Machine Learning Paradigms for Speech Recognition: An Overview

Li Deng, *Fellow, IEEE*, and Xiao Li, *Member, IEEE*

***Abstract*—Automatic Speech Recognition (ASR) has historically been a driving force behind many machine learning (ML) techniques, including the ubiquitously used hidden Markov model, discriminative learning, structured sequence learning, Bayesian learning, and adaptive learning. Moreover, ML can and occasionally does use ASR as a large-scale, realistic application to rigorously test the effectiveness of a given technique, and to inspire new problems arising from the inherently sequential and dynamic nature of speech. On the other hand, even though ASR is available commercially for some applications, it is largely an unsolved problem—for almost all applications, the performance of ASR is not on par with human performance. New insight from modern ML methodology shows great promise to advance the state-of-the-art in ASR technology. This overview article provides readers with an overview of modern ML techniques as utilized in the current and as relevant to future ASR research and systems. The intent is to foster further cross-pollination between the ML and ASR communities than has occurred in the past. The article is organized according to the major ML paradigms that are either popular already or have potential for making significant contributions to ASR technology. The paradigms presented and elaborated in this overview include: generative and discriminative learning; supervised, unsupervised, semi-supervised, and active learning; adaptive and multi-task learning; and Bayesian learning. These learning paradigms are motivated and discussed in the context of ASR technology and applications. We finally present and analyze recent developments of deep learning and learning with sparse representations, focusing on their direct relevance to advancing ASR technology.**

***Index Terms*—Machine learning, speech recognition, supervised, unsupervised, discriminative, generative, dynamics, adaptive, Bayesian, deep learning.**

I. Introduction

In recent years, the machine learning (ML) and automatic speech recognition (ASR) communities have had increasing influences on each other. This is evidenced by a number of dedicated workshops by both communities recently, and by the fact that major ML-centric conferences contain speech processing sessions and vice versa. Indeed, it is not uncommon for the ML community to make assumptions about a problem, develop precise mathematical theories and algorithms to tackle the problem given those assumptions, but then evaluate on data sets that are relatively small and sometimes synthetic. ASR research, on the other hand, has been driven largely by rigorous empirical evaluations conducted on very large, standard corpora from real world. ASR researchers often found formal theoretical results and mathematical guarantees from ML of less use in preliminary work. Hence they tend to pay less attention to these results than perhaps they should, possibly missing insight and guidance provided by the ML theories and formal frameworks even if the complex ASR tasks are often beyond the current state-of-the-art in ML.

This overview article is intended to provide readers of IEEE Transactions on Audio, Speech, and Language Processing with a thorough overview of the field of modern ML as exploited in ASR's theories and applications, and to foster technical communications and cross pollination between the ASR and ML communities. The importance of such cross pollination is twofold: First, ASR is still an unsolved problem today even though it appears in many commercial applications (e.g. iPhone's Siri) and is sometimes perceived, incorrectly, as a solved problem. The poor performance of ASR in many contexts, however, renders ASR a frustrating experience for users and thus precludes including ASR technology in applications where it could be extraordinarily useful. The existing techniques for ASR, which are based primarily on the hidden Markov model (HMM) with Gaussian mixture output distributions, appear to be facing diminishing returns, meaning that as more computational and data resources are used in developing an ASR system, accuracy improvements are slowing down. This is especially true when the test conditions do not well match the training conditions [1], [2]. New methods from ML hold promise to advance ASR technology in an appreciable way. Second, ML can use ASR as a large-scale, realistic problem to rigorously test the effectiveness of the developed techniques, and to inspire new problems arising from special sequential properties of speech and their solutions. All this has become realistic due to the recent advances in both ASR and ML. These advances are reflected notably in the emerging development of the ML methodologies that are effective in modeling deep, dynamic structures of speech, and in handling time series or sequential data and nonlinear interactions between speech and the acoustic environmental variables which can be as complex as mixing speech from other talkers; e.g., [3]–[5].

The main goal of this article is to offer insight from multiple perspectives while organizing a multitude of ASR techniques into a set of well-established ML schemes. More specifically, we provide an overview of common ASR techniques by establishing several ways of categorization and characterization of the common ML paradigms, grouped by their learning

Manuscript received December 02, 2011; revised June 04, 2012 and October 13, 2012; accepted December 21, 2012. Date of publication January 30, 2013; date of current version nulldate. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhi-Quan (Tom) Luo.

L. Deng is with Microsoft Research, Redmond, WA 98052 USA (e-mail: deng@microsoft.com).

X. Li was with Microsoft Research, Redmond, WA 98052 USA. She is now with Facebook Corporation, Palo Alto, CA 94025 USA (e-mail: mimily@gmail.com).

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.2013.2244083

styles. The learning styles upon which the categorization of the learning techniques are established refer to the key attributes of the ML algorithms, such as the nature of the algorithm's input or output, the decision function used to determine the classification or recognition output, and the loss function used in training the models. While elaborating on the key distinguishing factors associated with the different classes of the ML algorithms, we also pay special attention to the related arts developed in ASR research.

In its widest scope, the aim of ML is to develop automatic systems capable of generalizing from previously observed examples, and it does so by constructing or learning functional dependencies between arbitrary input and output domains. ASR, which is aimed to convert the acoustic information in speech sequence data into its underlying linguistic structure, typically in the form of word strings, is thus fundamentally an ML problem; i.e., given examples of inputs as the continuous-valued acoustic feature sequences (or possibly sound waves) and outputs as the nominal (categorical)-valued label (word, phone, or phrase) sequences, the goal is to predict the new output sequence from a new input sequence. This prediction task is often called *classification* when the temporal segment boundaries of the output labels are assumed known. Otherwise, the prediction task is called *recognition*. For example, phonetic classification and phonetic recognition are two different tasks: the former with the phone boundaries given in both training and testing data, while the latter requires no such boundary information and is thus more difficult. Likewise, isolated word "recognition" is a standard classification task in ML, except with a variable dimension in the input space due to the variable length of the speech input. And continuous speech recognition is a special type of structured ML problems, where the prediction has to satisfy additional constraints with the output having structure. These additional constraints for the ASR problem include: 1) linear sequence in the discrete output of either words, syllables, phones, or other finer-grained linguistic units; and 2) segmental property that the output units have minimal and variable durations and thus cannot switch their identities freely.

The major components and topics within the space of ASR are: 1) feature extraction; 2) acoustic modeling; 3) pronunciation modeling; 4) language modeling; and 5) hypothesis search. However, to limit the scope of this article, we will provide the overview of ML paradigms mainly on the acoustic modeling component, which is arguably the most important one with greatest contributions to and from ML.

The remaining portion of this paper is organized as follows: We provide background material in Section II, including mathematical notations, fundamental concepts of ML, and some essential properties of speech subject to the recognition process. In Sections III and IV, two most prominent ML paradigms, generative and discriminative learning, are presented. We use the two axes of modeling and loss function to categorize and elaborate on numerous techniques developed in both ML and ASR areas, and provide an overview on the generative and discriminative models in historical and current use for ASR. The many types of loss functions explored and adopted in ASR are also reviewed. In Section V, we embark on the discussion of active learning and semi-supervised learning, two different but closely related ML paradigms widely used in ASR. Section VI is devoted to transfer learning, consisting of adaptive learning and multi-task learning, where the former has a long and prominent history of research in ASR and the latter is often embedded in the ASR system design. Section VII is devoted to two emerging areas of ML that are beginning to make inroad into ASR technology with some significant contributions already accomplished. In particular, as we started writing this article in 2009, deep learning technology was only taking shape, and now in 2013 it is gaining full momentum in both ASR and ML communities. Finally, in Section VIII, we summarize the paper and discuss future directions.

TABLE I
DEFINITIONS OF A SUBSET OF COMMONLY USED SYMBOLS AND NOTATIONS IN THIS ARTICLE

Symbol	Meaning
$\mathcal{X}$	Space of input vectors
$\mathcal{Y}$	Set of output labels
$p(\mathbf{x}, y)$	Joint distribution $p(\mathbf{X} = \mathbf{x}, Y = y)$
$\mathcal{F}$	Space of decision functions $f : \mathcal{X} \to \mathcal{Y}$
$f(\mathbf{x}; \lambda)$	Decision function
$d_y(\mathbf{x}; \lambda)$	Discriminant function
λ	Model or decision function parameters
$L(f(\mathbf{x}), y)$	Loss function
$\mathrm{E}_{p(\mathbf{x},y)}[\cdot]$	Expectation $\mathrm{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\cdot]$
$\mathcal{D}$	Training data $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})_{i=1}^{m}$

II. BACKGROUND

A. Fundamentals

In this section, we establish some fundamental concepts in ML most relevant to the ASR discussions in the remainder of this paper. We first introduce our mathematical notations in Table 1.

Consider the canonical setting of classification or regression in machine learning. Assume that we have a training set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$ drawn from the distribution $p(\mathbf{x}, y)$, $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$. The goal of learning is to find a *decision function* $f : \mathcal{X} \to \mathcal{Y}$ that correctly predicts the output of a future input drawn from the same distribution. The prediction task is called *classification* when the output takes categorical values, which we assume in this work. ASR is fundamentally a classification problem. In a multi-class setting, a decision function is determined by a set of *discriminant functions*, i.e.,

$$f(\mathbf{x}) = \arg\max_{y} d_y(\mathbf{x}); \tag{1}$$

Each discriminant function d_y is a class-dependent function of $\mathbf{x}$. In binary classification where $\mathcal{Y} = \{\pm 1\}$, however, it is common to use a single "discriminant function" as follows,

$$f(\mathbf{x}) = \operatorname{sgn} d(\mathbf{x}) \tag{2}$$

Formally, learning is concerned with finding a decision function (or equivalently a set of discriminant functions) that minimizes the *expected risk*, i.e.,

$$R_p(f) = \mathrm{E}_{p(\mathbf{x},y)}\left[L\left(f(\mathbf{x}), y\right)\right] \tag{3}$$

under some *loss function* $L(f(\mathbf{x}), y)$. Here the loss function measures the "cost" of making the decision $f(\mathbf{x})$ while the true

output is y; and the expected risk is simply the expected value of such a cost. In ML, it is important to understand the difference between the decision function and the loss function. The former is often referred to as the "model". For example, a linear model is a particular form of the decision function, meaning that input features are linearly combined at classification time. On the other hand, how the parameters of a linear model are estimated depends on the loss function (or, equivalently, the training objective). A particular model can be estimated using different loss functions, while the same loss function can be applied to a variety of models. We will discuss the choice of models and loss functions in more detail in Section III and Section IV.

Apparently, the expected risk is hard to optimize directly as $p(\mathbf{x}, y)$ is generally unknown. In practice, we often aim to find a decision function that minimizes the *empirical risk*, i.e.,

$$R_{\text{emp}}(f) = \frac{1}{m}\sum_{i=1}^{m} L\left(f(\mathbf{x}^{(i)}), y^{(i)}\right) \tag{4}$$

with respect to the training set. It has been shown that, if L satisfies certain constraints, $R_{\text{emp}}(f)$ converges to $R_p(f)$ in probability for any f [6]. The training set, however, is almost always insufficient. It is therefore crucial to apply certain type of regularization to improve generalization. This leads to a practical training objective referred to as *accuracy-regularization* which takes the following general form:

$$J(f) = R_{\text{emp}}(f) + \gamma C(f) \tag{5}$$

where $C(f)$ is a regularizer that measures "complexity" of f, and γ is a tradeoff parameter.

In fact, a fundamental problem in ML is to derive such forms of $C(f)$ that guarantee the generalization performance of learning. Among the most popular theorems on generalization error bound is the VC bound theorem [7]. According to the theorem, if two models describe the training data equally well, the model with the smallest VC dimension has better generalization performance. The VC dimension, therefore, can naturally serve as a regularizer in empirical risk minimization, provided that it has a mathematically convenient form, as in the case of large-margin hyperplanes [7], [8].

Alternatively, regularization can be viewed from a Bayesian perspective, where f itself is considered a random variable. One needs to specify a prior belief, denoted as $p(f)$, *before* seeing the training data ($\mathcal{D}$). In contrast, the posterior probability of the model is derived *after* training data is observed:

$$q(f) = p(f|\mathcal{D}) = \frac{p(\mathcal{D}|f)p(f)}{p(\mathcal{D})}, \tag{6}$$

Maximizing (6) is known as *maximum a posteriori* (MAP) estimation. Notice that by taking logarithm, this learning objective fits the general form of (5); $R_{\text{emp}}(f)$ is now represented by a particular loss function $\ln p(\text{Data}|f)$ and $C(f)$ by $-\ln p(f)$. The choice of the prior distribution has usually been a compromise between a realistic assessment of beliefs and choosing a parametric form that simplifies analytical calculations. In practice, certain forms of the prior are preferred due mainly to their mathematical tractability. For example, in the case of generative models, a *conjugate prior* $\pi(f)$ with respect to the joint sample distribution $p(\mathbf{x}, y|f)$ is often used, so that the posterior $p(f|\mathbf{x}, y)$ belongs to the same functional family as the prior.

All discussions above are based on the goal of finding a point-estimate of the model. In the Bayesian approach, it is often beneficial to have a decision function that takes into account the uncertainty of the model itself. A *Bayesian predictive classifier* is precisely for this purpose:

$$f_{Bayes}(\mathbf{x}) \triangleq \mathrm{E}_{q(f)}\left[f(\mathbf{x})\right] \tag{7}$$

In other words, instead of using one point-estimate of the model (as is in MAP), we consider the entire posterior distribution, thereby making the classification decision less subject to the variance of the model.

The use of Bayesian predictive classifiers apparently leads to a different learning objective; it is now the posterior distribution $q(f)$ that we are interested in estimating as opposed to a particular f. As a result, the training objective becomes $R_{p(\mathbf{x},y)}(q(f))$. Similar to our earlier discussion, this objective can be estimated via empirical risk minimization with regularization. For example, McAllester's PAC-Bayesian bound [9] suggests the following training objective,

$$q^*(f) = \arg\min_q \left(\mathrm{E}_{q(f)}\left[R_{\text{emp}}(f)\right] + \lambda D\left(q(f)\|p(f)\right)\right) \tag{8}$$

which finds a posterior distribution that minimizes both the marginalized empirical risk as well as the divergence from the prior distribution of the model. Similarly, *Maximum entropy discrimination* [10] seeks $q(f)$ that minimizes $D(q(f)\|p(f))$ under the constraints that $\mathrm{E}_{q(f)}[\mathrm{E}_{p(\gamma_i)}[(y_i f(\mathbf{x}_i) - \gamma_i)]] \geq 0$.

Finally, it is worth noting that Bayesian predictive classifiers should be distinguished from the notion of *Bayesian minimum risk* (BMR) classifiers. The latter is a form of point-estimate classifiers in (1) that are based on Bayesian probabilities. We will discuss BMR in detail in the discriminative learning paradigm in Section IV.

B. Speech Recognition: A Structured Sequence Classification Problem in Machine Learning

Here we address the fundamental problem of ASR. From a functional view, ASR is the conversion process from the acoustic data sequence of speech into a word sequence. From the technical view of ML, this conversion process of ASR requires a number of sub-processes including the use of (discrete) time stamps, often called frames, to characterize the speech waveform data or acoustic features, and the use of categorical labels (e.g. words, phones, etc.) to index the acoustic data sequence. The fundamental issues in ASR lie in the nature of such labels and data. It is important to clearly understand the unique attributes of ASR, in terms of both input data and output labels, as a central motivation to connect the ASR and ML research areas and to appreciate their overlap.

From the output viewpoint, ASR produces sentences that consist of a variable number of words. Thus, at least in principle, the number of possible classes (sentences) for the classification is so large that it is virtually impossible to construct ML models for complete sentences without the use of structure. From the input

viewpoint, the acoustic data are also a sequence with a variable length, and typically, the length of data input is vastly different from that of label output, giving rise to the special problem of segmentation or alignment that the "static" classification problems in ML do not encounter. Combining the input and output viewpoints, we state the fundamental problem as a structured sequence classification task, where a (relatively long) sequence of acoustic data is used to infer a (relatively short) sequence of the linguistic units such as words. More detailed exposition on the structured nature of input and output of the ASR problem can be found in [11], [12].

It is worth noting that the sequence structure (i.e. sentence) in the output of ASR is generally more complex than most of classification problems in ML where the output is a fixed, finite set of categories (e.g., in image classification tasks). Further, when sub-word units and context dependency are introduced to construct structured models for ASR, even greater complexity can arise than the straightforward word sequence output in ASR discussed above.

The more interesting and unique problem in ASR, however, is on the input side, i.e., the variable-length acoustic-feature sequence. The unique characteristic of speech as the acoustic input to ML algorithms makes it a sometimes more difficult object for the study than other (static) patterns such as images. As such, in the typical ML literature, there has typically been less emphasis on speech and related "temporal" patterns than on other signals and patterns.

The unique characteristic of speech lies primarily in its temporal dimension—in particular, in the huge variability of speech associated with the elasticity of this temporal dimension. As a consequence, even if two output word sequences are identical, the input speech data typically have distinct lengths; e.g., different input samples from the same sentence usually contain different data dimensionality depending on how the speech sounds are produced. Further, the discriminative cues among separate speech classes are often distributed over a reasonably long temporal span, which often crosses neighboring speech units. Other special aspects of speech include class-dependent acoustic cues. These cues are often expressed over diverse time spans that would benefit from different lengths of analysis windows in speech analysis and feature extraction. Finally, distinguished from other classification problems commonly studied in ML, the ASR problem is a special class of structured pattern recognition where the recognized patterns (such as phones or words) are embedded in the overall temporal sequence pattern (such as a sentence).

Conventional wisdom posits that speech is a one-dimensional temporal signal in contrast to image and video as higher dimensional signals. This view is simplistic and does not capture the essence and difficulties of the ASR problem. Speech is best viewed as a two-dimensional signal, where the spatial (or frequency or tonotopic) and temporal dimensions have vastly different characteristics, in contrast to images where the two spatial dimensions tend to have similar properties. The "spatial" dimension in speech is associated with the frequency distribution and related transformations, capturing a number of variability types including primarily those arising from environments, speakers, accent, speaking style and rate. The latter type induces correlations between spatial and temporal dimensions, and the environment factors include microphone characteristics, speech transmission channel, ambient noise, and room reverberation.

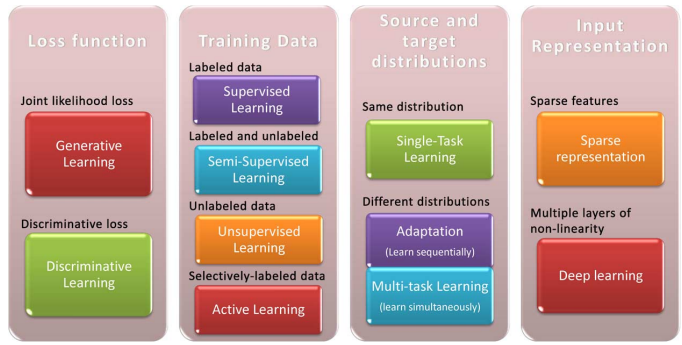


Fig. 1. An overview of ML paradigms and their distinct characteristics.

The temporal dimension in speech, and in particular its correlation with the spatial or frequency-domain properties of speech, constitutes one of the unique challenges for ASR. Some of the advanced generative models associated with the generative learning paradigm of ML as discussed in Section III have aimed to address this challenge, where Bayesian approaches are used to provide temporal constraints as prior knowledge about the human speech generation process.

C. A High-Level Summary of Machine Learning Paradigms

Before delving into the overview detail, here in Fig. 1 we provide a brief summary of the major ML techniques and paradigms to be covered in the remainder of this article. The four columns in Fig. 1 represent the key attributes based on which we organize our overview of a series of ML paradigms. In short, using the nature of the loss function (as well as the decision function), we divide the major ML paradigms into generative and discriminative learning categories. Depending on what kind of training data are available for learning, we alternatively categorize the ML paradigms into supervised, semi-supervised, unsupervised, and active learning classes. When disparity between source and target distributions arises, a more common situation in ASR than many other areas of ML applications, we classify the ML paradigms into single-task, multi-task, and adaptive learning. Finally, using the attribute of input representation, we have sparse learning and deep learning paradigms, both more recent developments in ML and ASR and connected to other ML paradigms in multiple ways.

III. Generative Learning

Generative learning and discriminative learning are the two most prevalent, antagonistically paired ML paradigms developed and deployed in ASR. There are two key factors that distinguish generative learning from discriminative learning: the nature of the model (and hence the decision function) and the loss function (i.e., the core term in the training objective). Briefly speaking, generative learning consists of

- Using a generative model, and
- Adopting a training objective function based on the joint likelihood loss defined on the generative model.

Discriminative learning, on the other hand, requires either

- Using a discriminative model, or

- Applying a discriminative training objective function to a generative model.

