac{\sum_{t=1}^T c_t(\Lambda) x_t}{\sum_{t=1}^T c_t(\Lambda)}.$$

We obtain the EBW update (23) by setting the line-search parameter α in (21) as follows:

$$\alpha = \frac{\beta(\Lambda)}{\beta(\Lambda) + D} = \frac{\sum_{t=1}^T c_t(\Lambda)}{\sum_{t=1}^T c_t(\Lambda) + D}. \tag{25}$$

A similar, albeit more complicated derivation can be done for the update (24).

Note that α can be made arbitrarily small by choosing D sufficiently large. Thus, by applying the Taylor series argument mentioned above, we can show that $O(\Lambda^+) > O(\Lambda)$ for all D sufficiently large—we are guaranteed to see ascent in O.

C. EBW Discussion

We mention a few known results about EBW. The paper [68] describes a unified objective function that includes as special cases two established approaches in discriminative training—Maximum Mutual Information (MMI) and Minimum Classification Error (MCE)—that can be optimized using EBW updates of the form (23). In [69], it was shown that improved recognition accuracy can be obtained by ensuring that the update models do not stray too far from initial models in the EBW update formula. The paper [70] shows that EBW for an MMI objective function can be derived from a regularization based on the Kulback-Leibler (KL) divergence between two probability distributions. It was also observed that an approximate Hessian has been used implicitly to determine step size for each parameter update in (23, 24), so that the EBW update formula is comparable with an approximated quadratic Newton search in terms of convergence behavior (see [9], [10] for details).

In practice there are several strategies for choosing D in (23, 24), and thus the line search parameter α in (25). In [4] it was shown that different Gaussians should use different values for D. Some heuristics for choosing D that are described in , [4] and [10] produce robust convergence results for MMIE, MCE, MPE/MWE and other discriminative training criteria. Another way to find D is to use convex lower bound for some variants of objective functions $f(\xi)$. For example, Afify [71] derived D from a lower bound obtained via the reverse Jensen's inequality [72]. In Section VII, we discuss another technique for lower bounding of D.

VII. A Lower Bound for Discriminative Training

In this section, we extend the discussion of the previous section to emphasize the lower-bounding property of the function that approximates the true objective O at each iteration. This lower-bounding property has proved of particular interest in discriminative training models in speech and language processing applications. It is the key to the derivation of the Generalized Iterative Scaling (GIS) approach. We also discuss connections among the various optimization techniques describe so far.

We say that (twice differentiable) L_O is a lower bound for O if it is an $\mathcal{A}$-function (see Section VI-A) with the following additional property [73, Chapter 9]:

3. $L_O(\tilde{\Lambda}; \Lambda) \leq O(\tilde{\Lambda})$ with equality for $\tilde{\Lambda} = \Lambda$.

This third property actually implies the second property of an $\mathcal{A}$-function. These stronger assumptions on the nature of the approximating function result in stronger convergence properties. In particular, we can obtain the new iterate Λ^+ by solving the following subproblem:

$$\Lambda^+ = \arg\max_{\tilde{\Lambda} \in U} L_O(\tilde{\Lambda}; \Lambda). \tag{26}$$

Unlike in Section VI-A, no line search is necessary; we are guaranteed to have $O(\Lambda^+) > O(\Lambda)$, unless Λ is already a minimizer of O. Further, it can be shown that the iteration sequence converges to a critical point of O [74]. In some cases, the subproblem (26) can be solved analytically. In other cases—for example, Generalized EM [73, Chapter 9.4]—numerical optimization procedures are used to solve this subproblem.

With the EM algorithm for maximum-likelihood estimation of Gaussian mixture models and the Baum-Welch algorithm for maximum-likelihood estimation of HMMs, lower bounds have a long tradition in speech recognition [75]. The introduction of discriminative training for HMMs [4], [76]–[78] raised the question of whether there exist lower bounds for the discriminative formulation as well. In fact, the EBW approach described in the previous section arose in part from consideration of this question.

A. A Lower Bound for Discriminative GMMs and HMMs

We demonstrate the derivation for multivariate Gaussian densities with parameters $\Lambda = \{\mu_s, \Sigma_s, p(s)\}_{s=1}^S$, where the prior $p(s)$ is included. Following (22), but extending to multiple Gaussians, we have the intermediate variables

$$\xi_{ts} = p(s)\frac{|\Sigma_s|^{-1/2}}{(2\pi)^{n/2}}e^{-1(x_t-\mu_s)^T\Sigma_s^{-1}(x_t-\mu_s)/2},$$
$$t = 1, 2, \ldots, T; \quad s = 1, 2, \ldots, S.$$

We assume that the MMI criterion is used, that is,

$$f(\xi_t) = f^+(\xi_t) - f^-(\xi_t) = \log\sum_{s=1}^S \chi_{ts}\xi_{ts} - \log\sum_{s=1}^S \xi_{ts} \quad (27)$$

with $\xi_t := \{\xi_{ts}\}_{s=1,2,\ldots,S}$, $\chi_{t\cdot}$ is the characteristic function to filter out the densities s representing the truth at t. The objective O is similar to the one used in Section VI, that is,

$$O(\Lambda) := \sum_{t=1}^T f(\xi_t(\Lambda)). \quad (28)$$

If this formulation can be reduced to a log-linear model, we can apply existing lower bounds for such models. To this end, we transform the parameters as follows:

$$\xi_{ts} = e^{x_t^T\lambda_s^{(2)}x_t + \lambda_s^{(1)T}x_t + \lambda_s^{(0)}},$$

where $\lambda_s^{(2)}$ is a symmetric $n \times n$ matrix, $\lambda_s^{(1)} \in \mathbb{R}^n$, and $\lambda_s^{(0)} \in \mathbb{R}$. The log-linear parameters, $\Lambda^L = \{\lambda_s^{(2)}, \lambda_s^{(1)}, \lambda_s^{(0)}\}_{s=1}^S$, and the Gaussian parameters, $\Lambda = \{\mu_s, \Sigma_s, p(s)\}_{s=1}^S$, are related as follows:

$$\lambda_s^{(2)} = -\frac{1}{2}\Sigma_s^{-1} \quad \lambda_s^{(1)} = \Sigma_s^{-1}\mu_s$$
$$\lambda_s^{(0)} = -\frac{1}{2}(\mu_s^T\Sigma_s^{-1}\mu_s + \log|2\pi\Sigma_s|) + \log p(s), \quad (29)$$

and

$$\Sigma_s = -\frac{1}{2}(\lambda_s^{(2)} + \Delta\lambda^{(2)})^{-1} \quad \mu_s = \Sigma_s\lambda_s^{(1)}$$
$$p(s) = e^{\lambda_s^{(0)} + \frac{1}{2}(\mu_s^T\Sigma_s^{-1}\mu_s + \log|2\pi\Sigma_s|)}/Z, \quad (30)$$

These transformations do not change the value of the objective $f(\xi_t)$ in (27). In the latter transformation, Z is the normalization constant over s. The (symmetric) matrix $\Delta\lambda^{(2)}$ does not affect the objective and is used only to guarantee positive definite covariance matrices Σ_s. See [30, Chapter 4] and [79] for further details and extensions.

By applying the Expectation-Maximization (EM) bound [80] to f^+ in (27) and the Generalized Iterative Scaling (GIS) bound [81], [82] to f^- in (27), we obtain a lower bound for the objective in the log-linear parameters, $O(\Lambda^L)$:

$$L_O(\tilde{\Lambda}^L; \Lambda^L) = \sum_{p\in\{0,1,2\}}\sum_{s=1}^S \Big((\tilde{\lambda}_s^{(p)} - \lambda_s^{(p)})^T c_s^{(p)+}(\Lambda^L)$$
$$-(e^{F(\tilde{\lambda}_s^{(p)} - \lambda_s^{(p)})}/F)^T c_s^{(p)-}(\Lambda^L)\Big) + r(\Lambda^L).$$