In this and the next sections, we will discuss generative vs. discriminative learning from both the model and loss function perspectives. While historically there has been a strong association between a model and the loss function chosen to train the model, there has been no necessary pairing of these two components in the literature [13]. This section will offer a decoupled view of the models and loss functions commonly used in ASR for the purpose of illustrating the intrinsic relationship and contrast between the paradigms of generative vs. discriminative learning. We also show the hybrid learning paradigm constructed using mixed generative and discriminative learning.

This section, starting below, is devoted to the paradigm of generative learning, and the next Section IV to the discriminative learning counterpart.

A. Models

Generative learning requires using a generative model and hence a decision function derived therefrom. Specifically, a generative model is one that describes the joint distribution $p(\mathbf{x}, y; \lambda)$, where λ denotes generative model parameters. In classification, the discriminant functions have the following general form:

$$d_y(\mathbf{x}; \lambda) = \ln p(\mathbf{x}, y; \lambda) = \ln p(\mathbf{x}|y; \lambda) p(y; \lambda) \tag{9}$$

As a result, the output of the decision function in (1) is the class label that produces the highest joint likelihood. Notice that depending on the form of the generative model, the discriminant function and hence the decision function can be greatly simplified. For example, when $p(\mathbf{x}|y; \lambda)$ are Gaussian distributions with the same covariance matrix, $d_y(\mathbf{x}; \lambda)$, for all classes can be replaced by an affine function of $\mathbf{x}$.

One simplest form of generative models is the naïve Bayes classifier, which makes strong independence assumptions that features are independent of each other given the class label. Following this assumption, $p(\mathbf{x}|y; \lambda)$ is decomposed to a product of single-dimension feature distributions $\prod_i p(x_i|y; \lambda)$. The feature distribution at one dimension can be either discrete or continuous, either parametric or non-parametric. In any case, the beauty of the naïve Bayes approach is that the estimation of one feature distribution is completely decoupled from the estimation of others. Some applications have observed benefits by going beyond the naïve Bayes assumption and introducing dependency, partially or completely, among feature variables. One such example is a multivariate Gaussian distribution with a block-diagonal or full convariance matrix.

One can introduce latent variables to model more complex distributions. For example, latent topic models such as probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichilet Allocation (LDA), are widely used as generative models for text inputs. Gaussian mixture models (GMM) are able to approximate any continuous distribution with sufficient precision. More generally, dependencies between latent and observed variables can be represented in a graphical model framework [14].

The notion of graphical models is especially interesting when dealing with structured output. Dynamic Bayesian network is a directed acyclic graph with vertices representing variables and edges representing possible direct dependence relations among the variables. A Bayesian network represents all probability distributions that validly factor according to the network. The joint distribution of all variables in a distribution corresponding to the network factorizes over variables given their parents, i.e. $p(\mathbf{x}_{1:n}) = \prod_{i=1}^{n} p(\mathbf{x}^{(i)}|\text{parents}(\mathbf{x}^{(i)}))$. By having fewer edges in the graph, the network has stronger conditional independence properties and the resulting model has fewer degrees of freedom. When an integer expansion parameter representing discrete time is associated with a Bayesian network, and a set of rules is given to connect together two successive such "chunks" of Bayesian network, then a dynamic Bayesian network arises. For example, hidden Markov models (HMMs), with simple graph structures, are among the most popularly used dynamic Bayesian networks.

Similar to a Bayesian network, a Markov random field (MRF) is a graph that expresses requirements over a family of probability distributions. A MRF, however, is an undirected graph, and thus is capable of representing certain distributions that a Bayesian network can not represent. In this case, the joint distribution of the variables is the product of potential functions over *cliques* (the maximal fully-connected sub-graphs). Formally, $p(\mathbf{x}_{1:n}) = (1/Z) \prod_k \phi_k(\mathbf{x}_{\{k\}})$, where $\phi_k(\mathbf{x}_{\{k\}})$ is the potential function for clique k, and Z is a normalization constant. Again, the graph structure has a strong relation to the model complexity.

B. Loss Functions

As mentioned in the beginning of this section, generative learning requires using a generative model *and* a training objective based on *joint likelihood loss*, which is given by

$$L\left(f(\mathbf{x}), y\right) = -\ln p(\mathbf{x}, y; \lambda) \tag{10}$$

One advantage of using the joint likelihood loss is that the loss function can often be decomposed into independent sub-problems which can be optimized separately. This is especially beneficial when the problem is to predict structured output (such as a sentence output of an ASR system), denoted as bolded $\mathbf{y}$. For example, in a Beysian network, $p(\mathbf{x}, \mathbf{y})$ can be conveniently rewritten as $p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$, where each of $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{y})$ can be further decomposed according to the input and output structure. In the following subsections, we will present several joint likelihood forms widely used in ASR.

The generative model's parameters learned using the above training objective are referred to as *maximum likelihood estimates* (MLE), which is statistically consistent under the assumptions that (a) the generative model structure is correct, (b) the training data is generated from the true distribution, and (c) we have an infinite amount of such training data. In practice, however, the model structure we choose can be wrong and training data is almost never sufficient, making MLE suboptimal for learning tasks. Discriminative loss functions, as will be introduced in Section IV, aim at directly optimizing predicting performance rather than solving a more difficult density estimation problem.

C. Generative Learning in Speech Recognition—An Overview

In ASR, the most common generative learning approach is based on Gaussian-Mixture-Model based Hidden Markov models, or GMM-HMM; e.g., [15]–[18]. A GMM-HMM is

parameterized by $\lambda = (\pi, A, B)$. π is a vector of state prior probabilities; $A = (a_{i,j})$ is a state transition probability matrix; and $B = \{b_1, \ldots, b_n\}$ is a set where b_j represents the Gaussian mixture model of state j. The state is typically associated with a sub-segment of a phone in speech. One important innovation in ASR is the introduction of context-dependent states (e.g. [19]), motivated by the desire to reduce output variability associated with each state, a common strategy for "detailed" generative modeling. A consequence of using context dependency is a vast expansion of the HMM state space, which, fortunately, can be controlled by regularization methods such as state tying. (It turns out that such context dependency also plays a critical role in the more recent advance of ASR in the area of discriminative-based deep learning [20], to be discussed in Section VII-A.)

The introduction of the HMM and the related statistical methods to ASR in mid 1970s [21], [22] can be regarded the most significant paradigm shift in the field, as discussed in [1]. One major reason for this early success was due to the highly efficient MLE method invented about ten years earlier [23]. This MLE method, often called the Baum-Welch algorithm, had been the principal way of training the HMM-based ASR systems until 2002, and is still one major step (among many) in training these systems nowadays. It is interesting to note that the Baum-Welch algorithm serves as one major motivating example for the later development of the more general Expectation-Maximization (EM) algorithm [24].

The goal of MLE is to minimize the empirical risk with respect to the joint likelihood loss (extended to sequential data), i.e.,

$$R_{\text{emp}}(f) = -\sum_i \ln p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}; \pi, A, B) \tag{11}$$

where $\mathbf{x}$ represents acoustic data, usually in the form of a sequence feature vectors extracted at frame-level; $\mathbf{y}$ represents a sequence of linguistic units. In large-vocabulary ASR systems, it is normally the case that word-level labels are provided, while state-level labels are latent. Moreover, in training HMM-based ASR systems, parameter tying is often used as a type of regularization [25]. For example, similar acoustic states of the triphones can share the same Gaussian mixture model. In this case, the $C(f)$ term in (5) is expressed by

$$C(f) = \prod_{(m,n)\in\mathcal{T}} \delta(b_m = b_n) \tag{12}$$

where $\mathcal{T}$ represents a set of tied state pairs.

The use of the generative model of HMMs, including the most popular Gaussian-mixture HMM, for representing the (piecewise stationary) dynamic speech pattern and the use of MLE for training the tied HMM parameters constitute one most prominent and successful example of generative learning in ASR. This success was firmly established by the ASR community, and has been widely spread to the ML and related communities; in fact, HMM has become a standard tool not only in ASR but also in ML and their related fields such as bioinformatics and natural language processing. For many ML as well as ASR researchers, the success of HMM in ASR is a bit surprising due to the well-known weaknesses of the HMM. The remaining part of this section and part of Section VII will aim to address ways of using more advanced ML models and techniques for speech.

Another clear success of the generative learning paradigm in ASR is the use of GMM-HMM as prior "knowledge" within the Bayesian framework for environment-robust ASR. The main idea is as follows. When the speech signal, to be recognized, is mixed with noise or another non-intended speaker, the observation is a combination of the signal of interest and interference of no interest, both unknown. Without prior information, the recovery of the speech of interest and its recognition would be ill defined and subject to gross errors. Exploiting generative models of Gaussian-mixture HMM (also serving the dual purpose of recognizer), or often a simpler Gaussian mixture or even a single Gaussian, as Bayesian prior for "clean" speech overcomes the ill-posed problem. Further, the generative approach allows probabilistic construction of the model for the relationship among the noisy speech observation, clean speech, and interference, which is typically nonlinear when the log-domain features are used. A set of generative learning approaches in ASR following this philosophy are variably called "parallel model combination" [26], vector Taylor series (VTS) method [27], [28], and Algonquin [29]. Notably, the comprehensive application of such a generative learning paradigm for single-channel multitalker speech recognition is reported and reviewed in [5], where the authors apply successfully a number of well established ML methods including loopy belief propagation and structured mean-field approximation. Using this generative learning scheme, ASR accuracy with loud interfering speakers is shown to exceed human performance.

D. Trajectory/Segment Models

Despite some success of GMM-HMMs in ASR, their weaknesses, such as the conditional independence assumption, have been well known for ASR applications [1], [30]. Since early 1990's, ASR researchers have begun the development of statistical models that capture the dynamic properties of speech in the temporal dimension more faithfully than HMM. This class of beyond-HMM models have been variably called stochastic segment model [31], [32], trended or nonstationary-state HMM [33], [34], trajectory segmental model [32], [35], trajectory HMMs [36], [37], stochastic trajectory models [38], hidden dynamic models [39]–[45], buried Markov models [46], structured speech model [47], and hidden trajectory model [48] depending on different "prior knowledge" applied to the temporal structure of speech and on various simplifying assumptions to facilitate the model implementation. Common to all these beyond-HMM models is some temporal trajectory structure built into the models, hence trajectory models. Based on the nature of such structure, we can classify these models into two main categories. In the first category are the models focusing on temporal correlation structure at the "surface" acoustic level. The second category consists of hidden dynamics, where the underlying speech production mechanisms are exploited as the Bayesian prior to represent the "deep" temporal structure that accounts for the observed speech pattern. When the mapping from the hidden dynamic layer to the observation layer limited to linear (and deterministic), then the generative hidden dynamic models in the second category reduces to the first category.

The temporal span of the generative trajectory models in both categories above is controlled by a sequence of linguistic labels, which segment the full sentence into multiple regions from left to right; hence segment models.

In a general form, the trajectory/segment models with hidden dynamics makes use of the switching state space formulation, intensely studied in ML as well as in signal processing and control. They use temporal recursion to define the hidden dynamics, $\mathbf{z}(k)$, which may correspond to articulatory movement during human speech production. Each discrete region or segment, s, of such dynamics is characterized by the s-dependent parameter set $\mathbf{\Lambda}_s$, with the "state noise" denoted by $\mathbf{w}_s(k)$. The memory-less nonlinear mapping function is exploited to link the hidden dynamic vector $\mathbf{z}(k)$ to the observed acoustic feature vector $\mathbf{o}(k)$, with the "observation noise" denoted by $\mathbf{v}_s(k)$, and parameterized also by segment-dependent parameters. The combined "state equation" (13) and "observation equation" (14) below form a general switching nonlinear dynamic system model:

$$\mathbf{z}(k+1) = \mathbf{g}_k\left[\mathbf{z}(k), \mathbf{\Lambda}_s\right] + \mathbf{w}_s(k) \tag{13}$$

$$\mathbf{o}(k') = \mathbf{h}_{k'}\left[\mathbf{z}(k'), \mathbf{\Omega}_{s'}\right] + \mathbf{v}_{s'}(k'). \tag{14}$$

where subscripts k and k' indicate that the functions $\mathbf{g}[.]$ and $\mathbf{h}[.]$ are time varying and may be asynchronous with each other. s or s' denotes the dynamic region correlated with phonetic categories.

There have been several studies on switching nonlinear state space models for ASR, both theoretical [39], [49] and experimental [41]–[43], [50]. The specific forms of the functions of $\mathbf{g}_k[\mathbf{z}(k), \mathbf{\Lambda}_s]$ and $\mathbf{h}_{k'}[\mathbf{z}(k'), \mathbf{\Omega}_{s'}]$ and their parameterization are determined by prior knowledge based on current understanding of the nature of the temporal dimension in speech. In particular, state equation (13) takes into account the temporal elasticity in spontaneous speech and its correlation with the "spatial" properties in hidden speech dynamics such as articulatory positions or vocal tract resonance frequencies; see [45] for a comprehensive review of this body of work.

When nonlinear functions of $\mathbf{g}_k[\mathbf{z}(k), \mathbf{\Lambda}_s]$ and $\mathbf{h}_{k'}[\mathbf{z}(k'), \mathbf{\Omega}_{s'}]$ in (13) and (14) are reduced to linear functions (and when synchrony between the two equations are eliminated), the switching nonlinear dynamic system model is reduced to its linear counterpart, or switching linear dynamic system (SLDS). The SLDS can be viewed as a hybrid of standard HMMs and linear dynamical systems, with a general mathematical description of

$$\mathbf{z}(k+1) = \mathbf{A}_s\mathbf{z}(k) + \mathbf{B}_s\mathbf{w}_s(k) \tag{15}$$

$$\mathbf{o}(k) = \mathbf{C}_s\mathbf{z}(k) + \mathbf{v}_s(k). \tag{16}$$

There has also been an interesting set of work on SLDS applied to ASR. The early set of studies have been carefully reviewed in [32] for generative speech modeling and for its ASR applications. More recently, the studies reported in [51], [52] applied SLDS to noise-robust ASR and explored several approximate inference techniques, overcoming intractability in decoding and parameter learning. The study reported in [53] applied another approximate inference technique, a special type of Gibbs sampling commonly used in ML, to an ASR problem.

During the development of trajectory/segment models for ASR, a number of ML techniques invented originally in non-ASR communities, e.g. variational learning [50], pseudo-Bayesian [43], [51], Kalman filtering [32], extended Kalman filtering [39], [45], Gibbs sampling [53], orthogonal polynomial regression [34], etc., have been usefully applied with modifications and improvement to suit the speech-specific properties and ASR applications. However, the success has mostly been limited to small-scale tasks. We can identify four main sources of difficulty (as well as new opportunities) in successful applications of trajectory/segment models to large-scale ASR. First, scientific knowledge on the precise nature of the underlying articulatory speech dynamics and its deeper articulatory control mechanisms is far from complete. Coupled with the need for efficient computation in training and decoding for ASR applications, such knowledge was forced to be again simplified, reducing the modeling power and precision further. Second, most of the work in this area has been placed within the generative learning setting, having a goal of providing parsimonious accounts (with small parameter sets) for speech variations due to contextual factors and co-articulation. In contrast, the recent joint development of deep learning by both ML and ASR communities, which we will review in Section VII, combines generative and discriminative learning paradigms and makes use of massive instead of parsimonious parameters. There is a huge potential for synergy of research here. Third, although structural ML learning of switching dynamic systems via Bayesian nonparametrics has been maturing and producing successful applications in a number of ML and signal processing tasks (e.g. the tutorial paper [54]), it has not entered mainstream ASR; only isolated studies have been reported on using Bayesian nonparametrics for modeling aspects of speech dynamics [55] and for language modeling [56]. Finally, most of the trajectory/segment models developed by the ASR community have focused on only isolated aspects of speech dynamics rooted in deep human production mechanisms, and have been constructed using relatively simple and largely standard forms of dynamic systems. More comprehensive modeling and learning/inference algorithm development would require the use of more general graphical modeling tools advanced by the ML community. It is this topic that the next subsection is devoted to.

E. Dynamic Graphical Models

The generative trajectory/segment models for speech dynamics just described typically took specialized forms of the more general dynamic graphical model. Overviews on the general use of dynamic Bayesian networks, which belong to directed form of graphical models, for ASR have been provided in [4], [57], [58]. The undirected form of graphical models, including Markov random field and the product of experts model as its special case, has been applied successfully in HMM-based parametric speech synthesis research and systems [59]. However, the use of undirected graphical models has not been as popular and successful. Only quite recently, a restricted form of the Markov random field, called restricted Boltzmann machine (RBM), has been successfully used as one of the several components in the speech model for use in ASR. We will discuss RBM for ASR in Section VII-A.

Although the dynamic graphical networks have provided highly generalized forms of generative models for speech

modeling, some key sequential properties of the speech signal, e.g. those reviewed in Section II-B, have been expressed in specially tailored forms of dynamic speech models, or the trajectory/segment models reviewed in the preceding subsection. Some of these models applied to ASR have been formulated and explored using the dynamic Bayesian network framework [4], [45], [60], [61], but they have focused on only isolated aspects of speech dynamics. Here, we expand the previous use of the dynamic Bayesian network and provide more comprehensive modeling of deep generative mechanisms of human speech.

Shown in Fig. 2 is an example of the directed graphical model or Bayesian network representation of the observable distorted speech feature sequence $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K$ of length K given its "deep" generative causes from both top-down and bottom up directions. The top-down causes represented in Fig. 2 include the phonological/pronunciation model (denoted by sequence $s_1, s_2, \ldots, s_K$), articulatory control model (denoted by sequence $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_K$), articulatory dynamic model (denoted by sequence $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$), and the articultory-to-acoustic mapping model (denoted by the conditional relation from $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$ to $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_K$). The bottom-up causes include nonstationary distortion model, and the interaction model among "hidden" clean speech, observed distorted speech, and the environmental distortion such as channel and noise.

The semantics of the Bayesian network in Fig. 2, which specifies dependency among a set of time varying random variables involved in the full speech production process and its interactions with acoustic environments, is summarized below. First, the probabilistic segmental property of the target process is represented by the conditional probability [62]:

$$p\left[\mathbf{t}(k)|s_k, s_{k-1}, \mathbf{t}(k-1)\right] = \begin{cases} \delta\left[\mathbf{t}(k) - \mathbf{t}(k-1)\right] & if\ s_k = s_{k-1}, \\ \mathcal{N}\left(\mathbf{t}(k); \mathbf{m}(s_k), \mathbf{\Sigma}(s_k)\right) & otherwise. \end{cases} \quad (17)$$

Second, articulatory dynamics controlled by the target process is given by the conditional probability:

$$p_{\mathbf{z}}\left[\mathbf{z}(k+1)|\mathbf{z}(k), \mathbf{t}(k), \mathbf{s}_k\right] = p_{\mathbf{w}}\left[\mathbf{z}(k+1) - \mathbf{\Phi}_{s_k}\mathbf{z}(k) - \left(\mathbf{I} - \mathbf{\Phi}_{s_k}\right)\mathbf{t}(k)\right], \quad (18)$$

or equivalently the target-directed state equation with state-space formulation [63]:

$$\mathbf{z}(k+1) = \mathbf{\Phi}_s\mathbf{z}(k) + (\mathbf{I} - \mathbf{\Phi}_s)\mathbf{t}_s + \mathbf{w}(k). \quad (19)$$

Third, the "observation" equation in the state-space model governing the relationship between distortion-free acoustic features of speech and the corresponding articulatory configuration is represented by

$$\mathbf{o}(k) = \mathbf{h}\left[\mathbf{z}(k)\right] + \mathbf{w}_0(k), \quad (20)$$

where $\mathbf{o}(k)$ is the distortion-free speech vector, $\mathbf{w}_0(k)$ is the observation noise vector uncorrelated with the state noise $\mathbf{w}$, and $\mathbf{h}[.]$ is the static memory-less transformation from the articulatory vector to its corresponding acoustic vector. $\mathbf{h}[.]$ was implemented by a neural network in [63].

Finally, the dependency of the observed environmentally-distorted acoustic features of speech $\mathbf{v}(k)$ on its distortion-free counterpart $\mathbf{o}(k)$, on the non-stationary noise $\mathbf{n}(k)$, and on the stationary channel distortion $\mathbf{h}$ is represented by

$$p_{\mathbf{v}}\left(\mathbf{v}(k)|\mathbf{o}(k), \mathbf{h}, \mathbf{n}(k)\right) = p_{\mathbf{r}}\left[\mathbf{v}(k) - \mathbf{o}(k) + \mathbf{h} + \mathbf{C}\log\left[\mathbf{I} + \exp\left[\mathbf{C}^{-1}\left(\mathbf{n}(k) - \mathbf{o}(k) - \mathbf{h}\right)\right]\right]\right]. \quad (21)$$

where the distribution $p_{\mathbf{r}}$ on the prediction residual has typically taken a Gaussian form with a constant variance [29] or with an SNR-dependent variance [64].

Inference and learning in the comprehensive generative model of speech shown in Fig. 2 are clearly not tractable. Numerous sub-problems and model components associated with the overall model have been explored or solved using inference and learning algorithm developed in ML; e.g. variational learning [50] and other approximate inference methods [5], [45], [53]. Recently proposed new techniques for learning graphical model parameters given all sorts of approximations (in inference, decoding, and graphical model structure) are interesting alternatives to overcoming the intractability problem [65].

Despite the intractable nature of the learning problem in comprehensive graphical modeling of the generative process for human speech, it is our belief that accurate "generative" representation of structured speech dynamics holds a key to the ultimate success of ASR. As will be discussed in Section VII, recent advance of deep learning has reduced ASR errors substantially more than the purely generative graphical modeling approach while making much weaker use of the properties of speech dynamics. Part of that success comes from well designed integration of (unstructured) generative learning with discriminative learning (although more serious but difficult modeling of dynamic processes with temporal memory based on deep recurrent neural networks is a new trend). We devote the next section to discriminative learning, noting a strong future potential of integrating structured generative learning discussed in this section with the increasingly successful deep learning scheme with a hybrid generative-discriminative learning scheme, a subject of Section VII-A.

IV. Discriminative Learning

As discussed earlier, the paradigm of discriminative learning involves either using a discriminative model or applying discriminative training to a generative model. In this section, we first provide a general discussion of the discriminative models and of the discriminative loss functions used in training, followed by an overview of the use of discriminative learning in ASR applications including its successful hybrid with generative learning.

A. Models

Discriminative models make direct use of the conditional relation of labels given input vectors. One major school of such models are referred to as *Bayesian Mininum Risk* (BMR) classifiers [66]–[68]:

$$f(\mathbf{x}; \lambda) = -\arg\min_{y'} \sum_{y} \Delta(y', y) p(y|\mathbf{x}; \lambda) \quad (22)$$

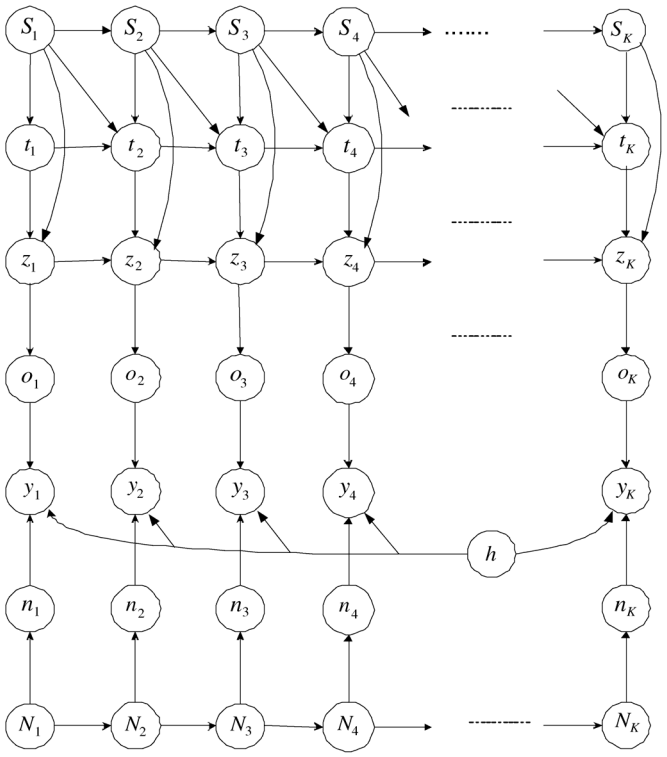


Fig. 2. A directed graphical model, or Bayesian network, which represents the deep generative process of human speech production and its interactions with the distorting acoustic environment; adopted from [45], where the variables y_k represent the "visible" or measurable distorted speech features which are denoted by $\mathbf{v}(k)$ in the text.

where $\Delta(y', y)$ represents the cost of classifying $\mathbf{x}$ as y' while the true classification is y. Δ is sometimes referred to as "loss function", but this loss function is applied at classification time, which should be distinguished from the loss function applied at training time as in (3).

When 0–1 loss is used in classification, (22) is reduced to finding the class label that yields the highest conditional probability, i.e.,

$$f(\mathbf{x}; \lambda) = \arg\min_y p(y|\mathbf{x}; \lambda) \tag{23}$$

The corresponding discriminant function can be represented as

$$d(\mathbf{x}; \lambda) = \ln p(y|\mathbf{x}; \lambda) \tag{24}$$

Conditional log linear models (Chapter 4 in [69]) and multilayer perceptrons (MLPs) with softmax output (Chapter 5 in [69]) are both of this form.

Another major school of discriminative models focus on the decision boundary instead of the probabilistic conditional distribution. In support vector machines (SVMs, see (Chapter 7 in [69])), for example, the discriminant functions (extended to multi-class classification) can be written as

$$d_y(\mathbf{x}; \lambda) = \lambda \cdot \phi(\mathbf{x}, y) \tag{25}$$

where $\phi(\mathbf{x}; y)$ is a feature vector derived from the input and the class label, and is implicitly determined by a reproducing kernel. Notice that for conditional log linear models and MLPs, the discriminant functions in (24) can be equivalently replaced by (25), by ignoring their common denominators.

B. Loss Functions

This section introduces a number of discriminative loss functions. The first group of loss functions are based on probabilistic models, while the second group on the notion of *margin*.

1) Probability-Based Loss: Similar to the joint likelihood loss discussed in the preceding section on generative learning, *conditional likelihood loss* is a probability-based loss function but is defined upon the conditional relation of class labels given input features:

$$L\left(f(\mathbf{x}), y\right) = -\ln p(y|\mathbf{x}; \lambda). \tag{26}$$

This loss function is strongly tied to probabilistic discriminative models such as conditional log linear models and MLPs, while they can be applied to generative models as well, leading to a school of discriminative training methods which will be discussed shortly. Moreover, conditional likelihood loss can be naturally extended to predicting structure output. For example, when applying (26) to Markov random fields, we obtain the training objective of conditional random fields (CRFs) [70]:

$$p(\mathbf{y}|\mathbf{x}; \lambda) = \frac{1}{Z_\lambda(\mathbf{x})} \exp \lambda \cdot f(\mathbf{y}, \mathbf{x}). \tag{27}$$

The partition function $Z_\lambda(\mathbf{x})$ is a normalization factor. λ is a weight vector and $f(\mathbf{y}, \mathbf{x})$ is a vector of feature functions referred to as a feature vector. In ASR tasks where state-level labels are usually unknown, hidden CRF have been introduced to model conditional likelihood with the presence of hidden variables [71], [72]:

$$p(\mathbf{y}|\mathbf{x}; \lambda) = \frac{1}{Z_\lambda(\mathbf{x})} \sum_{\mathbf{z}} \exp \lambda \cdot f(\mathbf{y}, \mathbf{z}, \mathbf{x}). \tag{28}$$

Note that in most of the ML as well as the ASR literature, one often calls the training method using the conditional likelihood loss above as simply maximal likelihood estimation (MLE). Readers should not confuse this type of discriminative learning with the MLE in the generative learning paradigm we discussed in the preceding section.

A generalization of conditional likelihood loss is Minimum Bayes Risk training. This is consistent with the criterion of MBR classifiers described in the previous subsection. The loss function of (MBR) in training is given by

$$L\left(f(\mathbf{x}), \mathbf{y}\right) = -\ln \sum_{\mathbf{y}} \Delta(\mathbf{y}', \mathbf{y}) p(\mathbf{y}|\mathbf{x}; \lambda) \tag{29}$$

where Δ is the cost (loss) function used in classification. This loss function is especially useful in models with structured output; dissimilarity between different outputs $\mathbf{y}$ can be formulated using the cost function, e.g., word or phone error rates in speech recognition [73]–[75], and BLEU score in machine translation [76]–[78]. When Δ is based on 0–1 loss, (29) is reduced to conditional likelihood loss.

2) Margin-Based Loss: Margin-based loss, as discussed and analyzed in detail in [6], represents another class of loss functions. In binary classification, they follow a general expression $L(f(\mathbf{x}), y) = h(yd(\mathbf{x}))$, where $d(\mathbf{x})$ is the discriminant function defined in (2), and $yd(\mathbf{x})$ is known as the margin.

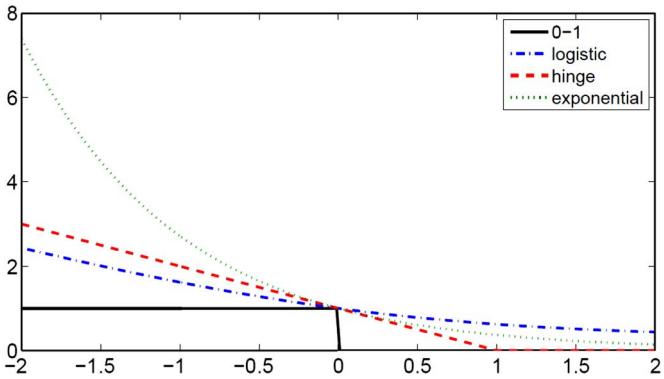


Fig. 3. Convex surrogates of 0–1 loss as discussed and analyzed in [6].

Margin-based loss functions, including *logistic loss*, *hinge loss* used in SVMs, and *exponential loss* used in boosting, are all motivated by upper bounds of 0–1 loss, as illustrated in Fig. 3, with the highly desirable convexity property for ease of optimization. Empirical risk minimization under such loss functions are related to the minimization of classification error rate. In a multi-class setting, the notion of "margin" can be generally viewed as a discrimination metric between the discriminant function of the true class and those of the competing classes, e.g., $d_y(\mathbf{x}) - d_{y'}(\mathbf{x})$, for all $y' \neq y$. Margin-based loss, then, can be defined accordingly such that minimizing the loss would enlarge the "margins" between $d_y(\mathbf{x})$ and $d_{y'}(\mathbf{x})$, $y' \neq y$.

One functional form that fits this intuition is introduced in the minimum classification error (MCE) training [79], [80] commonly used in ASR:

$$L\left(f(\mathbf{x}), y\right) = \sigma\left(-d_y(\mathbf{x};\lambda) + \ln\left[\frac{1}{|\mathcal{Y}|-1}\sum_{y'\neq y}\exp\{d_y(\mathbf{x};\lambda)\eta\}\right]^{\frac{1}{\eta}}\right) \quad (30)$$

where $\sigma(\cdot)$ is a smooth function, which is non-convex and which maps the "margin" to a 0–1 continuum. It is easy to see that in a binary setting where $\mathcal{Y} = \{\pm 1\}$ and where $d_{+1}(\mathbf{x}) = -d_{-1}(\mathbf{x}) = d(\mathbf{x})$, this loss function can be simplified to $\sigma(-d_y(\mathbf{x};\lambda) + d_{y'}(\mathbf{x};\lambda)) = \sigma(-yd(\mathbf{x}))$ which has exactly the same form as logistic loss for binary classification [6].

Similarly, there have been a host of work that generalizes hinge loss to the multi-class setting. One well known approach [81] is to have

$$L\left(f(\mathbf{x}), y\right) = \sum_{y'\neq y} |1 - d_y(\mathbf{x};\lambda) + d_{y'}(\mathbf{x};\lambda)|_+ \quad (31)$$

(where sum is often replaced by max). Again when there are only two classes, (31) is reduced to hinge loss $|1 - yd(\mathbf{x})|_+$.

To be even more general, margin based loss can be extended to structured output as well. In [82], loss functions are defined based on $\Delta(\mathbf{y}, \mathbf{y}')$, where Δ is a measure of discrepancy between two output structures. Analogous to (31), we have

$$L\left(f(\mathbf{x}), \mathbf{y}\right) = \sum_{\mathbf{y}'\neq \mathbf{y}} |\Delta(\mathbf{y}, \mathbf{y}') - d_{\mathbf{y}}(\mathbf{x};\lambda) + d_{\mathbf{y}'}(\mathbf{x};\lambda)|_+ \quad (32)$$

Intuitively, if two output structures are more similar, their discriminant functions should produce more similar output values on the same input data. When Δ is based on 0–1 loss, (32) is reduced to (31).

C. Discriminative Learning in Speech Recognition—An Overview

Having introduced the models and loss functions for the general discriminative learning settings, we now review the use of these models and loss functions in ASR applications.

1) Models: When applied to ASR, there are "direct" approaches which use maximum entropy Markov models (MEMMs) [83], conditional random fields (CRFs) [84], [85], hidden CRFs (HCRFs) [71], augmented CRFs [86], segmental CRFs (SCARFs) [72], and deep-structured CRFs [87], [88]. The use of neural networks in the form of MLP (typically with one hidden layer) with the softmax nonlinear function at the final layer was popular in 1990's. Since the output of the MLP can be interpreted as the conditional probability [89], when the output is fed into an HMM, a good discriminative sequence model, or hybrid MLP-HMM, can be created. The use of this type of discriminative model for ASR has been documented and summarized in detail in [90]–[92] and analyzed recently in [93]. Due mainly to the difficulty in learning MLPs, this line of research has been switched to a new direction where the MLP simply produces a subset of "feature vectors" in combination with the traditional features for use in the generative HMM [94]. Only recently, the difficulty associated with learning MLPs has been actively addressed, which we will discuss in Section VII. All these models are examples of the probabilistic discriminative models expressed in the form of conditional probabilities of speech classes given the acoustic features as the input.

The second school of discriminative models focus on decision boundaries instead of class-conditional probabilities. Analogous to MLP-HMMs, SVM-HMMs have been developed to provide more accurate state/phone classification scores, with interesting results reported [95]–[97]. Recent work has attempted to directly exploit structured SVMs [98], and have obtained significant performance gains in noise-robustness ASR.

2) Conditional Likelihood: The loss functions in discriminative learning for ASR applications have also taken more than one form. The conditional likelihood loss, while being most natural for use in probabilistic discriminative models, can also be applied to generative models. The *maximum mutual information estimation* (MMIE) of generative models, highly popular in ASR, uses an equivalent loss function to the conditional likelihood loss that leads to the empirical risk of

$$R_{\text{emp}}(\lambda) = -\sum_i \ln \frac{p(\mathbf{x}^{(i)}, y^{(i)};\lambda)}{p(\mathbf{x}^{(i)};\lambda)} \quad (33)$$

See a simple proof of their equivalence in [74]. Due to its discriminative nature, MMIE has demonstrated significant performance improvement over using the joint likelihood loss in training Gaussian-mixture HMM systems [99]–[101].

For non-generative or direct models in ASR, the conditional likelihood loss has been naturally used in training. These discriminative probabilistic models including MEMMs [83], CRFs [85], hidden CRFs [71], semi-Markov CRFs [72], and MLP-HMMs [91], all belonging to the class of conditional log linear models. The empirical risk has the same form as (33) except

that $p(\mathbf{y}_i|\mathbf{x}_i;\lambda)$ can be computed directly from the conditional models by

$$R_{\text{emp}}(f) = -\sum_i \ln p(\mathbf{y}_i|\mathbf{x}^{(i)};\lambda) \tag{34}$$

For the conditional log linear models, it is common to apply a Gaussian prior on model parameters, i.e.,

$$C(f) = -\ln p(\lambda) =\propto \frac{\|\lambda\|^2}{2\sigma^2}. \tag{35}$$

3) Bayesian Minimum Risk: Loss functions based on Bayesian minimum risk or BMR (of which the conditional likelihood loss is a special case) have received strong success in ASR, as their optimization objectives are more consistent with ASR performance metrics. Using sentence error, word error and phone error as Δ in (29) leads to their respective methods commonly called Minimum Classification Error (MCE), Minimum Word Error (MWE) and Minimum Phone Error (MPE) in the ASR literature. In practice, due to the non-continuity of these objectives, they are often substituted by continuous approximations, making them closer to margin-based loss in nature.

The MCE loss, as represented by (30) is among the earliest adoption of BMR with margin-based loss form in ASR. It was originated from MCE training of the generative model of Gaussian-mixture HMM [79], [102]. The analogous use of the MPE loss has been developed in [73]. With a slight modification of the original MCE objective function where the bias parameter in the sigmoid smoothing function is annealed over each training iteration, highly desirable discriminative margin is achieved while producing the best ASR accuracy result for a standard ASR task (TI-Digits) in the literature [103], [104].

While the MCE loss function has been developed originally and used pervasively for generative models of HMM in ASR, the same MCE concept can be applied to training discriminative models. As pointed out in [105], the underlying principle of MCE is decision feedback, where the discriminative decision function that is used as the scoring function in the decoding process becomes a part of the optimization procedure of the entire system. Using this principle, a new MCE-based learning algorithm is developed in [106] with success for a speech understanding task which embeds ASR as a sub-component, where the parameters of a log linear model is learned via a generalized MCE criterion. More recently, a similar MCE-based decision-feedback principle is applied to develop a more advanced learning algorithm with success for a speech translation task which also embeds ASR as a sub-component [107].

Most recently, excellent results on large-scale ASR are reported in [108] using the direct BMR (state-level) criterion to train massive sets of ASR model parameters. This is enabled by distributed computing and by a powerful technique called Hessian-free optimization. The ASR system is constructed in a similar framework to the deep neural networks of [20], which we will describe in more detail in Section VII-A.

4) Large Margin: Further, the hinge loss and its variations lead to a variety of large-margin training methods for ASR. Equation (32) represents a unified framework for a number of such large-margin methods. When using a generative model discriminant function $d_y(\mathbf{x};\lambda) = \ln p(\mathbf{x},\mathbf{y};\lambda)$, we have

$$L\left(f(\mathbf{x}),\mathbf{y}\right) = \sum_{\mathbf{y}'\neq\mathbf{y}} \left|\Delta(\mathbf{y},\mathbf{y}') - \ln\frac{p(\mathbf{x},\mathbf{y};\lambda)}{p(\mathbf{x},\mathbf{y};\lambda)}\lambda)\right|_+ \tag{36}$$

Similarly, by using $d_y(\mathbf{x};\lambda) = \ln p(\mathbf{y}|\mathbf{x};\lambda)$, we obtain a large-margin training objective for conditional models:

$$L\left(f(\mathbf{x}),\mathbf{y}\right) = \sum_{\mathbf{y}'\neq\mathbf{y}} \left|\Delta(\mathbf{y},\mathbf{y}') - \ln\frac{p(\mathbf{y}|\mathbf{x};\lambda)}{p(\mathbf{y}|\mathbf{x};\lambda)}\lambda)\right|_+ \tag{37}$$

In [109], a quadratic discriminant function of

$$d_y(\mathbf{x};\Psi) = -\mathbf{z}^T\Psi_y\mathbf{z} \text{ where } \mathbf{z} = [\mathbf{x},1]^T \tag{38}$$

is defined as the decision function for ASR, where Ψ_y, $y\in\mathcal{Y}$, are positive semidefinite matrices that incorporate means and covariance matrices of Gaussians. Note that due to the missing log-variance term in (38), the underlying ASR model is no longer probabilistic and generative. The goal of learning in the approach developed in [109] is to minimize the empirical risk under the hinge loss function in (31), i.e.,

$$R_{\text{emp}}(f) = \sum_i \sum_{y'\neq y} \left|1 + \mathbf{z}^{(i)^T}(\Pi_y - \Psi_{y'})\mathbf{z}^{(i)}\right|_+ \tag{39}$$

while regularizing on model parameters:

$$C(f) = \sum_y \text{trace}(\Psi_y) \tag{40}$$

The minimization of $R_{\text{emp}}(f) + \gamma C(f)$ can be solved as a constrained convex optimization problem, which gives a huge computational advantage over most other discriminative learning algorithms in training ASR which are non-convex in the objective functions. The readers are referred to a recent special issue of IEEE Signal Processing Magazine on the key roles that convex optimization plays in signal processing including speech recognition [110].

A different but related margin-based loss function was explored in the work of [111], [112], where the empirical risk is expressed by

$$R_{\text{emp}}(f) = \min_i \left(d_y(\mathbf{x}_i;\lambda) - \max_{y'\neq y} d_{y'}(\mathbf{x}_i;\lambda)\right), \tag{41}$$

following the standard definition of multiclass separation margin developed in the ML community for probabilistic generative models; e.g., [113], and the discriminant function d in (41) is taken to be the log likelihood function of the input data. Here, the main difference between the two approaches to the use of large margin for discriminative training in ASR is that one is based on the probabilistic generative model of HMM [111], [114], and the other based in non-generative discriminant function [109], [115]. However, similar to [109], [115], the work described in [111], [114], [116], [117] also exploits convexity of the optimization objective by using constraints imposed on model parameters, offering similar kind of compensational advantage. A geometric perspective on large-margin training that analyzes the above two types of loss

functions has appeared recently in [118], which is tested in a vowel classification task.

In order to improve discrimination, many methods have been developed for combining different ASR systems. This is one area with interesting overlaps between the ASR and ML communities. Due to space limitation, we will not cover this ensemble learning paradigm in this paper, except to point out that many common techniques from ML in this area have not made strong impact in ASR and further research is needed.