The remaining terms collected in r only depend on the current iterate, Λ^L, but not the parameters to be optimized, $\tilde{\Lambda}^L = \{\tilde{\lambda}_s^{(2)}, \tilde{\lambda}_s^{(1)}, \tilde{\lambda}_s^{(0)}\}_{s=1}^S$, and thus do not affect the optimum of the lower bound. This bound holds true if all feature components x_{td} of the feature vector $x_t = [x_{td}]$ are nonnegative and $F = \max_t\{\sum_{d=1}^D 1 + x_{td} + x_{td}^2\}$ for all t, where F is called the *feature count.* [1] The state occupancies and numerator/denominator statistics are defined as

$$c_{ts}^{+/-}(\Lambda^L) := \frac{\partial f^{+/-}(\xi_t)}{\partial\xi_{ts}}\xi_{ts}(\Lambda^L),$$
$$c_s^{(p)+/-}(\Lambda^L) := \sum_{t=1}^T c_{ts}^{+/-}(\Lambda^L)x_t^p. \quad (31)$$

This lower bound can be maximized analytically, leading to the following update rules:

$$\lambda_s^{(p)+} = \lambda_s^{(p)} + \frac{1}{F}\log\left(\frac{c_s^{(p)+}}{c_s^{(p)-}}\right), \quad p = 0, 1, 2, \quad (32)$$

wit the new iterate $\Lambda^{L+} = \{\lambda_s^{(2)+}, \lambda_s^{(1)+}, \lambda_s^{(0)+}\}_{s=1}^S$. The Gaussian parameters can be recovered by applying the transformations in (30). It is tempting to translate the update rules into the Gaussian domain. We refrain from doing so here because the equivalence makes it unnecessary, and because the mixing of the different types of parameters in (30) does not allow further insight to be gained from a simple analytical expression.

This lower bound can be extended to more sophisticated models and training criteria, including mixture models, HMMs, and MPE. In general, however, the convergence speed of this approach is slow; see [30, Chapter 6] and [13], [83].

B. Comparison with Other Approaches

Like the iteration constant D for EBW in (23) and (24), the feature count F defined above controls the convergence speed. In contrast to D, the value of F is known and can be computed efficiently before training.

The parameter α for the line-search A-function technique (Section VI-A) is determined by a numerical line search. Alternatively, we can use a lower bound along the search direction $\hat{\Lambda}^L - \Lambda^L$ to compute an analytical value for α (and thus D in EBW). The lower bound above restricted to the search direction $\hat{\Lambda}^L - \Lambda^L$ implies the intermediate variables

$$\xi_{ts}(\nu, \bar{\nu}) = \xi_{ts}(\Lambda^L) \cdot e^{\nu f(x_t,s) + \bar{\nu}\bar{f}(x_t,s)}$$

[1] Non-negative features are obtained by applying a suitable affine transformation. The sum constraint can be satisfied by introducing a dummy feature, which cancels for this specific choice of feature functions.

with $\Lambda^L = \{\lambda_s^{(2)}, \lambda_s^{(1)}, \lambda_s^{(0)}\}_{s=1}^S$,$\hat{\Lambda}^L = \{\hat{\lambda}_s^{(2)}, \hat{\lambda}_s^{(1)}, \hat{\lambda}_s^{(0)}\}_{s=1}^S$, $\nu, \bar{\nu} \in \mathbb{R}$, feature function $f(x,s) = \sum_{p\in\{0,1,2\}} (\hat{\lambda}_s^{(p)} - \lambda_s^{(p)})^T x^p$ and a dummy feature function $\bar{f}(x,s) = F - f(x,s)$, the feature count is chosen such that $f(x,s) \geq 0$. Optimization of the associated lower bound yields:

$$\nu = \frac{1}{F}\log\left(\frac{\sum\limits_{p\in\{0,1,2\}}\sum\limits_{s}\left(\hat{\lambda}_s^{(p)} - \lambda_s^{(p)}\right)^T c_s^{(p)+}(\Lambda^L)}{\sum\limits_{p\in\{0,1,2\}}\sum\limits_{s}\left(\hat{\lambda}_s^{(p)} - \lambda_s^{(p)}\right)^T c_s^{(p)-}(\Lambda^L)}\right)$$

$$\bar{\nu} = \frac{1}{F}\log\left(\frac{TF - \sum\limits_{p\in\{0,1,2\}}\sum\limits_{s}\left(\hat{\lambda}_s^{(p)} - \lambda_s^{(p)}\right)^T c_s^{(p)+}(\Lambda^L)}{TF - \sum\limits_{p\in\{0,1,2\}}\sum\limits_{s}\left(\hat{\lambda}_s^{(p)} - \lambda_s^{(p)}\right)^T c_s^{(p)-}(\Lambda^L)}\right).$$