The above discussions have touched only lightly on discriminative learning for HMM [79], [111], while focusing on the two general aspects of discriminative learning for ASR with respect to modeling and to the use of loss functions. Nevertheless, there has been a very large body of work in the ASR literature, which belongs to the more specific category of the discriminative learning paradigm when the generative model takes the form of GMM-HMM. Recent surveys have provided detailed analysis on and comparisons among the various popular techniques within this specific paradigm pertaining to HMM-like generative models, as well as a unified treatment of these techniques [74], [114], [119], [120]. We now turn to a brief overview on this body of work.

D. Discriminative Learning for HMM and Related Generative Models

The overview article of [74] provides the definitions and intuitions of four popular discriminative learning criteria in use for HMM-based ASR, all being originally developed and steadily modified and improved by ASR researchers since mid-1980's. They include: 1) MMI [101], [121]; 2) MCE, which can be interpreted as minimal sentence error rate [79] or approximate minimal phone error rate [122]; 3) MPE or minimal phone error [73], [123]; and 4) MWE or minimal word error. A discriminative learning objective function is the empirical average of the related loss function over all training samples.

The essence of the work presented in [74] is to reformulate all the four discriminative learning criteria for an HMM into a common, unified mathematical form of rational functions. This is trivial for MMI by the definition, but non-trivial for MCE, MPE, and MWE. The critical difference between MMI and MCE/MPE/MWE is the product form vs. the summation form in the respective loss function, while the form of rational function requires the product form and requires a non-trivial conversion for the MCE/MPE/MWE criteria in order to arrive at a unified mathematical expression with MMI. The tremendous advantage gained by the unification is the enabling of a natural application of the powerful and efficient optimization technique, called growth-transformation or extended Baum-Welch algorithm, to optimization all parameters in parametric generative models. One important step in developing the growth-transformation algorithm is to derive two key auxiliary functions for intermediate levels of optimization. Technical details including major steps in the derivation of the estimation formulas are provided for growth-transformation based parameter optimization for both the discrete HMM and the Gaussian HMM. Full technical details including the HMM with the output distributions using the more general exponential family, the use of lattices in computing the needed quantities in the estimation formulas, and the supporting experimental results in ASR are provided in [119].

The overview article of [114] provides an alternative unified view of various discriminative learning criteria for an HMM. The unified criteria include 1) MMI; 2) MCE; and 3) LME (large-margin estimate). Note the LME is the same as (41) when the discriminant function d takes the form of log likelihood function of the input data in an HMM. The unification proceeds by first defining a "margin" as the difference between the HMM log likelihood on the data for the correct class minus the geometric average the HMM log likelihoods on the data for all incorrect classes. This quantity can be intuitively viewed as a measure of distance from the data to the current decision boundary, and hence "margin". Then, given the fixed margin function definition, three different functions of the same margin function over the training data samples give rise to 1) MMI as a sum of the margins over the data; 2) MCE as sum of exponential functions of the margin over the data; and 3) LME as a minimum of the margins over the data.

Both the motivation and the mathematical form of the unified discriminative learning criteria presented in [114] are quite different from those presented in [74], [119]. There is no common rational functional form to enable the use of the extended Baum-Welch algorithm. Instead, the interesting constrained optimization technique was developed by the authors and presented. The technique consists of two steps: 1) Approximation step, where the unified objective function is approximated by an auxiliary function in the neighborhood of the current model parameters; and 2) Maximization step, where the approximated auxiliary function was optimized using the locality constraint. Importantly, a relaxation method was exploited, which was also used in [117] with an alternative approach, to further approximate the auxiliary function into a form of positive semi-definite matrix. Thus, the efficient convex optimization technique for a semi-definite programming problem can be developed for this M-step.

The work described in [124] also presents a unified formula for the objective function of discriminative learning for MMI, MP/MWE, and MCE. Similar to [114], both contain a generic nonlinear function, with its varied forms corresponding to different objective functions. Again, the most important distinction between the product vs. summation forms of the objective functions was not explicitly addressed.

One interesting area of ASR research on discriminative learning for HMM has been to extend the learning of HMM parameters to the learning of parametric feature extractors. In this way, one can achieve end-to-end optimization for the full ASR system instead of just the model component. One earliest work in this area was from [125], where dimensionality reduction in the Mel-warped discrete Fourier transform (DFT) feature space was investigated subject to maximal preservation of speech classification information. An optimal linear transformation on the Mel-warped DFT was sought, jointly with the HMM parameters, using the MCE criterion for optimization. This approach was later extended to use filter-bank parameters, also jointly with the HMM parameters, with similar success [126]. In [127], an auditory-based feature extractor was parameterized by a set of weights in the auditory filters, and had its output fed into an HMM speech recognizer. The MCE-based discriminative learning procedure was applied to both filter parameters and HMM parameters, yielding superior performance over the separate training of auditory filter parameters and HMM

parameters. The end-to-end approach to speech understanding described in [106] and to speech translation described in [107] can be regarded as extensions of the earlier set of work discussed here on "joint discriminative feature extraction and model training" developed for ASR applications.

In addition to the many uses of discriminative learning for HMM as a generative model, for other more general forms of generative models for speech that are surveyed in Section III, discriminative learning has been applied with success in ASR. The early work in the area can be found in [128], where MCE is used to discriminatively learn all the polynomial coefficients in the trajectory model discussed in Section III. The extension from the generative learning for the same model as described in [34] to the discriminative learning (via MCE, e.g.) is motivated by the new model space for smoothness-constrained, state-bound speech trajectories. Discriminative learning offers the potential to re-structure the new, constrained model space and hence to provide stronger power to disambiguate the observational trajectories generated from nonstationary sources corresponding to different speech classes. In more recent work of [129] on the trajectory model, the time variation of the speech data is modeled as a semi-parametric function of the observation sequence via a set of centroids in the acoustic space. The model parameters of this model are learned discriminatively using the MPE criterion.

E. Hybrid Generative-Discriminative Learning Paradigm

Toward the end of discussing generative and discriminative learning paradigms, here we would like to provide a brief overview on the hybrid paradigm between the two. Discriminative classifiers directly relate to classification boundaries, do not rely on assumptions on the data distribution, and tend to be simpler for the design. On the other hand, generative classifiers are most robust to the use of unlabeled data, have more principled ways of treating missing information and variable-length data, and are more amenable to model diagnosis and error analysis. They are also coherent, flexible, and modular, and make it relatively easy to embed knowledge and structure about the data. The modularity property is a particularly key advantage of generative models: due to local normalization properties, different knowledge sources can be used to train different parts of the model (e.g., web data can train a language model independent of how much acoustic data there is to train an acoustic model). See [130] for a comprehensive review of how speech production knowledge is embedded into design and improvement of ASR systems.

The strengths of both generative and discriminative learning paradigms can be combined for complementary benefits. In the ML literature, there are several approaches aimed at this goal. The work of [131] makes use of the Fisher kernel to exploit generative models in discriminative classifiers. Structured discriminability as developed in the graphical modeling framework also belongs to the hybrid paradigm [57], where the structure of the model is formed to be inherently discriminative so that even a generative loss function yields good classification performance. Other approaches within the hybrid paradigm use the loss functions that blend the joint likelihood with the conditional likelihood by linearly interpolating them [132] or by conditional modeling with a subset of the observation data. The hybrid paradigm can also be implemented by staging generative learning ahead of discriminative learning. A prime example of this hybrid style is the use of a generative model to produce features that are fed to the discriminative learning module [133], [134] in the framework of deep belief network, which we will return to in Section VII. Finally, we note that with appropriate parameterization some classes of generative and discriminative models can be made mathematically equivalent [135].

V. Semi-Supervised and Active Learning

The preceding overview of generative and discriminative ML paradigms uses the attributes of loss and decision functions to organize a multitude of ML techniques. In this section, we use a different set of attributes, namely the nature of the training data in relation to their class labels. Depending on the way that training samples are labeled or otherwise, we can classify many existing ML techniques into several separate paradigms, most of which have been in use in the ASR practice. Supervised learning assumes that *all* training samples are labeled, while unsupervised learning assumes *none*. Semi-supervised learning, as the name suggests, assumes that both labeled and unlabeled training samples are available. Supervised, unsupervised and semi-supervised learning are typically referred to under the *passive learning* setting, where labeled training samples are generated randomly according to an unknown probability distribution. In contrast, *active learning* is a setting where the learner can intelligently choose which samples to label, which we will discuss at the end of this section. In this section, we concentrate mainly on semi-supervised and active learning paradigms. This is because supervised learning is reasonably well understood and unsupervised learning does not directly aim at predicting outputs from inputs (and hence is beyond the focus of this article); We will cover these two topics only briefly.

A. Supervised Learning

In supervised learning, the training set consists of pairs of inputs and outputs drawn from a joint distribution. Using notations introduced in Section II-A,

- $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})|(\mathbf{x}^{(i)}, y^{(i)}) \sim p(\mathbf{x}, y)\}_{i=1}^{m}$

The learning objective is again empirical risk minimization with regularization, i.e., $R_{\text{emp}}(f) + \gamma C(\lambda)$, where both input data $\mathbf{x}^{(i)}$ and the corresponding output labels $y^{(i)}$ are provided. In Sections III and IV, we provided an overview of the generative and discriminative approaches and their uses in ASR all under the setting of supervised learning.

Notice that there may exist multiple levels of label variables, notably in ASR. In this case, we should distinguish between the *fully supervised* case, where labels of all levels are known, the *partially supervised* case, where labels at certain levels are missing. In ASR, for example, it is often the case that the training set consists of waveforms and their corresponding word-level transcriptions as the labels, while the phone-level transcriptions and time alignment information between the waveforms and the corresponding phones are missing.

Therefore, strictly speaking, what is often called supervised learning in ASR is actually partially supervised learning. It is due to this "partial" supervision that ASR often uses EM algorithm [24], [136], [137]. For example, in the Gaussian mixture model for speech, we may have a label variable y representing

the Gaussian mixture ID and z representing the Gaussian component ID. In the latter case, our goal is to maximize the incomplete likelihood

$$\ln \sum_{z} p(\mathbf{x}, y, z; \lambda) \tag{42}$$

which cannot be optimized directly. However, we can apply EM algorithm that iteratively maximizes its lower bound. The optimization objective at each iteration, then, is given by

$$\sum_{i} \ln \mathrm{E}_{p\left(z^{(i)}|\mathbf{x}^{(i)}, y^{(i)}; \lambda^{g}\right)} \left[p\left(\mathbf{x}^{(i)}, y^{(i)}, z^{(i)}; \lambda\right) \right] \tag{43}$$

B. Unsupervised Learning

In ML, unsupervised learning in general refers to learning with the input data only. This learning paradigm often aims at building representations of the input that can be used for prediction, decision making or classification, and data compression. For example, density estimation, clustering, principle component analysis and independent component analysis are all important forms of unsupervised learning. Use of vector quantization (VQ) to provide discrete inputs to ASR is one early successful application of unsupervised learning to ASR [138].

More recently, unsupervised learning has been developed as a component of staged hybrid generative-discriminative paradigm in ML. This emerging technique, based on the deep learning framework, is beginning to make impact on ASR, which we will discuss in Section VII. Learning sparse speech representations, to be discussed in Section VII also, can also be regarded as unsupervised feature learning, or learning feature representations in absence of classification labels.

C. Semi-Supervised Learning—An Overview

The semi-supervised learning paradigm is of special significance in both theory and applications. In many ML applications including ASR, unlabeled data is abundant but labeling is expensive and time-consuming. It is possible and often helpful to leverage information from unlabeled data to influence learning. Semi-supervised learning is targeted at precisely this type of scenario, and it assumes the availability of both labeled $\mathcal{D}$ and unlabeled $\mathcal{U}$ data, i.e.,

- $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}) | (\mathbf{x}^{(i)}, y^{(i)}) \sim p(\mathbf{x}, y)\}_{i=1}^{m}$
- $\mathcal{U} = \{\mathbf{x}^{(i)} | \mathbf{x}^{(i)} \sim p(\mathbf{x})\}\}_{i=m+1}^{m+n}$

The goal is to leverage both data sources to improve learning performance.

There have been a large number of semi-supervised learning algorithms proposed in the literature and various ways of grouping these approaches. An excellent survey can be found in [139]. Here we categorize semi-supervised learning methods based on their *inductive* or *transductive* nature. The key difference between inductive and transductive learning is the outcome of learning. In the former setting, the goal is to find a decision function that not only correctly classifies training set samples, but also generalizes to *any* future sample. In contrast, transductive learning aims at directly predicting the output labels of a test set, without the need of generalizing to other samples. In this regard, the direct outcome of transductive semi-supervised learning is a set of labels instead of a decision function. All learning paradigms we have presented in Sections III and IV are inductive in nature.

An important characteristic of transductive learning is that both training and test data are explicitly leveraged in learning. For example, in transductive SVMs [7], [140], test-set outputs are estimated such that the resulting hyper-plane separates *both* training and test data with maximum margin. Although transductive SVMs implicitly use a decision function (hyper-plane), the goal is no longer to generalize to future samples but to predict as accurately as possible the outputs of the test set. Alternatively, transductive learning can be conducted using graph-based methods that utilize the similarity matrix of the input [141], [142]. It is worth noting that transductive learning is often mistakenly equated to semi-supervised learning, as both learning paradigms receive partially labeled data for training. In fact, semi-supervised learning can be either inductive or transductive, depending on the outcome of learning. Of course, many transductive algorithms can produce models that can be used in the same fashion as would the outcome of an inductive learner. For example, graph-based transductive semi-supervised learning can produce a non-parametric model that can be used to classify any new point, not in the training and "test" set, by finding where in the graph any new point might lie, and then interpolating the outputs.

1) Inductive Approaches: Inductive approaches to semi-supervised learning require the construction of classification models f. A general semi-supervised learning objective can be expressed as

$$R_{\mathrm{emp}}(f) + \alpha R_{\mathcal{U}}(f) + \gamma C(\lambda) \tag{44}$$

where $R_{\mathrm{emp}}(f)$ again is the empirical risk on labeled data $\mathcal{D}$, $R_{\mathcal{U}}(f)$ is a "risk" measured on unlabeled data $\mathcal{U}$.

For generative models (Section III), a common measure on unlabeled data is the incomplete-data likelihood, i.e.,

$$R_{\mathcal{U}}(f) = -\sum_{i=m+1}^{m+n} \ln p(\mathbf{x}^{(i)}; \lambda) \tag{45}$$

The goal of semi-supervised learning, therefore, becomes to maximize the complete-data likelihood on $\mathcal{D}$ and the incomplete-data likelihood on $\mathcal{U}$. One way of solving this optimization problem is applying the EM algorithm or its variations to unlabeled data [143], [144]. Furthermore, when discriminative loss functions, e.g., (26), (29), or (32), are used in $R_{\mathrm{emp}}(f)$, the learning objective becomes equivalent to applying discriminative training on $\mathcal{D}$ and while applying maximum likelihood estimation on $\mathcal{U}$.

The above approaches, however, are not applicable to discriminative models (which model conditional relations rather than joint distributions). For conditional models, one solution to semi-supervised learning is *minimum entropy regularization* [145], [146] that defines $R_{\mathcal{U}}(f)$ as the conditional entropy of unlabeled data:

$$R_{\mathcal{U}}(f) = H(y|\mathbf{x}; \lambda) \tag{46}$$

The semi-supervised learning objective is then to maximize the conditional likelihood of $\mathcal{D}$ while minimizing the conditional

entropy of $\mathcal{U}$. This approach generally would result in "sharper" models which can be data-sensitive in practice.

Another set of results makes an additional assumption that prior knowledge can be utilized in learning. *Generalized expectation criteria* [147] represent prior knowledge as labeled features,

$$R_{\mathcal{U}}(f) = D(\hat{p} \| \tilde{p}_\lambda) \tag{47}$$

In the last term, $\hat{p}$ and $\tilde{p}_\lambda$ both refer to conditional distributions of labels given a feature. While the former is specified by prior knowledge, and the latter is estimated by applying model λ on unlabeled data. In [148], prior knowledge is encoded as *virtual evidence* [149], denoted as v. They model the distribution $p(y, v|\mathbf{x})$ explicitly and formulate the semi-supervised learning problem as follows,

$$R_{\mathcal{U}}(f) = \sum_{i=m+1}^{m+n} \log p_\lambda(\mathbf{v}^{(i)}|\mathbf{x}^{(i)}; \lambda) \tag{48}$$

where $p_\lambda(\mathbf{v}^{(i)}|\mathbf{x}^{(i)}\ \lambda)$ can be optimized in an EM fashion. This type of methods has been most used in sequence models, where prior knowledge on frame- or segment-level features/labels is available. This can be potentially interesting to ASR as a way of incorporating linguistic knowledge into data-driven systems.

The concept of semi-supervised SVMs ($S^3\mathrm{VM}$) was originally inspired by transductive SVMs [7]. The intuition is to find a labeling of $\mathcal{U}$ such that the SVM trained on $\mathcal{D}$ and newly labeled $\mathcal{U}$ would have the largest margin. In a binary classification setting, the learning objective is given by a $R_{\mathrm{emp}}(f)$ based on hinge loss and

$$R_{\mathcal{U}}(f) = \sum_{i=m+1}^{m+n} |1 - |d(\mathbf{x}_i; \lambda)\|_+ \tag{49}$$

where $d(\mathbf{x}_i; \lambda)$ represents a linear function; $|d(\mathbf{x}_i; \lambda)|$ is derived from $(\mathrm{sgn}\ d(\mathbf{x}_i; \lambda)) \cdot d(\mathbf{x}_i; \lambda)$. Various works have been proposed to approximate the optimization problem (which is no longer convex due to the second term), e.g., [140], [150]–[152]. In fact, a transductive SVM is in the strict sense an inductive learner, although it is by convention called "transductive" for its intention to minimize the generalization error bound on the target inputs.

While the methods introduced above are model-dependent, there are inductive algorithms that can be applied across different models. *Self-training* [153] extends the idea of EM to a wider range of classification models—the algorithm iteratively trains a seed classifier using the labeled data, and uses predictions on the unlabeled data to expand the training set. Typically the most confident predictions are added to the training set. The EM algorithm on generative models can be considered a special case of self-training in that all unlabeled samples are used in re-training, weighted by their posterior probabilities. The disadvantage of self-training is that it lacks a theoretical justification for optimality and convergence, unless certain conditions are satisfied [153].

Co-training [154] assumes that the input features can be split into two conditionally independent subsets, and that each subset is sufficient for classification. Under these assumptions, the algorithm trains two separate classifiers on these two subsets of features, and each classifier's predictions on new unlabeled samples are used to enlarge the training set of the other. Similar to self-training, co-training often selects data based on confidence. Certain work has found it beneficial to probabilistically label $\mathcal{U}$, leading to the *co-EM* paradigm [155]. Some variations of co-training include split data and ensemble learning.

2) Transductive Approaches: Transductive approaches do not necessarily require a classification model. Instead, the goal is to produce a set of labels $\{y_i\}_{i=m}^{m+n}$ for $\mathcal{U}$. Such approaches are often based on graphs, with nodes representing labeled and unlabeled samples and edges representing the similarity between the samples. Let W denote an $m+n$ by $m+n$ similarity matrix, F denote an $m+n$ by $|\mathcal{Y}|$ matrix representing classification scores of all $\mathbf{x}_i$ with respect to all classes, and Y denote another $m+n$ by $|\mathcal{Y}|$ matrix representing known label information. The goal of graph-based learning is to find a classification of all data that satisfies the constraints imposed by the labeled data and is smooth over the entire graph. This can be expressed by a general objective function of

$$\min_F L(F, Y) + \gamma C(F, W) \tag{50}$$

which consists of a loss term and regularization term. The loss term evaluates the discrepancy between classification outputs and known labels while the regularization term ensures that similar inputs have similar outputs. Different graph-based algorithms, including mincut [156], random walk [157], label propagation [158], local and global consistency [159] and manifold regularization [160], and measure propagation [161] vary only in the forms of the loss and regularization functions.

Notice that compared to inductive approaches to semi-supervised learning, transductive learning has rarely been used in ASR. This is mainly because of the usually very large amount of data involved in training ASR systems, which makes it prohibitive to directly use affinity between data samples in learning. The methods we will review shortly below all fit into the inductive category. We believe, however, it is important to introduce readers to some powerful transductive learning techniques and concepts which have made fundamental impact to machine learning. They also have the potential for make impact in ASR as example- or template-based approaches have increasingly been explored in ASR more recently. Some of the recent work of this type will be discussed in Section VII-B.

D. Semi-Supervised Learning in Speech Recognition

We first point out that the standard description of semi-supervised learning discussed above in the ML literature has been used loosely in the ASR literature, and often been referred to as unsupervised learning or unsupervised training. This (minor) confusion is caused by the fact that while there are both transcribed/labeled and un-transcribed sets of training data, the latter is significantly greater in the amount than the former. Technically, the need for semi-supervised learning in ASR is obvious. State of the art performance in large vocabulary ASR systems usually requires thousands of hours of manually annotated speech and millions of words of text. The manual transcription is often too expensive or impractical. Fortunately, we can rely upon the assumption that any domain which requires ASR technology will have thousands of hours of audio

available. Unsupervised acoustics model training builds initial models from small amounts of transcribed acoustic data and then use them to decode much larger amounts of un-transcribed data. One then trains new models using part or all of these automatic transcripts as the label. This drastically reduces the labeling requirements for ASR in the sparse domains.

The above training paradigm falls into the self-training category of semi-supervised learning described in the preceding subsection. Representative work includes [162]–[164], where an ASR trained on a small transcribed set is used to generate transcriptions for larger quantities of un-transcribed data first. The recognized transcriptions are selected then based on confidence measures. The selected transcriptions are treated as the correct ones and are used to train the final recognizer. Specific techniques include incremental training where the high-confidence (as determined with a threshold) utterances are combined with transcribed utterances to retrain or to adapt the recognizer. Then the retrained recognizer is used to transcribe the next batch of utterances. Often, generalized expectation maximization is used where all utterances are used but with different weights determined by the confidence measure. This approach fits into the general framework of (44), and has also been applied to combining discriminative training with semi-supervised learning [165]. While straightforward, it has been shown that such confidence-based self-training approaches are associated with the weakness of reinforcing what the current model already knows and sometimes even reinforcing the errors. Divergence is frequently observed when the performance of the current model is relatively poor.

Similar to the objective of (46), in the work of [166] the global entropy defined over the entire training data set is used as the basis for assigning labels in the un-transcribed portion of the training utterances for semi-supervised learning. This approach differs from the previous ones by making the decision based on the global dataset instead of individual utterances only. More specifically, the developed algorithm focuses on the improvement to the overall system performance by taking into consideration not only the confidence of each utterance but also the frequency of similar and contradictory patterns in the un-transcribed set when determining the right utterance-transcription pair to be included in the semi-supervised training set. The algorithm estimates the expected entropy reduction which the utterance-transcription pair may cause on the full un-transcribed dataset.

Other ASR work [167] in semi-supervised learning leverages prior knowledge, e.g., closed-captions, which are considered as low-quality or noisy labels, as constraints in otherwise standard self-training. The idea is akin to (48). One particular constraint exploited is to align the closed captions with recognized transcriptions and to select only segments that agree. This approach is called lightly supervised training in [167]. Alternatively, recognition has been carried out by using a language model which is trained on the closed captions.

We would like to point out that many effective semi-supervised learning algorithms developed in ML as surveyed in Section V-D have yet to be explored in ASR, and this is one area expecting growing contributions from the ML community.

E. Active Learning—An overview

Active learning is a similar setting to semi-supervised learning in that, in addition to a small amount of labeled data $\mathcal{D}$, there is a large amount of unlabeled data $\mathcal{U}$ available; i.e.,

- $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})|(\mathbf{x}^{(i)}, y^{(i)}) \sim p(\mathbf{x}, y)\}_{i=1}^{m}$
- $\mathcal{U} = \{\mathbf{x}^{(i)}|\mathbf{x}^{(i)} \sim p(\mathbf{x})\}\}_{i=m+1}^{m+n}$

The goal of active learning, however, is to query the most informative set of inputs to be labeled, hoping to improve classification performance with the minimum number of queries. That is, in active learning, the learner may play an *active* role in deciding the data set $\mathcal{D}$ rather than it be passively given.

The key idea behind active learning is that a ML algorithm can achieve greater performance, e.g., higher classification accuracy, with fewer training labels if it is allowed to choose the subset of data that has labels. An active learner may pose queries, usually in the form of unlabeled data instances to be labeled (often by a human). For this reason, it is sometimes called *query learning*. Active learning is well-motivated in many modern ML problems, where unlabeled data may be abundant or easily obtained, but labels are difficult, time-consuming, or expensive to obtain. This is the situation for speech recognition. Broadly, active learning comes in two forms: batch active learning, where a subset of data is chosen, *a priori* in a batch to be labeled. The labels of the instances in the batch chosen to be labeled may not, under this approach, influence other instances to be selected since all instances are chosen at once. In *online active learning*, on the other hand, instances are chosen one-by-one, and the true labels of all previously labeled instances may be used to select other instances to be labeled. For this reason, online active learning is sometimes considered more powerful.

A recent survey of active learning can be found in [168]. Below we briefly review a few commonly used approaches with relevance to ASR.

1) Uncertainty Sampling: Uncertainty sampling is probably the simplest approach to active learning. In this framework, unlabeled inputs are selected based on an uncertainty (informativeness) measure,

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{U}} \phi(\mathbf{x}; \lambda); \tag{51}$$

where λ denote model parameters estimated on $\mathcal{D}$. There are various choices of the certainty measure [169]–[171], including

- *posterior*: $\phi(\mathbf{x}; \lambda) = 1 - p(\hat{y}|\mathbf{x}; \lambda)$ where $\hat{y} = \arg\max_y p(y|\mathbf{x}; \lambda)$;
- *margin*: $\phi(\mathbf{x}; \lambda) = -(p(\hat{y}_1|\mathbf{x}; \lambda) - p(\hat{y}_2|\mathbf{x}; \lambda))$, where $\hat{y}_1$ and $\hat{y}_2$ are the first and second most likely label under model λ; and
- *entropy*: $\phi(\mathbf{x}; \lambda) = H(y|\mathbf{x}; \lambda)$

For non-probabilistic models, similar measures can be constructed from discriminant functions. For example, the distance to the decision boundary is used as a measure for active learning associated with SVM [172].

2) Query-by-Committee: The query-by committee algorithm enjoys a more theoretical explanation [173], [174]. The idea is to construct a committee of learners, denoted by

$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$, all trained on labeled samples. The unlabeled samples upon which the committee disagree the most are selected to be labeled by human, i.e.,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{U}} \phi(\mathbf{x}; \Lambda) \tag{52}$$

The key problems in committee-based methods consist of (1) constructing a committee Λ that represents competing hypotheses and (2) having a measure of disagreement ϕ. The first problem is often tackled by sampling the model space, by splitting the training data or by splitting the feature space. For the second problem, one popularly used disagreement measure is *vote entropy* [175] $\phi(\mathbf{x}; \Lambda) = H(V(y)/C)$ where $V(y)$ is the number of votes the class y receives from the committee regarding input $\mathbf{x}$ and C is the committee size.

3) Exploiting Structures in Data: Both uncertainty sampling and query-by committee may encounter the *sampling bias* problem; i.e., the selected inputs are not representatives of the true input distribution. Recent work proposed to select inputs not only based on an uncertainty/disagreement measure but also on a "density" measure [171], [176]. Mathematically, the decision is

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{U}} \phi(\mathbf{x}) s(\mathbf{x}); \tag{53}$$

where $\phi(\mathbf{x})$ can be either $\phi(\mathbf{x}; \lambda)$ in uncertainty sampling of $\phi(\mathbf{x}; \Lambda)$ in query-by-committee; $s(\mathbf{x})$ is a density term that can be estimated by computing similarity with other inputs with or without clustering. Such methods have achieved active learning performance superior to those that do not take structure or density into consideration.

4) Submodular Active Selection: A recent and novel approach to batch active learning for speech recognition was proposed in [177] that made use of sub-modular functions; in this work, results outperformed many of the active learning methods mentioned above. Sub-modular functions are a rich class of functions on discrete sets and subsets thereof that capture the notion of diminishing returns—an item is worth less as the context in which it is evaluated gets larger. Sub-modular functions are relevant for batch active learning either in speech recognition and other areas of machine learning [178], [179].

5) Comparisons Between Semi-Supervised and Active Learning: Active learning and semi-supervised learning both aim at making the most out of unlabeled data. As a result, there are conceptual overlaps between these two paradigms of ML. As an example, in self-training of semi-supervised technique as discussed earlier, the classifier is first trained with a small amount of labeled data, and then used to classify the unlabeled data. Typically the most confident unlabeled instances, together with their predicted labels, are added to the training set, and the process repeats. A corresponding technique in active learning is uncertainty sampling, where the instances about which the model is least confident are selected for querying. As another example, co-training in semi-supervised learning initially trains separate models with the labeled data. The models then classify the unlabeled data, and "teach" the other models with a few unlabeled examples about which they are most confident. This corresponds to the query-by-committee approach in active learning.

This analysis shows that active learning and semi-supervised learning attack the same problem from opposite directions. While semi-supervised methods exploit what the learner thinks it knows about the unlabeled data, active methods attempt to explore the unknown aspects.

F. Active Learning in Speech Recognition

The main motivation for exploiting active learning paradigm in ASR to improve the systems performance in the applications where the initial accuracy is very low and only a small amount of data can be transcribed. A typical example is the voice search application, with which users may search for information such as phone numbers of a business with voice. In the ASR component of a voice search system, the vocabulary size is usually very large, and the users often interact with the system using free-style instantaneous speech under real noisy environments. Importantly, acquisition of un-transcribed acoustic data for voice systems is usually as inexpensive as logging the user interactions with the system, while acquiring transcribed or labeled acoustic data is very costly. Hence, active learning is of special importance for ASR here. In light of the recent popularity of and availability of infrastructure for crowding sourcing, which has the potential to stimulate a paradigm shift in active learning, the importance of active learning in ASR applications in the future is expected to grow.

As described above, the basic approach of active learning is to actively ask a question based on all the information available so far, so that some objective function can be optimized when the answer becomes known. In many ASR related tasks, such as designing dialog systems and improving acoustic models, the question to be asked is limited to selecting an utterance for transcribing from a set of un-transcribed utterances.

There have been many studies on how to select appropriate utterance for human transcription in ASR. The key issue here is the criteria for selecting utterances. First, confidence measures is used as the criterion as in the standard uncertainty sampling method discussed earlier [180]–[182]. The initial recognizer in these approaches, which is prepared beforehand, is first used to recognize all the utterances in the training set. Those utterances that have recognition results with less confidence are then selected. The word posterior probabilities for each utterance have often been used as confidence measures. Second, in the query-by-committee based approach proposed in [183], samples that cause the largest different opinions from a set of recognizers (committee) are selected. These multiple recognizers are also prepared beforehand, and the recognition results produced by these recognizers are used for selecting utterances. The authors apply the query-by-committee technique not only to acoustic models but also to language models and their combination. Further, in [184], the confusion or entropy reduction based approach is developed where samples that reduce the entropy about the true model parameters are selected for transcribing. Similarly, in the error rate-based approach the samples that can minimize the expected error rate most is selected.

A rather unique technique of active learning for ASR is developed in [166]. It recognizes the weakness of the most commonly used, confidence-based approach as follows. Frequently,

the confidence-based active learning algorithm is prone to select noise and garbage utterances since these utterances typically have low confidence scores. Unfortunately, transcribing these utterances is usually difficult and carries little value in improving the overall ASR performance. This limitation originates from the utterance-by-utterance decision, which is based on the information from each individual utterance only. that is, transcribing the least confident utterance may significantly help recognize that utterance but it may not help improve the recognition accuracy on other utterances. Consider two speech utterances A and B. Say A has a slightly lower confidence score than B. If A is observed only once and B occurs frequently in the dataset, a reasonable choice is to transcribe B instead of A. This is because transcribing B would correct a larger fraction of errors in the test data than transcribing A and thus has better potential to improve the performance of the whole system. This example shows that the active learning algorithm should select the utterances that can provide the most benefit to the full dataset. Such a global criterion for active learning has been implemented in [166] based on maximizing the expected lattice entropy reduction over all un-transcribed data. Optimizing the entropy is shown to be more robust than optimizing the top choice [184], since it considers all possible outcomes weighted with probabilities.

VI. Transfer Learning

The ML paradigms and algorithms discussed so far in this paper have the goal of producing a classifier that generalizes across samples drawn from the same distribution. Transfer learning, or learning with "knowledge transfer", is a new ML paradigm that emphasizes producing a classifier that generalizes across distributions, domains, or tasks. Transfer learning is gaining growing importance in ML in recent years but is in general less familiar to the ASR community than other learning paradigms discussed so far. Indeed, numerous highly successful adaptation techniques developed in ASR are aimed to solve one of the most prominent problems that transfer learning researchers in ML try to address—mismatch between training and test conditions. However, the scope of transfer learning in ML is wider than this, and it also encompasses a number of schemes familiar to ASR researchers such as audio-visual ASR, multi-lingual and cross-lingual ASR, pronunciation learning for word recognition, and detection-based ASR. We organize such diverse ASR methodologies into a unified categorization scheme under the very broad transfer learning paradigm in this section, which would otherwise be viewed as isolated ASR applications. We also use the standard ML notations in Section II to describe all ASR topics in this section.

There is vast ML literature on transfer learning. To organize our presentation with considerations to existing ASR applications, we create the four-way categorization of major transfer learning techniques, as shown in Table II, using the following two axes. The first axis is the manner in which knowledge is transferred. **Adaptive** learning is one form of transfer learning in which knowledge transfer is done in a *sequential* manner, typically from a source task to a target task. In contrast, **multi-task** learning is concerned with learning multiple tasks *simultaneously*.

TABLE II
Four-Way Categorization of Transfer Learning

	$\mathcal{X}^S = \mathcal{X}^T$ and $\mathcal{Y}^S = \mathcal{Y}^T$	$\mathcal{X}^S \neq \mathcal{X}^T$ and/or $\mathcal{Y}^S \neq \mathcal{Y}^T$
Adaptive	homogeneous transfer	heterogeneous transfer
Multi-task	homogeneous multi-task	heterogeneous multi-task

Transfer learning can be orthogonally categorized using the second axis as to whether the input/output space of the target task is different from that of the source task. It is called **homogeneous** if the source and target task have the same input/output space, and is **heterogeneous** otherwise. Note that both adaptive learning and multi-task learning can be either homogeneous or heterogeneous.

A. Homogeneous Transfer

Interestingly, homogeneous transfer, i.e., adaptation, is one paradigm of transfer learning that has been more extensively developed (and also earlier) in the speech community rather than the ML community. To be consistent with earlier sections, we first present adaptive learning from the ML theoretical perspective, and then discuss how it is applied to ASR.

1) Basics: At this point, it is helpful for the readers to review the notations set up in Section II which will be used intensively in this section. In this setting, the input space $\mathcal{X}$ in the target task is the same as that in the source task, so is the output space $\mathcal{Y}$. Most of the ML techniques discussed earlier in this article assume that the source-task (training) and target-task (test) samples are generated from the same underlying distribution $p(\mathbf{x}, y)$ over $\mathcal{X} \times \mathcal{Y}$. Often, however, in most ASR applications classifier f^S is trained on samples drawn from a source distribution $p^S(\mathbf{x}, y)$ that is different from, yet similar to, the target distribution $p^T(\mathbf{x}, y)$. Moreover, while there may be a large amount of training data from the source task, only a limited amount of data (labeled and/or unlabeled) from the target task is available. The problem of adaptation, then, is to learn a new classifier f^T leveraging the available information from the source and target tasks, ideally to minimize $R_{p^T}(f)$.

Homogeneous adaptation is important to many machine learning applications. In ASR, a source model (e.g., speaker-independent HMM for ASR) may be trained on a dataset consisting of samples from a large number of individuals, but the target distribution would correspond only to a specific user. In image classification, the lighting condition at application time may vary from that when training-set images are collected. In spam detection, the wording styles of spam emails or web pages are constantly evolving.

Homogeneous adaptation can be formulated in various ways depending on the type of source/target information available at adaptation time. Information from the source task may consist of the following:

- $\mathcal{D}^S = \{(\mathbf{x}^{(i)}, y^{(i)}) | (\mathbf{x}^{(i)}, y^{(i)}) \sim p^S(\mathbf{x}, y)\}$, i.e., labeled training data from the source task. A typical example of $\mathcal{D}^S$ in ASR is the transcribed speech data for training speaker-independent and environment-independent HMMs.
- f^S: a source model or classifier which is either an accurate representation or an approximately correct estimate of $\inf_{f \in \mathcal{F}} R_{p^S}(f)$, i.e., the risk minimizer for the source

task. A typical example of f^S in ASR is the HMM trained already using speaker-independent and environment-independent training data.

For the target task, one or both of the following data sources may be available:

- $\mathcal{D}^T = \{(\mathbf{x}^{(i)}, y^{(i)}) | (\mathbf{x}^{(i)}, y^{(i)}) \sim p^T(\mathbf{x}, y)\}$, i.e., labeled adaptation data from the target task. A typical example of $\mathcal{D}^T$ in ASR is the enrollment data for speech dictation systems.
- $\mathcal{U}^T = \{\mathbf{x}^{(i)} | \mathbf{x}^{(i)} \sim p^T(\mathbf{x})\}$, i.e., unlabeled adaptation data from the target task. A typical example of $\mathcal{U}^T$ in ASR is the actual conversation speech from the users of interactive voice response systems.

Below we present and analyze two major classes of methods for homogeneous adaptation.

2) Data Combination: When $\mathcal{D}^S$ is available at adaptation time, a natural approach is to seek intelligent ways of combining $\mathcal{D}^S$ and $\mathcal{D}^T$ (and sometimes $\mathcal{U}^T$). The work by [185] derived generalization error bounds for a learner that minimizes a convex combination of source and target empirical risks,

$$J(f) = \alpha R^S_{\text{emp}}(f) + (1-\alpha) R^T_{\text{emp}}(f) + \gamma C(f) \tag{54}$$

where R^S_{emp} and R^T_{emp} are defined with respect to $\mathcal{D}^S$ and $\mathcal{D}^T$ respectively. Data combination is also implicitly used in many practical studies on SVM adaptation. In [116], [186], [187], the support vectors as derived data from f^S are combined with $\mathcal{D}^S$, with different weights, for retraining a target model.

In many applications, however, it is not always feasible to use $\mathcal{D}^S$ in adaptation. In ASR, for example, $\mathcal{D}^S$ may consist of hundreds or even thousands of hours of speech, making any data combination approach prohibitive.

3) Model Adaptation: Here we focus on alternative classes of approaches which attempt to adapt directly from f^S. These approaches can be less optimal (due to the loss of information) but much more efficient compared with data combination. Depending on which target-data source is used, adaptation of f^S can be conducted in a supervised or unsupervised fashion. Unsupervised adaptation is akin to the semi-supervised learning setting already discussed in Section V-C, which we do not repeat here.

In supervised adaptation, labeled data $\mathcal{D}^T$, usually in a very small amount, is used to adapt f^S. The learning objective consists of minimizing the target empirical risk while regularizing toward the source model,

$$J(f) = R^T_{\text{emp}}(f) + \gamma C(f; f^S) \tag{55}$$

Different adaptation techniques essentially differ in how regularization works.

One school of methods are based on Bayesian model selection. In other words, regularization is achieved by a prior distribution on model parameters, i.e.,

$$C(f) = -\ln p(f; f^S) \tag{56}$$

where the hyper-parameters of the prior distribution are usually derived from source model parameters. The function form of the prior distribution depends on classification model. For generative models, it is mathematically convenient to use the conjugate prior of the likelihood function such that the posterior belongs to the same function family as the prior. For example, normal-Wishart priors have been used in adapting Gaussians [188], [189]; Dirichlet priors have been used in adapting multinomial [188]–[190]. For discriminative models such as conditional maximum entropy models, SVMs and MLPs, Gaussian priors are commonly used [116], [191]. A unified view of these priors can be found in [116], which also relates the generalization error bound to the KL divergence of source and target sample distributions.

Another group of methods adapt model parameters in a more structured way by forcing the target model to be a transformation of the source model. The regularization term can be expressed as follows,

$$C(f) = \delta\left(f = \phi(f^S)\right) \tag{57}$$

where ϕ represents a transform function. For example, maximum likelihood linear regression (MLLR) [192], [193] adapts Gaussian parameters through shared transform functions. In [194], [195], the target MLP is obtained by augmenting the source MLP with an additional linear input layer.

Finally, other studies on model adaptation have related the source and target models via shared components. Both [196] and [197] proposed to construct MLPs whose input-to-hidden layer is shared by multiple related tasks. This layer represents an "internal representation" which, once learned, is fixed during adaptation. In [198], the source and target distributions were each assumed to a mixture of two components, with one mixture component shared between source and target tasks. [199], [200] assumed that the target distribution is a mixture of multiple source distributions. They proposed to combine source models weighted by source distributions, which has an expected loss guarantee with respect to any mixture.

B. Homogeneous Transfer in Speech Recognition

The ASR community is actually among the first to systematically investigate homogeneous adaptation, mostly in the context of speaker or noise adaptation. A recent survey on noise adaptation techniques for ASR can be found in [201].

One of the commonly used homogeneous adaptation techniques in ASR is *maximum a posteriori* (MAP) method [188], [189], [202], which places adaptation within the Bayesian learning framework and involves using a prior distribution on the model parameters as in (56). Specifically, to adapt Gaussian mixture models, MAP method applies a normal-Wishart prior on Gaussian means and covariance matrices, and a Dirichlet prior on mixture component weights.