After the optimization of the lower bound, we incorporate the dummy feature back into the regular feature, $\nu f(x,s) + \bar{\nu}\bar{f}(x,s) = \nu f(x,s) + \bar{\nu}(F - f(x,s)) = (\nu - \bar{\nu})f(x,s) + \bar{\nu}F$. We ignore the constant term $\bar{\nu}F$ that cancels in the objective and identify the parameter α with $\nu - \bar{\nu}$ restricted to the line segment

$$\alpha = \min\{\nu - \bar{\nu}, 1\}.$$

These equations are for the log-linear parameters. The Gaussian parameters can always be obtained with (30), although it does not allow for a simple expression in the Gaussian domain.

Finally, we make the link to gradient descent. Recalling the definitions of $O(\Lambda)$ (28), f^+ and f^- (27), and $c_s^{(p)+}$ and $c_s^{(p)-}$ (31), we have

$$\nabla_{\lambda_s^{(p)}} O(\Lambda^L) = c_s^{(p)+}(\Lambda^L) - c_s^{(p)-}(\Lambda^L).$$

The update rules (32) can be linearly approximated around a critical point $\nabla O(\Lambda^L) = 0$ by using the approximation $\log(1+\alpha) \approx \alpha + O(\alpha^2)$ to obtain

$$\lambda_s^{(p)+} = \lambda_s^{(p)} + \frac{1}{F}\left(\frac{\nabla_{\lambda_s^{(p)}} O(\Lambda^L)}{c_s^{(p)+}(\Lambda^L)}\right) + O\left(\left(\frac{\nabla_{\lambda_s^{(p)}} O(\Lambda^L)}{c_s^{(p)+}(\Lambda^L)}\right)^2\right).$$

Hence, the step is a diagonally scaled gradient ascent step, with diagonal scaling matrix, with diagonal entries $\left(\frac{1}{Fc_s^{(p)+}(\Lambda^L)}\right)$.

A related optimization technique, Rprop ("resilient backpropagation"), is a gradient-based, batch update algorithm that uses adaptive step sizes and will be briefly discussed in the next section. Compared to the step sizes in (32), $\Delta_s^{(p)} = \frac{1}{F}\left|\log\left(\frac{c_s^{(p)+}}{c_s^{(p)-}}\right)\right|$ (for this choice of model), Rprop uses empirically adapted step sizes Δ_i.

VIII. The Optimization Technique of Rprop

To continue the discussion on the topic of non-obvious connections among a number of popular optimization techniques used by the speech and language processing community, in this section, we discuss the optimization technique of Rprop ("resilient backpropagation").

Rprop was originally introduced for training of multilayer feedforward networks [84], but recent reports indicate its successful deployment in speech recognition [8], [30], [78].

Rprop only uses the sign of the partial derivatives of the training objective for the parameter update, rather than the actual values of the gradient. Specifically, the components of Λ are updated as follows:

$$\Lambda_i^+ = \Lambda_i + \text{sign}(\nabla_{\Lambda_i} O(\Lambda))\Delta_i, \quad i = 1, 2, \ldots, d,$$

where a separate step size $\Delta_i \geq 0$ is maintained independently for each component $i = 1, 2, \ldots, d$, according to a simple heuristic. If the sign of the partial derivative changed over the last iteration, the step size is reduced by the positive factor $\eta^- < 1$. If the partial derivative kept the same sign, the step size is increased by the factor $\eta^+ > 1$. That is, we have

$$\Delta_i^+ = \begin{cases} \eta^+\Delta_i, & \text{if } \nabla_{\Lambda_i} O(\Lambda) \cdot \nabla_{\Lambda_i} O(\Lambda^+) > 0 \\ \eta^-\Delta_i, & \text{if } \nabla_{\Lambda_i} O(\Lambda) \cdot \nabla_{\Lambda_i} O(\Lambda^+) < 0 \\ 0, & \text{otherwise.} \end{cases}$$

The factors η^+ and η^- are set empirically; values that have been found to work well in practice are $\eta^+ = 1.2$ and $\eta^- = 0.5$. The parameter constraints for Gaussian mixture models such as the normalization of the mixture weights are reimposed after each iteration.

Note that Rprop can be viewed as a search-direction method, of the type described at the end of Section II, where we define the direction to be

$$\Xi^k := [\text{sign}\,(\nabla_{\Lambda_i} O(\Lambda^k))\Delta_i]_{i=1,2,\ldots,d}.$$

Since the components of Ξ^k and $\nabla O(\Lambda^k)$ all have the same sign, by definition, the condition (10) is satisfied, indicating that Ξ^k is a direction of ascent for the objective O.

IX. Summary and Discussions

We have outlined a variety of optimization techniques that are used in the processing of speech and language data. Though far from exhaustive, our survey demonstrates the centrality of optimization methodology and the inventive ways in which optimization techniques have been adapted to the particular structure of these problems and to the needs arising from the applications.

We emphasize the importance of connecting researchers in the two largely separate communities of optimization and of speech and language processing. The topics selected for this review cover both optimization theory/algorithms and related speech/language applications with the goal of demonstrating the role of optimization in the applications and the insight that it provides. We have covered some optimization methods such as EBW in detail because they have found successful uses in machine translation and speech recognition applications and because they are equipped with strong theoretical properties which serve well for unified treatment of a range of optimization methods popular in applications.

One key novelty of the paper is to show how popular techniques in speech and language processing are related to general optimization methods, an insight which has not been apparent to the speech and language processing community, which nevertheless has exploited the techniques extensively. As one example: Section VII described a novel view on how EBW can be related to the lower-bound method GIS.

We have not addressed in detail the issues that arise when the objective function to be maximized is nonconcave, as is the case in several applications in speech processing, including deep learning. In such cases, algorithms based on model functions or search directions (as discussed in Section II) can generally guarantee convergence only to stationary points of the problem, which are not necessarily global or even local solutions. Algorithms that find the global solution are not generally available, or are extremely compute-intensive. However, good practical results can often be obtained with the local algorithms that are the focus of this paper. A consequence of nonconvexity is that different algorithms may identify different local solutions. (In fact, the same algorithm may give a different result if started from a different point.) Other factors that may contribute to varying results obtained with the different methods include different termination criteria, different convergence speeds (leading to premature termination for slowly converging methods), suboptimal tuning of the algorithmic parameters, numerical stability, and so on. Some issues along these lines are discussed in [8], [30].

While the optimization techniques in speech and language processing discussed in this paper were developed in the past with little input from specialist optimization researchers, the situation is changing as the amount of data and the complexity of processing tasks continue to grow. The need for more powerful optimization approaches is leading to collaborative work between the data processing community (including speech/language processing and more general machine learning [85]) and the optimization community on such topics as Hessian-free methods [12], [86], quasi-Newton methods (including L-BFGS), and stochastic gradient [87]. The trend of such collaborations has already arrived in speech recognition with the use of deep neural networks [12], [86], [88]–[92] and other types of deep learning architectures [93]–[95], and in language processing with the use of recurrent neural networks [96]. While this trend is too new to be surveyed in this article, we expect new optimization challenges to arise as deep learning and other techniques that depend heavily on optimization technology are deployed in demanding speech and language processing problems [97].

References

[1] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 1966.

[2] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput., Speech, Lang.*, vol. 9, pp. 171–185, 1995.

[3] R. O. Duda, P. E. Hart, and D. G. Stork*, Pattern Classification*. New York, NY, USA: Wiley, 2001.

[4] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden markov models for speech recognition," *Comput. Speech, Lang.*, pp. 25–47, 2002.