Maximum likelihood linear regression (MLLR) [192], [193] regularizes the model space in a more structured way than MAP in many cases. MLLR adapts Gaussian mixture parameters in HMMs through shared affine transforms such that each HMM state is more likely to generate the adaptation data and hence the target distribution. There are various techniques to combine the structural information captured by linear regression with the prior knowledge utilized in the Bayesian learning framework.

Maximum a posteriori linear regression (MAPLR) and its variations [203], [204] improve over MLLR by assuming a prior distribution on affine transforms.

Yet another important family of adaptation techniques have been developed, unique in ASR and not seen in the ML literature, in the frameworks of *speaker adaptive training* (SAT) [205] and *noise adaptive training* (NAT) [201], [206], [207]. These frameworks utilize speaker or acoustic-environment adaptation techniques, such as MLLR [192], [193], SPLICE [206], [208], [209], and vector Taylor series approximation [210], [211], during training to explicitly address speaker-induced or environment-induced variations. Since speaker and acoustic-environment variability has been explicitly accounted for by the transformations in training, the resulting speaker-independent and environment-independent models only need to address intrinsic phonetic variability and are hence more compact than conventional models.

There are a few extensions to the SAT and NAT frameworks based on the notion of "speaker clusters" or "environment clusters" [212], [213]. For example, [213] proposed *cluster adaptive training* where all Gaussian components in the system are partitioned into Gaussian classes, and all training speakers are partitioned into speaker clusters. It is assumed that a speaker-dependent model (either in adaptive training or in recognition) is a linear combination of cluster-conditional models, and that all Gaussian components in the same Gaussian class share the same set of weights. In a similar spirit, *eigenvoice* [214] constrains a speaker-dependent model to be a linear combination of a number of basis models. During recognition, a new speaker's super-vector is a linear combination of eigen-voices where the weights are estimated to maximize the likelihood of the adaptation data.

C. Heterogeneous Transfer

1) Basics: Heterogeneous transfer involves a higher level of generalization. The goal is to transfer knowledge learned from one task to a new task of a different nature. For example, an image classification task may benefit from a text classification task although they do not have the same input spaces. Speech recognition of a low-resource language can borrow information from a resource-rich language ASR system, despite the difference in their output spaces (i.e., different languages).

Formally, we define the input spaces $\mathcal{X}^S$ and $\mathcal{X}^T$ for the source and target tasks, respectively. Similarly, we define the corresponding output spaces as $\mathcal{Y}^S$ and $\mathcal{Y}^T$, respectively. While homogeneous adaptation assumes that $\mathcal{X}^S = \mathcal{X}^T$ *and* $\mathcal{Y}^S = \mathcal{Y}^T$, heterogeneous adaptation assumes that either $\mathcal{X}^S \neq \mathcal{X}^T$, or $\mathcal{Y}^S \neq \mathcal{Y}^T$, or both spaces are different. Let p^S denote the joint distribution over $\mathcal{X}^S \times \mathcal{Y}^S$, and Let p^T denote the joint distribution over $\mathcal{X}^T \times \mathcal{Y}^T$. The goal of heterogeneous adaptation is then to minimize $R_{p^T}(f)$ leveraging two data sources: (1) source task information in the form of $\mathcal{D}^S$ and/or f^S; (2) target task information in the form of $\mathcal{D}^T$ and/or $\mathcal{U}^T$.

Below we discuss the methods associated with two main conditions under which heterogeneous adaptation is typically applied.

2) $\mathcal{X}^S \neq \mathcal{X}^T$ and $\mathcal{Y}^S = \mathcal{Y}^T$: In this case, we often leverage the relationship between $\mathcal{X}^S$ and $\mathcal{X}^T$ for knowledge transfer. The basic idea is to map $\mathcal{X}^S$ and $\mathcal{X}^T$ to the same space where homogeneous adaptation can be applied. The mapping can be done directly from $\mathcal{X}^S$ to $\mathcal{X}^T$, i.e.,

$$\mathcal{X}^S \rightarrow \mathcal{X}^T \tag{58}$$

For example, a bilingual dictionary represents such a mapping that can be used in cross-language text categorization or retrieval [139], [215], where two languages are considered as two different domains or tasks.

Alternatively, both $\mathcal{X}^S$ to $\mathcal{X}^T$ can be transformed to a common latent space [216], [217]:

$$\phi_1 : \mathcal{X}^S \rightarrow \mathcal{L}, \phi_2 : \mathcal{X}^T \rightarrow \mathcal{L} \tag{59}$$

The mapping can also be modeled probabilistically in the form of a "translation" model [218],

$$p(x_2|x_1) \quad x_1 \in \mathcal{X}^S, x_2 \in \mathcal{X}^T \tag{60}$$

The above relationships can be estimated if we have a large number of correspondence data $\{(x_1^{(i)} \in \mathcal{X}^S, x_2^{(i)} \in \mathcal{X}^T)\}$. For example, the study of [218] uses images with text annotations as aligned input pairs to estimate $p(x_2|x_1)$. When correspondence data is not available, the study of [217] learns the mappings to the latent space that preserve the local geometry and neighborhood relationship.

3) $\mathcal{X}^S = \mathcal{X}^T$ and $\mathcal{Y}^S \neq \mathcal{Y}^T$: In this scenario, it is the relationship between the output spaces that methods of heterogeneous adaptation will leverage. Often, there may exist direct mappings between output spaces. For example, phone recognition (source task) has an output space consisting of phoneme sequences. Word recognition (target task), then, can be cast into a phone recognition problem followed by a phoneme-to-word transducer:

$$\mathcal{Y}^S \rightarrow \mathcal{Y}^T \tag{61}$$

Alternatively, the output spaces $\mathcal{Y}^S$ and $\mathcal{Y}^T$ can also be made related to each other via a latent space:

$$\phi_1 : \mathcal{L} \rightarrow \mathcal{Y}^S, \phi_2 : \mathcal{L} \rightarrow \mathcal{Y}^T \tag{62}$$

For example, $\mathcal{Y}^S$ and $\mathcal{Y}^T$ can be both transformed from a hidden layer space using MLPs [196]. Additionally, the relationship can be modeled in the form of constraints. In [219], the source task is part-of-speech tagging and the target task is named-entity recognition. By imposing constraints on the output variables, e.g., named entities should not be part of verb phrases, the author showed both theoretically and experimentally that it is possible to learn f^T with fewer samples from $\mathcal{D}^T$.

D. Multi-Task Learning

Finally, we briefly discuss the multi-task learning setting. While adaptive learning just described aims at transferring knowledge sequentially from a source task to a target task, multi-task learning focuses on learning different yet related tasks simultaneously. Let's index the individual tasks in the

multi-task learning setting by $k = 1, 2, \ldots, K$. We denote the input and output spaces of task k by $\mathcal{X}^k$ and $\mathcal{Y}^k$, respectively, and denote the joint input/output distribution for task k by $p^k(\mathbf{x}^k, y^k)$. Note that the tasks are homogeneous if the input/output spaces are the same across tasks, i.e., $\mathcal{X}^k = \mathcal{X}$ and $\mathcal{Y}^k = \mathcal{Y}$ for any k; and are otherwise heterogeneous. Multi-task learning described in ML literature is usually heterogeneous in nature. Furthermore, we assume a training set $\mathcal{D}^k$ is available for each task k with samples drawn from the corresponding joint distribution. The tasks relate to each other via a meta-parameter θ, the form of which will be discussed shortly. The goal of multi-task learning is to jointly find a meta-parameter θ and a set of decision functions $\mathbf{f} = (f^1, f^2, \ldots, f^K)$ that minimize the average expected risk, i.e.,

$$\min_{\theta, \mathbf{f}} \frac{1}{K} \sum_k R_{p^k}(f^k) \tag{63}$$

It has been theoretically proved that learning multiple tasks jointly is guaranteed to have better generalization performance than learning them independently, given that these tasks are related [197], [220]–[223]. A common approach is to minimize the empirical risk of each task while applying regularization that captures the relatedness between tasks, i.e.,

$$\min_{\theta, \mathbf{f}} \frac{1}{K} \sum_k R^k_{\text{emp}}(f^k) + \gamma C(\mathbf{f}; \theta) \tag{64}$$

where $R^k_{\text{emp}}(f^k)$ denotes the empirical risk on data set $\mathcal{D}^k$, and $C(\mathbf{f}; \theta)$ is a regularization term that is parameterized by θ.

As in the case of adaptation, regularization is the key to the success of multi-task learning. There have been many regularization strategies that exploit different types of relatedness. A large body of work is based on hierarchical Bayesian inference [220], [224]–[228]. The basic idea is to assume that (1) f_k are each generated from a prior $p(f; \theta_k)$; and (2) θ_k are each generated from the same hyper prior $p(\theta; \theta_0)$. Another approach, and probably one of the earliest to multi-task learning, is to let the decision functions of different tasks share common structures. For example, in [196], [197], some layers of MLPs are shared by all tasks while the remaining layers are task-dependent. With a similar motivation, other works apply various forms of regularization such that f_k of similar tasks are close to each other in the model parameter space [223], [229], [230].

Recently, multi-task learning, and transfer learning in general, has been approached by the ML community using a new, deep learning framework. The basic idea is that the feature representations learned in an unsupervised manner at higher layers in the hierarchical architectures tend to share the properties common among different tasks; e.g., [231]. We will briefly discuss an application of this new approach to multi-task learning to ASR next, and will devote the final section of this article to a more general introduction of deep learning.

E. Heterogeneous Transfer and Multi-Task Learning in Speech Recognition

The terms heterogeneous transfer and multi-task learning are often used exchangeably in the ML literature, as multi-task learning usually involves heterogeneous inputs or outputs, and the information transfer can go both directions between tasks.

One most interesting application of heterogeneous transfer and multi-task learning is multimodal speech recognition and synthesis, as well as recognition and synthesis of other sources of modality information such as video and image. In the recent study of [231], an instance of heterogeneous multi-task learning architecture of [196] is developed using more advanced hierarchical architectures and deep learning techniques. This deep learning model is then applied to a number of tasks including speech recognition, where the audio data of speech (in the form of spectrogram) and video data are fused to learn the shared representation of both speech and video in the mid layers of a deep architecture. This multi-task deep architecture extends the earlier deep architectures developed for single-task deep learning architecture for image pixels [133], [134] and for speech spectrograms [232] alone. The preliminary results reported in [231] show that both video and speech recognition tasks are improved with multi-task learning based on the deep architectures enabling shared speech and video representations.

Another successful example of heterogeneous transfer and multi-task learning in ASR is multi-lingual or cross-lingual speech recognition, where speech recognition for different languages is considered as different tasks. Various approaches have been taken to attack this rather challenging acoustic modeling problem for ASR, where the difficulty lies in low resources in either data or transcriptions or both due to economic considerations in developing ASR for all languages of the world. Cross-language data sharing and data weighing are common and useful approaches [233]. Another successful approach is to map pronunciation units across languages either via knowledge-based or data-driven methods [234].

Finally, when we consider phone recognition and word recognition as different tasks, e.g., phone recognition results are used not for producing text outputs but for language-type identification or for spoken document retrieval, then the use of pronunciation dictionary in almost all ASR systems to bridge phones to words can constitute another excellent example of heterogeneous transfer. More advanced frameworks in ASR have pushed this direction further by advocating the use of even finer units of speech than phones to bridge the raw acoustic information of speech to semantic content of speech via a hierarchy of linguistic structure. These atomic speech units include "speech attributes" [235], [236] in the detection-based and knowledge-rich modeling framework, and overlapping articulatory features in the framework that enables the exploitation of articulatory constraints and speech co-articulatory mechanisms for fluent speech recognition; e.g., [130], [237], [238]. When the articulatory information during speech can be recovered during speech recognition using articulatory based recognizers, such information can be usefully applied to a different task of pronunciation training.

VII. Emerging Machine Learning Paradigms

In this final section, we will provide an overview on two emerging and rather significant developments within both ASR

and ML communities in recent years: learning with deep architectures and learning with sparse representations. These developments share the commonality that they focus on learning input representations of the signals including speech, as shown in the last column of Fig. 1. Deep learning is intrinsically linked to the use of multiple layers of nonlinear transformations to derive speech features, while learning with sparsity involves the use of examplar-based representations for speech features which have high dimensionality but mostly empty entries.

Connections between the emerging learning paradigms reviewed in this section and those discussed in previous sections can be drawn. Deep learning described in Section VII-A below is an excellent example of hybrid generative and discriminative learning paradigms elaborated in Sections III and IV, where generative learning is used as "pre-training" and discriminative learning is used as "fine tuning". Since the "pre-training" phase typically does not make use of labels for classification, this also falls into the unsupervised learning paradigm discussed in Section V-B. Sparse representation in Section VII-B below is also linked to unsupervised learning; i.e. learning feature representations in absence of classification labels. It further relates to regularization in supervised or semi-supervised learning.

A. Learning Deep Architectures

Learning deep architectures, or more commonly called deep learning or hierarchical learning, has emerged since 2006 ignited by the publications of [133], [134]. It links and expands a number of ML paradigms that we have reviewed so far in this paper, including generative, discriminative, supervised, unsupervised, and multi-task learning. Within the past few years, the techniques developed from deep learning research have already been impacting a wide range of signal and information processing including notably ASR; e.g., [20], [108], [239]–[256].

Deep learning refers to a class of ML techniques, where many layers of information processing stages in hierarchical architectures are exploited for unsupervised feature learning and for pattern classification. It is in the intersections among the research areas of neural network, graphical modeling, optimization, pattern recognition, and signal processing. Two important reasons for the popularity of deep learning today are the significantly lowered cost of computing hardware and the drastically increased chip processing abilities (e.g., GPU units). Since 2006, researchers have demonstrated the success of deep learning in diverse applications of computer vision, phonetic recognition, voice search, spontaneous speech recognition, speech and image feature coding, semantic utterance classification, hand-writing recognition, audio processing, information retrieval, and robotics.

1) A Brief Historical Account: Until recently, most ML techniques had exploited shallow-structured architectures. These architectures typically contain a single layer of nonlinear feature transformations and they lack multiple layers of adaptive non-linear features. Examples of the shallow architectures are conventional HMMs which we discussed in Section III, linear or nonlinear dynamical systems, conditional random fields, maximum entropy models, support vector machines, logistic regression, kernel regression, and multi-layer perceptron with a single hidden layer. A property common to these shallow learning models is the simple architecture that consists of only one layer responsible for transforming the raw input signals or features into a problem-specific feature space, which may be unobservable. Take the example of a SVM. It is a shallow linear separation model with one or zero feature transformation layer when kernel trick is and is not used, respectively. Shallow architectures have been shown effective in solving many simple or well-constrained problems, but their limited modeling and representational power can cause difficulties when dealing with more complicated real-world applications involving natural signals such as human speech, natural sound and language, and natural image and visual scenes.

Historically, the concept of deep learning was originated from artificial neural network research. It was not until recently that the well known optimization difficulty associated with the deep models was empirically alleviated when a reasonably efficient, unsupervised learning algorithm was introduced in [133], [134]. A class of deep generative models was introduced, called deep belief networks (DBNs, not to be confused with Dynamic Bayesian Networks discussed in Section III). A core component of the DBN is a greedy, layer-by-layer learning algorithm which optimizes DBN weights at time complexity linear to the size and depth of the networks. The building block of the DBN is the restricted Boltzmann machine, a special type of Markov random field, discussed in Section III-A, that has one layer of stochastic hidden units and one layer of stochastic observable units.

The DBN training procedure is not the only one that makes deep learning possible. Since the publication of the seminal work in [133], [134], a number of other researchers have been improving and developing alternative deep learning techniques with success. For example, one can alternatively pre-train the deep networks layer by layer by considering each pair of layers as a de-noising auto-encoder [257].

2) A Review of Deep Architectures and Their Learning: A brief overview is provided here on the various architectures of deep learning, including and beyond the original DBN. As described earlier, deep learning refers to a rather wide class of ML techniques and architectures, with the hallmark of using many layers of non-linear information processing stages that are hierarchical in nature. Depending on how the architectures and techniques are intended for use, e.g., synthesis/generation or recognition/classification, one can categorize most of the work in this area into three types summarized below.

The first type consists of generative deep architectures, which are intended to characterize the high-order correlation properties of the data or joint statistical distributions of the visible data and their associated classes. Use of Bayes rule can turn this type of architecture into a discriminative one. Examples of this type are various forms of deep auto-encoders, deep Boltzmann machine, sum-product networks, the original form of DBN and its extension to the factored higher-order Boltzmann machine in its bottom layer. Various forms of generative models of hidden speech dynamics discussed in Section III-D and III-E, the deep dynamic Bayesian network model discussed in Fig. 2, also belong to this type of generative deep architectures.

The second type of deep architectures are discriminative in nature, which are intended to provide discriminative power for pattern classification and to do so by characterizing the posterior distributions of class labels conditioned on the visible data.

Examples include deep-structured CRF, tandem-MLP architecture [94], [258], deep convex or stacking network [248] and its tensor version [242], [243], [259], and detection-based ASR architecture [235], [236], [260].

In the third type, or hybrid deep architectures, the goal is discrimination but this is assisted (often in a significant way) with the outcomes of generative architectures. In the existing hybrid architectures published in the literature, the generative component is mostly exploited to help with discrimination as the final goal of the hybrid architecture. How and why generative modeling can help with discriminative can be examined from two viewpoints: 1)The optimization viewpoint where generative models can provide excellent initialization points in highly nonlinear parameter estimation problems (The commonly used term of "pre-training" in deep learning has been introduced for this reason); and/or 2) The regularization perspective where generative models can effectively control the complexity of the overall model. When the generative deep architecture of DBN is subject to further discriminative training, commonly called "fine-tuning" in the literature, we obtain an equivalent architecture of deep neural network (DNN, which is sometimes also called DBN or deep MLP in the literature). In a DNN, the weights of the network are "pre-trained" from DBN instead of the usual random initialization. The surprising success of this hybrid generative-discriminative deep architecture in the form of DNN in large vocabulary ASR was first reported in [20], [250], soon verified by a series of new and bigger ASR tasks carried out vigorously by a number of major ASR labs worldwide.

Another typical example of the hybrid deep architecture was developed in [261]. This is a hybrid of DNN with a shallow discriminative architecture of CRF. Here, the overall architecture of DNN-CRF is learned using the discriminative criterion of sentence-level conditional probability of labels given the input data sequence. It can be shown that such DNN-CRF is equivalent to a hybrid deep architecture of DNN and HMM, whose parameters are learned jointly using the full-sequence maximum mutual information (MMI) between the entire label sequence and the input data sequence. This architecture is more recently extended to have sequential connections or temporal dependency in the hidden layers of DBN, in addition to the output layer [244].

3) Analysis and Perspectives: As analyzed in Section III, modeling structured speech dynamics and capitalizing on the essential temporal properties of speech are key to high accuracy ASR. Yet the DBN-DNN approach, while achieving dramatic error reduction, has made little use of such structured dynamics. Instead, it simply accepts the input of a long window of speech features as its acoustic context and outputs a very large number of context-dependent sub-phone units, using many hidden layers one on top of another with massive weights.

The deficiency in temporal aspects of the DBN-DNN approach has been recognized and much of current research has focused on recurrent neural network using the same massive-weight methodology. It is not clear such a brute-force approach can adequately capture the underlying structured dynamic properties of speech, but it is clearly superior to the earlier use of long, fixed-sized windows in DBN-DNN. How to integrate the power of generative modeling of speech dynamics, elaborated in Section III-D and Section III-E, into the discriminative deep architectures explored vigorously by both ML and ASR communities in recent years is a fruitful research direction.

Active research is currently ongoing by a growing number of groups, both academic and industrial, in applying deep learning to ASR. New and more effective deep architectures and related learning algorithms have been reported in every major ASR-related and ML-related conferences and workshops since 2010. This trend is expected to continue in coming years.

B. Sparse Representations

1) A Review of Recent Work: In recent years, another active area of ASR research that is closely related to ML has been the use of sparse representation. This refers to a set of techniques used to reconstruct a structured signal from a limited number of training examples, a problem which arises in many ML applications where reconstruction relates to adaptively finding a dictionary which best represents the signal on a per-sample basis. The dictionary can either include random projections, as is typically done for signal reconstruction, or include actual training samples from the data, as explored also in many ML applications. Like deep learning, sparse representation is another emerging and rapidly growing area with contributions in a variety of signal processing and ML conferences, including ASR in recent years.

We review the recent applications of sparse representation to ASR here, highlighting the relevance to and contributions from ML. In [262], [263], exemplar-based sparse representations are systematically explored to map test features into the linear span of training examples. They share the same "non-parametric" ML principle as the nearest-neighbor approach explored in [264] and the SVM method in directly utilizing information about individual training examples. Specifically, given a set of acoustic-feature sequences from the training set that serve as a dictionary, the test data is represented as a linear combination of these training examples by solving a least square regression problem constrained by sparseness on the weight solution. The use of such constraints is typical of regularization techniques, which are fundamental in ML and discussed in Section II. The sparse features derived from the sparse weights and dictionary are then used to map the test samples back into the linear span of training examples in the dictionary. The results show that the frame-level speech classification accuracy using sparse representations exceeds that of Gaussian mixture model. In addition, sparse representations not only move test features closer to training, they also move the features closer to the correct class. Such sparse representations are used as additional features to the existing high-quality features and error rate reduction is reported in both phone recognition and large vocabulary continuous speech recognition tasks with detailed experimental conditions provided in [263].