[5] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, Cambridge, U.K., 2003.

[6] L. Deng and D. O'Shaughnessy*, Speech Processing—A Dynamic and Optimization-Oriented Approach*. New York, NY, USA: Marcel Dekker, 2003.

[7] W. Macherey and H. Ney, "A comparative study on maximum entropy and discriminative training for acoustic modeling in automatic speech recognition," in *Proc. Eurospeech*, 2003, pp. 493–496.

[8] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Interspeech*, Sep. 2005.

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

[10] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition—a unifying review for optimization-oriented speech recognition," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 14–36, Sep. 2008.

[11] T. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, and A. Sethy, "Exemplar-based sparse representation features for speech recognition," in *Proc. Interspeech*, 2010.

[12] T. Sainath, B. Kingsbury, H. Soltau, and B. Ramabhadran, "Optimization techniques to improve training speech of deep belief networks for large speech tasks," *IEEE Trans. Audio, Speech, Lang. Process., Spec. Iss. Large-Scale Optimization for Audio, Speech, Lang. Process.*, vol. 21, no. 11, Nov. 2013.

[13] G. Heigold, S. Hahn, P. Lehnen, and H. Ney, "EM-style optimization of hidden conditional random fields for grapheme-to-phoneme conversion," in *Proc. ICASSP*, 2011, pp. 4920–4923.

[14] V. Hautamäki, K. A. Lee, T. Kinnunen, B. Ma, and H. Li, "Optimizing the performance of spoken language recognition with discriminative training," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 8, pp. 1622–1631, Aug. 2013.

[15] T. P. Minka, "Algorithms for maximum-likelihood logistic regression," Carnegie Mellon Univ., Tech. Rep. 738, 2001.

[16] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech Lang.*, pp. 210–229, Apr. 2006.

[17] G. Saon and J.-T. Chien, "Bayesian sensing hidden Markov models," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 20, no. 1, pp. 43–54, Jan. 2012.

[18] C. H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proc. IEEE*, vol. 88, no. 8, pp. 1241–1269, Aug. 2000.

[19] S. Watanabe and A. Nakamura, "Bayesian approaches to acoustic modeling: A Review," *APSIPA Trans. Signal Inf. Process.*, vol. 1, 2012.

[20] L. Xiao and L. Deng, "A geometric perspective of large-margin training of Gaussian models," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 118–123, Nov. 2010.

[21] T.-H. Chang, Z.-Q. Luo, L. Deng, and C.-Y. Chi, "A convex optimization method for joint mean and variance parameter estimation of large-margin CDHMM," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 4053–4056.

[22] X. He and L. Deng, "Speech-centric information processing: An optimization-oriented approach," *Proc. IEEE*, vol. 101, no. 5, pp. 1116–1135, May 2013.

[23] J. Nocedal and S. J. Wright*, Numerical Optimization*, 2nd ed. ed. New York, NY, USA: Springer, 2006.

[24] S. P. Boyd and L. Vandenberghe*, Convex optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[25] B. T. Polyak*, Introduction to Optimization*. Ann Arbor, MI, USA: Optimization Software, 1987.

[26] C. Liu, Y. Hu, L.-R. Dai, and H. Jiang, "Trust region-based optimization for maximum mutual information estimation of hmms in speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2474–2485, Nov. 2011.

[27] M. Gibson and T. Hain, "Error approximation and minimum phone error acoustic model estimation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1269–1279, Aug. 2010.

[28] S. Chen and R. Rosenfeld, A Gaussian prior for smoothing maximum entropy models Comput. Sci. Dept., Carnegie Mellon Univ., Tech. Rep. CMUCS-99-108, 1999.

[29] 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. Acoustic, Speech, Signal Process.*, Las Vegas, NV, USA, Apr. 2008, pp. 4057–4060.

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

[31] Y. Bar-Hillel, "The present status of automatic translation of languages," *Adv. Comput.*, pp. 158–163, 1960.

[32] P. Brown, S. Pietra, V. Pietra, and R. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, 1993.