In the studies of [265], [266], various uncertainty measures are developed to characterize the expected accuracy of a sparse imputation, an exemplar-based reconstruction method based on representing segments of the noisy speech signal as linear combinations of as few clean speech example segments as possible. The exemplars used are time-frequency patches of real speech, each spanning multiple time frames. Then after the distorted speech is modeled as a linear combination of noise and speech

exemplars, an algorithm is developed and applied to recover the sparse linear combination of exemplars from the observed noisy speech. In experiments on noisy large vocabulary speech data, the use of observation uncertainties and sparse representations improves ASR performance significantly.

In a further study reported in [232], [267], [268], in deriving sparse feature representations for speech, an auto-associative neural network is used, whose internal hidden-layer output is constrained to be sparse. In [268], the fundamental concept of regularization in ML is used, where a sparse regularization term is added to the original reconstruction error or a cross-entropy cost function and by updating the parameters of the network to minimize the overall cost. Significant phonetic recognition error reduction is reported.

Finally, motivated by the sparse Bayesian learning technique and relevance vector machines developed by the ML community (e.g. [269]), an extension is made from the generic unstructured data to structured data of speech and to ASR applications by ASR researchers. In the Bayesian-sensing HMM reported in [270], speech feature sequences are represented using a set of HMM state-dependent basis vectors. Again, model regularization is used to perform sparse Bayesian sensing in face of heterogeneous training data. By incorporating a prior density on sensing weights, the relevance of different bases to a feature vector is determined by the corresponding precision parameters. The model parameters that consist of the basis vectors, the precision matrices of sensing weights and the precision matrices of reconstruction errors, are jointly estimated using a recursive solution, in which the standard Bayesian technique of marginalization (over the weight priors) is exploited. Experimental results reported in [270] as well as in a series of earlier work on a large-scale ASR task show consistent improvements.

2) Analysis and Perspectives: Sparse representation has close links to fundamental ML concepts of regularization and unsupervised feature learning, and also has a deep root in neuroscience. However, its applications to ASR are quite recent and their success, compared with deep learning, is more limited in scope and size, despite the huge success of sparse coding and (sparse) compressive sensing in ML and signal/image processing with a relatively long history.

One possible limiting factor is that the underlying structure of speech features is less prone to sparsification and compression than the image counterpart. Nevertheless, the initial promising ASR results as reviewed above should encourage more work in this direction. It is possible that different types of raw speech features from what have been experimented will have greater potential and effectiveness for sparse representations. As an example, speech waveforms are obviously not a natural candidate for sparse representation but the residual signals after linear prediction would be.

Further, sparseness may not necessarily be exploited for representation purposes only in the unsupervised learning setting. Just as the success of deep learning comes from hybrid between unsupervised generative learning (pre-training) and supervised discriminative learning (fine-tuning), sparseness can be exploited in a similar way. The recent work reported in [271] formulates parameter sparseness as soft regularization and convex constrained optimization problems in a DNN system. Instead of placing sparseness constraint in the DNN's hidden nodes for feature representations as done in [232], [267], [268], sparseness is exploited for reducing non-zero DNN weights. The experimental results in [271] on a large scale ASR task show not only the DNN model size is reduced by 66% to 88%, the error rate is also slightly reduced by 0.2–0.3%. It is a fruitful research direction to exploit sparseness in multiple ways for ASR, and the highly successful deep sparse coding schemes developed by ML and computer vision researchers have yet to enter ASR.

VIII. Discussion and Conclusions

In this overview article, we introduce a set of prominent ML paradigms that are motivated in the context of ASR technology and applications. Throughout this review, readers can see that ML is deeply ingrained within ASR technology, and vice versa. On the one hand, ASR can be regarded only as an instance of a ML problem, just as is any "application" of ML such as computer vision, bioinformatics, and natural language processing. When seen in this way, ASR is a particularly useful ML application since it has extremely large training and test corpora, it is computationally challenging, it has a unique sequential structure in the input, it is also an instance of ML with structured output, and, perhaps most importantly, it has a large community of researchers who are energetically advancing the underlying technology. On the other hand, ASR has been the source of many critical ideas in ML, including the ubiquitous HMM, the concept of classifier adaptation, and the concept of discriminative training on generative models such as HMM—all these were developed and used in the ASR community long before they caught the interest of the ML community. Indeed, our main hypothesis in this review is that these two communities can and should be communicating regularly with each other. Our belief is that the historical and mutually beneficial influence that the communities have had on each other will continue, perhaps at an even more fruitful pace. It is hoped that this overview paper will indeed foster such communication and advancement.

To this end, throughout this overview we have elaborated on the key ML notion of structured classification as a fundamental problem in ASR—with respect to both the symbolic sequence as the ASR classifier's output and the continuous-valued vector feature sequence as the ASR classifier's input. In presenting each of the ML paradigms, we have highlighted the most relevant ML concepts to ASR, and emphasized the kind of ML approaches that are effective in dealing with the special difficulties of ASR including deep/dynamic structure in human speech and strong variability in the observations. We have also paid special attention to discussing and analyzing the major ML paradigms and results that have been confirmed by ASR experiments. The main examples discussed in this article include HMM-related and dynamics-oriented generative learning, discriminative learning for HMM-like generative models, complexity control (regularization) of ASR systems by principled parameter tying, adaptive and Bayesian learning for environment-robust and speaker-robust ASR, and hybrid supervised/unsupervised learning or hybrid generative/discriminative learning as exemplified in the more recent "deep learning" scheme involving DBN and DNN. However, we have also discussed a set of ASR models and methods that have not become mainstream but that have solid theoretical foundation

in ML and speech science, and in combination with other learning paradigms, they offer a potential to make significant contributions. We provide sufficient context and offer insight in discussing such models and ASR examples in connection with the relevant ML paradigms, and analyze their potential contributions.

ASR technology is fast changing in recent years, partly propelled by a number of emerging applications in mobile computing, natural user interface, and AI-like personal assistant technology. So is the infusion of ML techniques into ASR. A comprehensive overview on the topic of this nature unavoidably contains bias as we suggest important research problems and future directions where the ML paradigms would offer the potential to spur next waves of ASR advancement, and as we take position and carry out analysis on a full range of the ASR work spanning over 40 years. In the future, we expect more integrated ML paradigms to be usefully applied to ASR as exemplified by the two emerging ML schemes presented and analyzed in Section VII. We also expect new ML techniques that make an intelligent use of large supply of training data with wide diversity and large-scale optimization (e.g., [272]) to impact ASR, where active learning, semi-supervised learning, and even unsupervised learning will play more important roles than in the past and at present as surveyed in Section V. Moreover, effective exploration and exploitation of deep, hierarchical structure in conjunction with spatially invariant and temporary dynamic properties of speech is just beginning (e.g., [273]). The recent renewed interest in recurrent neural network with deep, multiple-level representations from both ASR and ML communities using more powerful optimization techniques than in the past is an example of the research moving towards this direction. To reap full fruit by such an endeavor will require integrated ML methodologies within and possibly beyond the paradigms we have covered in this paper.

ACKNOWLEDGMENT

The authors thank Prof. Jeff Bilmes for contributions during the early phase (2010) of developing this paper, and for valuable discussions with Geoff Hinton, John Platt, Mark Gales, Nelson Morgan, Hynek Hermansky, Alex Acero, and Jason Eisner. Appreciations also go to MSR for the encouragement and support of this "mentor-mentee project", to Helen Meng as the previous EIC for handling the white-paper reviews during 2009, and to the reviewers whose desire for perfection has made various versions of the revision steadily improve the paper's quality as new advances on ML and ASR frequently broke out throughout the writing and revision over past 3 years.

REFERENCES

[1] J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O'Shgughnessy, "Research developments and directions in speech recognition and understanding, part i," *IEEE Signal Process. Mag.*, vol. 26, no. 3, pp. 75–80, 2009.

[2] X. Huang and L. Deng, "An overview of modern speech recognition," in *Handbook of Natural Language Processing, Second Edition*, N. Indurkhya and F. J. Damerau, Eds. Boca Raton, FL, USA: CRC, Taylor and Francis.

[3] M. Jordan, E. Sudderth, M. Wainwright, and A. Wilsky, "Major advances and emerging developments of graphical models, special issue," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 17–138, Nov. 2010.

[4] J. Bilmes, "Dynamic graphical models," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 29–42, Nov. 2010.

[5] S. Rennie, J. Hershey, and P. Olsen, "Single-channel multitalker speech recognition—Graphical modeling approaches," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 66–80, Nov. 2010.

[6] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, risk bounds," *J. Amer. Statist. Assoc.*, vol. 101, pp. 138–156, 2006.

[7] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley-Interscience, 1998.

[8] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, pp. 273–297, 1995.

[9] D. A. McAllester, "Some PAC-Bayesian theorems," in *Proc. Workshop Comput. Learn. Theory*, 1998.

[10] T. Jaakkola, M. Meila, and T. Jebara, "Maximum entropy discrimination," Mass. Inst. of Technol., Artif. Intell. Lab., Tech. Rep. AITR-1668, 1999.

[11] M. Gales, S. Watanabe, and E. Fosler-Lussier, "Structured discriminative models for speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 70–81, Nov. 2012.

[12] S. Zhang and M. Gales, "Structured SVMs for automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 544–555, Mar. 2013.

[13] F. Pernkopf and J. Bilmes, "Discriminative versus generative parameter and structure learning of Bayesian network classifiers," in *Proc. Int. Conf. Mach. Learn.*, Bonn, Germany, 2005.

[14] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[15] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[16] B.-H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains," *IEEE Trans. Inf. Theory*, vol. IT-32, no. 2, pp. 307–309, Mar. 1986.

[17] L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, and P. Mermelsten, "Phonemic hidden Markov models with continuous mixture output densities for large vocabulary word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 39, no. 7, pp. 1677–1681, Jul. 1991.

[18] J. Bilmes, "What HMMs can do," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 3, pp. 869–891, Mar. 2006.

[19] L. Deng, M. Lennig, F. Seitz, and P. Mermelstein, "Large vocabulary word recognition using context-dependent allophonic hidden Markov models," *Comput., Speech, Lang.*, vol. 4, pp. 345–357, 1991.

[20] 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.

[21] J. Baker, "Stochastic modeling for automatic speech recognition," in *Speech Recogn.*, D. R. Reddy, Ed. New York, NY, USA: Academic, 1976.

[22] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, no. 4, pp. 532–557, Apr. 1976.

[23] 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.

[24] 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, pp. 1–38, 1977.

[25] X. D. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, System Development*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[26] M. Gales and S. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 352–359, Sep. 1996.

[27] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector taylor series for noisy speech recognition," in *Proc. Int. Conf. Spoken Lang, Process.*, 2000, pp. 869–872.

[28] L. Deng, J. Droppo, and A. Acero, "A Bayesian approach to speech feature enhancement using the dynamic cepstral prior," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2002, vol. 1, pp. I-829–I-832.

[29] B. Frey, L. Deng, A. Acero, and T. Kristjansson, "Algonquin: Iterating Laplaces method to remove multiple types of acoustic distortion for robust speech recognition," in *Proc. Eurospeech*, 2000.

[30] J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O'Shgughnessy, "Updated MINDS report on speech recognition and understanding," *IEEE Signal Process. Mag.*, vol. 26, no. 4, pp. 78–85, Jul. 2009.

[31] M. Ostendorf, A. Kannan, O. Kimball, and J. Rohlicek, "Continuous word recognition based on the stochastic segment model," in *Proc. DARPA Workshop CSR*, 1992.

[32] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 360–378, Sep. 1996.

[33] L. Deng, "A generalized hidden Markov model with state-conditioned trend functions of time for the speech signal," *Signal Process.*, vol. 27, no. 1, pp. 65–78, 1992.

[34] L. Deng, M. Aksmanovic, D. Sun, and J. Wu, "Speech recognition using hidden Markov models with polynomial regression functions as non-stationary states," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 2, no. 4, pp. 101–119, Oct. 1994.

[35] W. Holmes and M. Russell, "Probabilistic-trajectory segmental HMMs," *Comput. Speech Lang.*, vol. 13, pp. 3–37, 1999.

[36] H. Zen, K. Tokuda, and T. Kitamura, "An introduction of trajectory model into HMM-based speech synthesis," in *Proc. ISCA SSW5*, 2004, pp. 191–196.

[37] L. Zhang and S. Renals, "Acoustic-articulatory modelling with the trajectory HMM," *IEEE Signal Process. Lett.*, vol. 15, pp. 245–248, 2008.

[38] Y. Gong, I. Illina, and J.-P. Haton, "Modeling long term variability information in mixture stochastic trajectory framework," in *Proc. Int. Conf. Spoken Lang, Process.*, 1996.

[39] L. Deng, G. Ramsay, and D. Sun, "Production models as a structural basis for automatic speech recognition," *Speech Commun.*, vol. 33, no. 2–3, pp. 93–111, Aug. 1997.

[40] L. Deng, "A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition," *Speech Commun.*, vol. 24, no. 4, pp. 299–323, 1998.

[41] J. Picone, S. Pike, R. Regan, T. Kamm, J. Bridle, L. Deng, Z. Ma, H. Richards, and M. Schuster, "Initial evaluation of hidden dynamic models on conversational speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999, pp. 109–112.

[42] J. Bridle, L. Deng, J. Picone, H. Richards, J. Ma, T. Kamm, M. Schuster, S. Pike, and R. Reagan, "An investigation fo segmental hidden dynamic models of speech coarticulation for automatic speech recognition," Final Rep. for 1998 Workshop on Language Engineering, CLSP, Johns Hopkins 1998.

[43] J. Ma and L. Deng, "A path-stack algorithm for optimizing dynamic regimes in a statistical hidden dynamic model of speech," *Comput. Speech Lang.*, vol. 14, pp. 101–104, 2000.

[44] M. Russell and P. Jackson, "A multiple-level linear/linear segmental HMM with a formant-based intermediate layer," *Comput. Speech Lang.*, vol. 19, pp. 205–225, 2005.

[45] L. Deng, *Dynamic Speech Models—Theory, Algorithm, Applications*. San Rafael, CA, USA: Morgan and Claypool, 2006.

[46] J. Bilmes, "Buried Markov models: A graphical modeling approach to automatic speech recognition," *Comput. Speech Lang.*, vol. 17, pp. 213–231, Apr.–Jul. 2003.

[47] L. Deng, D. Yu, and A. Acero, "Structured speech modeling," *IEEE Trans. Speech Audio Process.*, vol. 14, no. 5, pp. 1492–1504, Sep. 2006.

[48] L. Deng, D. Yu, and A. Acero, "A bidirectional target filtering model of speech coarticulation: Two-stage implementation for phonetic recognition," *IEEE Trans. Speech Audio Process.*, vol. 14, no. 1, pp. 256–265, Jan. 2006.

[49] L. Deng, "Computational models for speech production," in *Computational Models of Speech Pattern Processing*. New York, NY, USA: Springer-Verlag, 1999, pp. 199–213.

[50] L. Lee, H. Attias, and L. Deng, "Variational inference and learning for segmental switching state space models of hidden speech dynamics," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2003, vol. 1, pp. I-872–I-875.

[51] J. Droppo and A. Acero, "Noise robust speech recognition with a switching linear dynamic model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, vol. 1, pp. I-953–I-956.

[52] B. Mesot and D. Barber, "Switching linear dynamical systems for noise robust speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 15, no. 6, pp. 1850–1858, Aug. 2007.

[53] A. Rosti and M. Gales, "Rao-blackwellised gibbs sampling for switching linear dynamical systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, vol. 1, pp. I-809–I-812.

[54] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Bayesian nonparametric methods for learning Markov switching processes," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 43–54, Nov. 2010.

[55] E. Ozkan, I. Y. Ozbek, and M. Demirekler, "Dynamic speech spectrum representation and tracking variable number of vocal tract resonance frequencies with time-varying Dirichlet process mixture models," *IEEE Audio, Speech, Lang. Process.*, vol. 17, no. 8, pp. 1518–1532, Nov. 2009.

[56] J.-T. Chien and C.-H. Chueh, "Dirichlet class language models for speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 27, no. 3, pp. 43–54, Mar. 2011.

[57] J. Bilmes, "Graphical models and automatic speech recognition," in *Mathematical Foundations of Speech and Language Processing*, R. Rosenfeld, M. Ostendorf, S. Khudanpur, and M. Johnson, Eds. New York, NY, USA: Springer-Verlag, 2003.

[58] J. Bilmes and C. Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 89–100, Sep. 2005.

[59] H. Zen, M. J. F. Gales, Y. Nankaku, and K. Tokuda, "Product of experts for statistical parametric speech synthesis," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 794–805, Mar. 2012.

[60] D. Barber and A. Cemgil, "Graphical models for time series," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 18–28, Nov. 2010.

[61] A. Miguel, A. Ortega, L. Buera, and E. Lleida, "Bayesian networks for discrete observation distributions in speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 19, no. 6, pp. 1476–1489, Aug. 2011.

[62] L. Deng, "Switching dynamic system models for speech articulation and acoustics," in *Mathematical Foundations of Speech and Language Processing*. New York, NY, USA: Springer-Verlag, 2003, pp. 115–134.

[63] L. Deng and J. Ma, "Spontaneous speech recognition using a statistical coarticulatory model for the hidden vocal-tract-resonance dynamics," *J. Acoust. Soc. Amer.*, vol. 108, pp. 3036–3048, 2000.

[64] L. Deng, J. Droppo, and A. Acero, "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 2, pp. 133–143, Mar. 2004.

[65] V. Stoyanov, A. Ropson, and J. Eisner, "Empirical risk minimization of graphical model parameters given approximate inference, decoding, model structure," in *Proc. AISTAT*, 2011.

[66] V. Goel and W. Byrne, "Minimum Bayes-risk automatic speech recognition," *Comput. Speech Lang.*, vol. 14, no. 2, pp. 115–135, 2000.

[67] V. Goel, S. Kumar, and W. Byrne, "Segmental minimum Bayes-risk decoding for automatic speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 3, pp. 234–249, May 2004.

[68] R. Schluter, M. Nussbaum-Thom, and H. Ney, "On the relationship between Bayes risk and word error rate in ASR," *IEEE Audio, Speech, Lang. Process.*, vol. 19, no. 5, pp. 1103–1112, Jul. 2011.

[69] C. Bishop, *Pattern Recognition and Mach. Learn.*. New York, NY, USA: Springer, 2006.

[70] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 282–289.

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

[72] G. Zweig and P. Nguyen, "SCARF: A segmental conditional random field toolkit for speech recognition," in *Proc. Interspeech*, 2010.

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

[74] 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, 2008.

[75] J. Pylkkonen and M. Kurimo, "Analysis of extended Baum-Welch and constrained optimization for discriminative training of HMMs," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 9, pp. 2409–2419, 2012.

[76] S. Kumar and W. Byrne, "Minimum Bayes-risk decoding for statistical machine translation," in *Proc. HLT-NAACL*, 2004.

[77] X. He and L. Deng, "Speech recognition, machine translation, speech translation—A unified discriminative learning paradigm," *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 126–133, Sep. 2011.

[78] X. He and L. Deng, "Maximum expected BLEU training of phrase and lexicon translation models," *Proc. Assoc. Comput. Linguist.*, pp. 292–301, 2012.

[79] 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.

[80] Q. Fu, Y. Zhao, and B.-H. Juang, "Automatic speech recognition based on non-uniform error criteria," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 780–793, Mar. 2012.

[81] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Eur. Symp. Artif. Neural Netw.*, 1999, pp. 219–224.

[82] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proc. Int. Conf. Mach. Learn.*, 2004.

[83] J. Kuo and Y. Gao, "Maximum entropy direct models for speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 873–881, May 2006.