[33] P. Koehn, F. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. HLT-NAACL*, 2003.

[34] F. Och, "Minimum error rate training in statistical machine translation," in *Proc. ACL*, 2003.

[35] X. He, L. Deng, and W. Chou, "Speech recognition, machine translation, and speech translation–a unified discriminative learning paradigm," *IEEE Signal Process. Mag.*, vol. 28, no. 5, pp. 126–133, Sep. 2011.

[36] D. Xiong, M. Zhang, and H. Li, "A maximum-entropy segmentation model for statistical machine translation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2494–2505, Nov. 2011.

[37] I. D. El-Kahlout and K. Oflazer, "Exploiting morphology and local word reordering in english-to-turkish phrase-based statistical machine translation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1313–1322, Aug. 2010.

[38] X. He and L. Deng, "Maximum expected bleu training of phrase and lexicon translation models," in *Proc. ACL, Assoc. Comput. Linguist.*, 2012.

[39] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist.*, 2002, pp. 311–318.

[40] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Commun.*, vol. 52, no. 1, pp. 12–40, 2010.

[41] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19–41, Jan. 2000.

[42] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 5, pp. 980–988, Jul. 2008.

[43] K. A. Lee, C. H. You, H. Li, T. Kinnunen, and K. C. Sim, "Using discrete probabilities with bhattacharyya measure for svm-based speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 861–870, May 2011.

[44] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[45] H. Li, B. Ma, and K. A. Lee, "Spoken language recognition: From fundamentals to practice," *Proc. IEEE*, vol. 101, no. 5, pp. 1136–1159, May 2013.

[46] H. Li, B. Ma, and C.-H. Lee, "Vector-based spoken language classification," in *Springer Handbook of Speech Processing*, J. Benesty, M. Sondhi, and A. Huang, Eds. New York, NY, USA: Springer, 2007.

[47] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 271–284, Jan. 2007.

[48] N. Brümmer and J. Preez, "Application-independent evaluation of speaker detection," *Comput. Speech Lang.*, vol. 20, no. 2, pp. 230–275, 2006.

[49] D. A. van Leeuwen and N. Brümmer, "An introduction to application independent evaluation of speaker recognition systems," in *Speaker Classification, Lecture Notes in Computer Science/Artificial Intelligence*, R. Müller, Ed. New York, NY, USA: Springer, 2007, vol. 4343.

[50] H. Li and B. Ma, "TechWare: Speaker and spoken language recognition resources," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 139–142, Nov. 2010.

[51] A. F. Martin and A. N. Le, "NIST 2007 language recognition evaluation," in *Proc. Odyssey: Speaker Lang. Recogn. Workshop*, 2008, p. 016.

[52] A. F. Martin and A. N. Le, "The current state of language recognition: NIST 2005 evaluation results," in *Proc. Odyssey: Speaker Lang. Recogn. Workshop*, 2006, pp. 1–6.

[53] A. F. Martin and J. S. Garofolo, "NIST speech processing evaluations: Lvcsr, speaker recognition, language recognition," in *Proc. IEEE Workshop Signal Process. Applicat. Public Security Forensics*, 2007, pp. 1–7.

[54] A. F. Martin and C. Greenberg, "NIST 2009 language recognition evaluation," in *Proc. Odyssey: Speaker Lang. Recogn. Workshop*, 2010, pp. 165–171.

[55] D. Zhu, H. Li, B. Ma, and C. H. Lee, "Optimizing the performance of spoken language recognition with discriminative training," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 8, pp. 1642–1653, Nov. 2008.

[56] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.

[57] N. Brümmer, "Focal bilinear: Tools for detector fusion and calibration, with use of side-information," [Online]. Available: https://sites.google.com/site/nikobrummer/focalbilinear.

[58] N. Brümmer, "Measuring, refining and calibrating speaker and language information extracted from speech," Ph.D. dissertation, Stellenbosch Univ., Stellenbosch, South Africa, 2010.

[59] N. Brümmer and D. Leeuwen, "On calibration of language recognition scores," in *Proc. IEEE Odyssey: Speaker Lang. Recogn. Workshop*, 2006, pp. 1–8.

[60] P. S. Gopalakrishnan, D. Kanevsky, D. Nahamoo, and A. Nadas, "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.

[61] Y. Normandin, "An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition," in *Proc. ICASSP*, 1991, pp. 537–540.

[62] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Univ. of Cambridge, Cambridge, U.K., 2003.

[63] D. Kanevsky, "Extended Baum transformations for general functions, II," Human Language Technol., IBM, Tech. Rep. RC23645(W0506-120), 2005.

[64] T. Jebara, "On reversing Jensen's inequality," in *Proc. NIPS*, 2002.

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

[66] D. Kanevsky, T. Sainath, B. Ramabhadran, and D. Nahamoo, "Generalization of extended Baum-Welch parameter estimation for discriminative training and decoding," in *Proc. Interspeech*, 2008.

[67] D. Kanevsky, D. Nahamoo, T. N. Sainath, B. Ramabhadran, and P. A. Olsen, "A-Functions: A generalization of extended Baum-Welch transformations to convex optimization," in *Proc. ICASSP*, 2011, pp. 5164–5167.

[68] R. Schlüter, W. Macherey, B. Müller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Commun.*, pp. 287–310, 2001.

[69] C. Liu, P. Liu, H. Jiang, F. Soong, and R. Wang, "Constrained Line Search Optimization for Discriminative Training in Speech Recognition," in *Proc. ICASSP*, 2007, pp. 329–332.

[70] DR. Hsiao and T. Schultz, "Generalized Baum-Welch algorithm and its application to new extended Baum-Welch algorithm," in *Proc. Interspeech*, 2011.

[71] M. Afify, "Extended Baum-Welch reestimation of Gaussian mixture models based on reverse Jensen inequality," in *Proc. Interspeech*, Lisbon, Portugal, Sep. 2005.

[72] T. Jebara and A. Pentland, "On reversing Jensen's inequality," *Adv. Neural Inf. Process. Syst.*, Dec. 2000.

[73] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[74] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.

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

[76] Y. Normandin, R. Cardin, and R. Demori, "High-performance connected digit recognition using maximum mutual information estimation," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 299–311, Apr. 1994.

[77] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. Acoustic, Speech, Signal Process.*, 2002, pp. 105–108.

[78] E. McDermott, T. J. Hazen, J. Le 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.

[79] G. Heigold, H. Ney, P. Lehnen, T. Gass, and R. Schlüter, "Equivalence of generative and log-linear models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 5, pp. 1138–1148, Jul. 2011.

[80] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. Ser. B.*, vol. 39, 1977.

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

[82] S. A. Della Pietra, V. J. 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.

[83] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "GIS-like estimation of log-linear models with hidden variables," in *Proc. ICASSP*, 2008, pp. 4045–4048.

[84] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm," in *IEEE International Conference on Neural Networks (ICNN)*, 1993.

[85] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 5, pp. 1060–1089, May 2013.

[86] B. Kingsbury, T. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. Interspeech*, 2012.

[87] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. W. Mao, M.-A. Ranzato, A.-W. Senior, P. A. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," *NIPS*, 2012.

[88] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[89] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[90] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[91] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 7–13, Jan. 2012.

[92] M. Siniscalchi, L. Deng, D. Yu, and C.-H. Lee, "Exploiting deep neural networks for detection-based speech recognition," *Neurocomputing*, pp. 148–157, 2013.

[93] L. Deng and D. Yu, "Deep convex network: A scalable architecture for speech pattern classification," in *Proc. Interspeech*, 2011.

[94] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 2133–2136.

[95] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1944–1957, Aug. 2013.

[96] I. Suskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011.

[97] L. Deng, G. E. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Proc. ICASSP*, 2013.

Stephen J. Wright, photograph and biography not available at the time of publication.

Dimitri Kanevsky, photograph and biography not available at the time of publication.

Li Deng, photograph and biography not available at the time of publication.

Xiaodong He, photograph and biography not available at the time of publication.

Georg Heigold, photograph and biography not available at the time of publication.

Haizhou Li, photograph and biography not available at the time of publication.