[84] J. Morris and E. Fosler-Lussier, "Combining phonetic attributes using conditional random fields," in *Proc. Interspeech*, 2006, pp. 597–600.
[85] I. Heintz, E. Fosler-Lussier, and C. Brew, "Discriminative input stream combination for conditional random field phone recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 17, no. 8, pp. 1533–1546, Nov. 2009.
[86] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *IEEE Audio, Speech, Lang. Process.*, vol. 17, no. 2, pp. 354–365, Mar. 2009.
[87] D. Yu, L. Deng, and A. Acero, "Hidden conditional random field with distribution constraints for phone classification," in *Proc. Interspeech*, 2009, pp. 676–679.
[88] D. Yu and L. Deng, "Deep-structured hidden conditional random fields for phonetic recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010.
[89] S. Renals, N. Morgan, H. Boulard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 1, pp. 161–174, Jan. 1994.
[90] H. Boulard and N. Morgan, "Continuous speech recognition by connectionist statistical methods," *IEEE Trans. Neural Netw.*, vol. 4, no. 6, pp. 893–909, Nov. 1993.
[91] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, ser. The Kluwer International Series in Engineering and Computer Science. Boston, MA, USA: Kluwer, 1994, vol. 247.
[92] H. Bourlard and N. Morgan, "Hybrid HMM/ANN systems for speech recognition: Overview and new research directions," in *Adaptive Processing of Sequences and Data Structures*. London, U.K.: Springer-Verlag, 1998, pp. 389–417.
[93] J. Pinto, S. Garimella, M. Magimai-Doss, H. Hermansky, and H. Bourlard, "Analysis of MLP-based hierarchical phoneme posterior probability estimator," *IEEE Audio, Speech, Lang. Process.*, vol. 19, no. 2, pp. 225–241, Feb. 2011.
[94] N. Morgan, Q. Zhu, A. Stolcke, K. Sonmez, S. Sivadas, T. Shinozaki, M. Ostendorf, P. Jain, H. Hermansky, D. Ellis, G. Doddington, B. Chen, O. Cretin, H. Bourlard, and M. Athineos, "Pushing the envelope—Aside [speech recognition]," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 81–88, Sep. 2005.
[95] A. Ganapathiraju, J. Hamaker, and J. Picone, "Hybrid SVM/HMM architectures for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000.
[96] J. Stadermann and G. Rigoll, "A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition," in *Proc. Interspeech*, 2004.
[97] M. Hasegawa-Johnson, J. Baker, S. Borys, K. Chen, E. Coogan, S. Greenberg, A. Juneja, K. Kirchhoff, K. Livescu, S. Mohan, J. Muller, K. Sonmez, and T. Wang, "Landmark-based speech recognition: Report of the 2004 johns hopkins summer workshop," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, pp. 213–216.
[98] S. Zhang, A. Ragni, and M. Gales, "Structured log linear models for noise robust speech recognition," *IEEE Signal Process. Lett.*, vol. 17, 2010.
[99] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of HMM parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dec. 1986, pp. 49–52.
[100] Y. Ephraim and L. Rabiner, "On the relation between modeling approaches for speech recognition," *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 372–380, Mar. 1990.
[101] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, pp. 25–47, 2002.
[102] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large vocabulary speech recognition using minimum classification error," *IEEE Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 203–223, Jan. 2007.
[103] D. Yu, L. Deng, X. He, and A. Acero, "Use of incrementally regulated discriminative margins in mce training for speech recognition," in *Proc. Int. Conf. Spoken Lang, Process.*, 2006, pp. 2418–2421.
[104] D. Yu, L. Deng, X. He, and A. Acero, "Large-margin minimum classification error training: A theoretical risk minimization perspective," *Comput. Speech Lang.*, vol. 22, pp. 415–429, 2008.
[105] 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.
[106] S. Yaman, L. Deng, D. Yu, Y. Wang, and A. Acero, "An integrative and discriminative technique for spoken utterance classification," *IEEE Audio, Speech, Lang. Process.*, vol. 16, no. 6, pp. 1207–1215, Aug. 2008.
[107] Y. Zhang, L. Deng, X. He, and A. Aceero, "A novel decision function and the associated decision-feedback learning for speech translation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 5608–5611.
[108] 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.
[109] F. Sha and L. Saul, "Large margin hidden Markov models for automatic speech recognition," in *Adv. Neural Inf. Process. Syst.*, 2007, vol. 19, pp. 1249–1256.
[110] Y. Eldar, Z. Luo, K. Ma, D. Palomar, and N. Sidiropoulos, "Convex optimization in signal processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 19–145, May 2010.
[111] H. Jiang, X. Li, and C. Liu, "Large margin hidden Markov models for speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1584–1595, Sep. 2006.
[112] X. Li and H. Jiang, "Solving large-margin hidden Markov model estimation via semidefinite programming," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2383–2392, Nov. 2007.
[113] K. Crammer and Y. Singer, "On the algorithmic implementation of multi-class kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2001.
[114] H. Jiang and X. Li, "Parameter estimation of statistical models using convex optimization," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 115–127, May 2010.
[115] F. Sha and L. Saul, "Large margin Gaussian mixture modeling for phonetic classification and recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, 2006, pp. 265–268.
[116] X. Li and J. Bilmes, "A Bayesian divergence prior for classifier adaptation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2007.
[117] 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.
[118] 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.
[119] X. He and L. Deng, *Discriminative Learning for Speech Recognition: Theory and Practice*. San Rafael, CA, USA: Morgan & Claypool, 2008.
[120] G. Heigold, S. Wiesler, M. Nubbaum-Thom, P. Lehnen, R. Schluter, and H. Ney, "Discriminative HMMs. log-linear models, CRFs: What is the difference?," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 5546–5549.
[121] C. Liu, Y. Hu, and H. Jiang, "A trust region based optimization for maximum mutual information estimation of HMMs in speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2474–2485, Nov. 2011.
[122] Q. Fu and L. Deng, "Phone-discriminating minimum classification error (p-mce) training for phonetic recognition," in *Proc. Interspeech*, 2007.
[123] M. Gibson and T. Hain, "Error approximation and minimum phone error acoustic model estimation," *IEEE Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1269–1279, Aug. 2010.
[124] R. Schlueter, W. Macherey, B. Mueller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Commun.*, vol. 31, pp. 287–310, 2001.
[125] R. Chengalvarayan and L. Deng, "HMM-based speech recognition using state-dependent, discriminatively derived transforms on mel-warped DFT features," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 243–256, May 1997.
[126] A. Biem, S. Katagiri, E. McDermott, and B. H. Juang, "An application of discriminative feature extraction to filter-bank-based speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 2, pp. 96–110, Feb. 2001.
[127] B. Mak, Y. Tam, and P. Li, "Discriminative auditory-based features for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 1, pp. 28–36, Jan. 2004.
[128] R. Chengalvarayan and L. Deng, "Speech trajectory discrimination using the minimum classification error learning," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 6, pp. 505–515, Nov. 1998.
[129] K. Sim and M. Gales, "Discriminative semi-parametric trajectory model for speech recognition," *Comput. Speech Lang.*, vol. 21, pp. 669–687, 2007.
[130] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *J. Acoust. Soc. Amer.*, vol. 121, pp. 723–742, 2007.
[131] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Adv. Neural Inf. Process. Syst.*, 1998, vol. 11.
[132] A. McCallum, C. Pal, G. Druck, and X. Wang, "Multi-conditional learning: Generative/discriminative training for clustering and classification," in *Proc. AAAI*, 2006.
[133] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[134] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, pp. 1527–1554, 2006.

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

[136] R. J. A. Little and D. B. Rubin, *Statistical Analysis With Missing Data*. New York, NY, USA: Wiley, 1987.

[137] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," ICSI, Tech. Rep. TR-97-021, 1997.

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

[139] J. Zhu, "Semi-supervised learning literature survey," Computer Sciences, Univ. of Wisconsin-Madison, Tech. Rep., 2006.

[140] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 1999.

[141] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon Univ., Philadelphia, PA, USA, Tech. Rep. CMU-CALD-02, 2002.

[142] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proc. Int. Conf. Mach. Learn.*, 2003.

[143] D. Miller and H. Uyar, "A mixture of experts classifier with learning based on both labeled and unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 1996.

[144] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, pp. 103–134, 2000.

[145] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004.

[146] F. Jiao, S. Wang, C. Lee, R. Greiner, and D. Schuurmans, "Semi-supervised conditional random fields for improved sequence segmentation and labeling," in *Proc. Assoc. Comput. Linguist.*, 2006.

[147] G. Mann and A. McCallum, "Generalized expectation criteria for semi-supervised learning of conditional random fields," in *Proc. Assoc. Comput. Linguist.*, 2008.

[148] X. Li, "On the use of virtual evidence in conditional random fields," in *Proc. EMNLP*, 2009.

[149] J. Bilmes, "On soft evidence in Bayesian networks," Univ. of Washington, Dept. of Elect. Eng., Tech. Rep. UWEETR-2004-0016, 2004.

[150] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1998, pp. 368–374.

[151] O. Chapelle, M. Chi, and A. Zien, "A continuation method for semi-supervised SVMs," in *Proc. Int. Conf. Mach. Learn.*, 2006.

[152] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, 2006.

[153] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. Assoc. Comput. Linguist.*, 1995, pp. 189–196.

[154] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. Workshop Comput. Learn. Theory*, 1998.

[155] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2000.

[156] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincut," in *Proc. Int. Conf. Mach. Learn.*, 2001.

[157] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, vol. 14.

[158] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, 2003.

[159] D. Zhou, O. Bousquet, J. Weston, T. N. Lal, and B. Schlkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003.

[160] V. Sindhwani, M. Belkin, P. Niyogi, and P. Bartlett, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, Nov. 2006.

[161] A. Subramanya and J. Bilmes, "Entropic graph regularization in non-parametric semi-supervised classification," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009.

[162] T. Kemp and A. Waibel, "Unsupervised training of a speech recognizer: Recent experiments," in *Proc. Eurospeech*, 1999.

[163] D. Charlet, "Confidence-measure-driven unsupervised incremental adaptation for HMM-based speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, pp. 357–360.

[164] F. Wessel and H. Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 13, no. 1, pp. 23–31, Jan. 2005.

[165] J.-T. Huang and M. Hasegawa-Johnson, "Maximum mutual information estimation with unlabeled data for phonetic classification," in *Proc. Interspeech*, 2008.

[166] D. Yu, L. Deng, B. Varadarajan, and A. Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion," *Comput. Speech Lang.*, vol. 24, pp. 433–444, 2009.

[167] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Comput. Speech Lang.*, vol. 16, pp. 115–129, 2002.

[168] B. Settles, "Active learning literature survey," Univ. of Wisconsin, Madison, WI, USA, Tech. Rep. 1648, 2010.

[169] D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 1994.

[170] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden Markov models for information extraction," in *Proc. Int. Conf. Adv. Intell. Data Anal. (CAIDA)*, 2001.

[171] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. EMNLP*, 2008.

[172] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 999–1006.

[173] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. ACM Workshop Comput. Learn. Theory*, 1992.

[174] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Mach. Learn.*, pp. 133–168, 1997.

[175] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proc. Int. Conf. Mach. Learn.*, 1995.

[176] H. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 623–630.

[177] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *Proc. Interspeech*, 2009.

[178] A. Guillory and J. Bilmes, "Interactive submodular set cover," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, 2010.

[179] D. Golovin and A. Krause, "Adaptive submodularity: A new approach to active learning and stochastic optimization," in *Proc. Int. Conf. Learn. Theory*, 2010.

[180] G. Riccardi and D. Hakkani-Tur, "Active learning: Theory and applications to automatic speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 4, pp. 504–511, Jul. 2005.

[181] D. Hakkani-Tur, G. Tur, M. Rahim, and G. Riccardi, "Unsupervised and active learning in automatic speech recognition for call classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, pp. 429–430.

[182] D. Hakkani-Tur and G. Tur, "Active learning for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2002, pp. 3904–3907.

[183] Y. Hamanaka, K. Shinoda, S. Furui, T. Emori, and T. Koshinaka, "Speech modeling based on committee-based active learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 4350–4353.

[184] H.-K. J. Kuo and V. Goel, "Active learning with minimum expected error for spoken language understanding," in *Proc. Interspeech*, 2005.

[185] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008.

[186] S. Rüping, "Incremental learning with support vector machines," in *Proc. IEEE. Int. Conf. Data Mining*, 2001.

[187] P. Wu and T. G. Dietterich, "Improving svm accuracy by training on auxiliary data sources," in *Proc. Int. Conf. Mach. Learn.*, 2004.

[188] J.-L. Gauvain and C.-H. Lee, "Bayesian learning of Gaussian mixture densities for hidden Markov models," in *Proc. DARPA Speech and Natural Language Workshop*, 1991, pp. 272–277.

[189] J.-L. Gauvain and C.-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[190] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2003, pp. 224–227.

[191] C. Chelba and A. Acero, "Adaptation of maximum entropy capitalizer: Little data can help a lot," in *Proc. EMNLP*, July 2004.

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

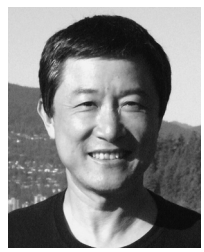
[193] M. Gales and P. Woodland, "Mean and variance adaptation within the mllr framework," *Comput. Speech Lang.*, vol. 10, 1996.

[194] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," in *Proc. Eurospeech*, 1995.

[195] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Proc. Eurospeech*, 1995.

[196] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, 1997.
[197] J. Baxter, "Learning internal representations," in *Proc. Workshop Comput. Learn. Theory*, 1995.
[198] H. Daumé and D. Marcu, "Domain adaptation for statistical classifiers," *J. Artif. Intell. Res.*, vol. 26, pp. 1–15, 2006.
[199] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Multiple source adaptation and the Renyi divergence," in *Proc. Uncertainty Artif. Intell.*, 2009.
[200] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Proc. Workshop Comput. Learn. Theory*, 2009.
[201] L. Deng, *Front-End, Back-End, Hybrid Techniques to Noise-Robust Speech Recognition. Chapter 4 in Book: Robust Speech Recognition of Uncertain Data*. Berlin, Germany: Springer-Verlag, 2011.
[202] G. Zavaliagkos, R. Schwarz, J. McDonogh, and J. Makhoul, "Adaptation algorithms for large scale HMM recognizers," in *Proc. Eurospeech*, 1995.
[203] C. Chesta, O. Siohan, and C. Lee, "Maximum *a posteriori* linear regression for hidden Markov model adaptation," in *Proc. Eurospeech*, 1999.
[204] T. Myrvoll, O. Siohan, C.-H. Lee, and W. Chou, "Structural maximum a posteriori linear regression for unsupervised speaker adaptation," in *Proc. Int. Conf. Spoken Lang, Process.*, 2000.
[205] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. Int. Conf. Spoken Lang, Process.*, 1996, pp. 1137–1140.
[206] L. Deng, A. Acero, M. Plumpe, and X. D. Huang, "Large vocabulary speech recognition under adverse acoustic environment," in *Proc. Int. Conf. Spoken Lang, Process.*, 2000, pp. 806–809.
[207] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 1889–1901, Nov. 2010.
[208] L. Deng, K. Wang, A. Acero, H. Hon, J. Droppo, Y. Wang, C. Boulis, D. Jacoby, M. Mahajan, C. Chelba, and X. Huang, "Distributed speech processing in mipad's multimodal user interface," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 9, pp. 2409–2419, Nov. 2012.
[209] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 568–580, Nov. 2003.
[210] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series," in *Proc. IEEE Workshop Autom. Speech Recogn. Understand.*, Dec. 2007, pp. 65–70.
[211] J. Y. Li, L. Deng, Y. Gong, and A. Acero, "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," *Comput. Speech Lang.*, vol. 23, pp. 389–405, 2009.
[212] M. Padmanabhan, L. R. Bahl, D. Nahamoo, and M. Picheny, "Speaker clustering and transformation for speaker adaptation in speech recognition systems," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 1, pp. 71–77, Jan. 1998.
[213] M. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.
[214] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.
[215] A. Gliozzo and C. Strapparava, "Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization," in *Proc. Assoc. Comput. Linguist.*, 2006.
[216] J. Ham, D. Lee, and L. Saul, "Semisupervised alignment of manifolds," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2005.
[217] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009.
[218] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008.
[219] H. Daume, "Cross-task knowledge-constrained self training," in *Proc. EMNLP*, 2008.
[220] J. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.*, vol. 12, pp. 149–198, 2000.
[221] S. Thrun and L. Y. Pratt, *Learning To Learn*. Boston, MA, USA: Kluwer, 1998.
[222] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Proc. Comput. Learn. Theory*, 2003.
[223] R. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, 2005.
[224] J. Baxter, "A Bayesian/information theoretic model of learning to learn via multiple task sampling," *Mach. Learn.*, pp. 7–39, 1997.
[225] T. Heskes, "Empirical Bayes for learning to learn," in *Proc. Int. Conf. Mach. Learn.*, 2000.
[226] K. Yu, A. Schwaighofer, and V. Tresp, "Learning Gaussian processes from multiple tasks," in *Proc. Int. Conf. Mach. Learn.*, 2005.
[227] Y. Xue, X. Liao, and L. Carin, "Multi-task learning for classification with Dirichlet process priors," *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, 2007.
[228] H. Daume, "Bayesian multitask learning with latent hierarchies," in *Proc. Uncertainty in Artif. Intell.*, 2009.
[229] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, pp. 615–637, 2005.
[230] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying, "Spectral regularization framework for multi-task structure learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007.
[231] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011.
[232] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Proc. Interspeech*, 2010.
[233] H. Lin, L. Deng, D. Yu, Y. Gong, and A. Acero, "A study on multilingual acoustic modeling for large vocabulary ASR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 4333–4336.
[234] D. Yu, L. Deng, P. Liu, J. Wu, Y. Gong, and A. Acero, "Cross-lingual speech recognition under run-time resource constraints," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 4193–4196.
[235] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next-generation automatic speech recognition," in *Proc. Int. Conf. Spoken Lang, Process.*, 2004, pp. 109–111.
[236] I. Bromberg, Q. Qian, J. Hou, J. Li, C. Ma, B. Matthews, A. Moreno-Daniel, J. Morris, M. Siniscalchi, Y. Tsao, and Y. Wang, "Detection-based ASR in the automatic speech attribute transcription project," in *Proc. Interspeech*, 2007, pp. 1829–1832.
[237] L. Deng and D. Sun, "A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features," *J. Acoust. Soc. Amer.*, vol. 85, pp. 2702–2719, 1994.
[238] J. Sun and L. Deng, "An overlapping-feature based phonological model incorporating linguistic constraints: Applications to speech recognition," *J. Acoust. Soc. Amer.*, vol. 111, pp. 1086–1101, 2002.
[239] 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.
[240] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, pp. 3207–3220, 2010.
[241] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.
[242] B. Hutchinson, L. Deng, and D. Yu, "A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4805–4808.
[243] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, to be published.
[244] G. Andrew and J. Bilmes, "Sequential deep belief networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4265–4268.
[245] D. Yu, S. Siniscalchi, L. Deng, and C. Lee, "Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4169–4172.
[246] G. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 4688–4691.
[247] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4153–4156.
[248] 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.
[249] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4277–4280.
[250] D. Yu, L. Deng, and G. Dahl, "Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2010.

[251] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, and M. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2011, pp. 5060–5063.
[252] D. Yu, L. Deng, and F. Seide, "Large vocabulary speech recognition using deep tensor neural networks," in *Proc. Interspeech*, 2012.
[253] Z. Tuske, M. Sundermeyer, R. Schluter, and H. Ney, "Context-dependent MLPs for LVCSR: Tandem, hybrid or both," in *Proc. Interspeech*, 2012.
[254] G. Saon and B. Kingbury, "Discriminative feature-space transforms using deep neural networks," in *Proc. Interspeech*, 2012.
[255] R. Gens and P. Domingos, "Discriminative learning of sum-product networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012.
[256] O. Vinyals, Y. Jia, L. Deng, and T. Darrell, "Learning with recursive perceptual representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012.
[257] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[258] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 7–13, Jan. 2012.
[259] D. Yu, L. Deng, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 388–396, Feb. 2013.
[260] M. Siniscalchi, L. Deng, D. Yu, and C.-H. Lee, "Exploiting deep neural networks for detection-based speech recognition," *Neurocomputing*, 2013.
[261] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in *Proc. Interspeech*, 2010.
[262] T. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, and A. Sethy, "Exemplar-based sparse representation features for speech recognition," in *Proc. Interspeech*, 2010.
[263] T. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky, "Exemplar-based sparse representation features: From TIMIT to LVCSR," *IEEE Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2598–2613, Nov. 2011.
[264] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template-based continuous speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1377–1390, May 2007.
[265] J. Gemmeke, U. Remes, and K. J. Palomki, "Observation uncertainty measures for sparse imputation," in *Proc. Interspeech*, 2010.
[266] J. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2067–2080, Sep. 2011.
[267] G. Sivaram, S. Ganapathy, and H. Hermansky, "Sparse auto-associative neural networks: Theory and application to speech recognition," in *Proc. Interspeech*, 2010.
[268] G. Sivaram and H. Hermansky, "Sparse multilayer perceptron for phoneme recognition," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 23–29, Jan. 2012.
[269] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, pp. 211–244, 2001.
[270] G. Saon and J. Chien, "Bayesian sensing hidden Markov models," *IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 43–54, Jan. 2012.
[271] D. Yu, F. Seide, G. Li, and L. Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4409–4412.
[272] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large scale distributed deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012.
[273] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, to be published.



Li Deng (F'05) received the Ph.D. degree from the University of Wisconsin-Madison. He joined Dept. Electrical and Computer Engineering, University of Waterloo, Ontario, Canada in 1989 as an assistant professor, where he became a tenured full professor in 1996. In 1999, he joined Microsoft Research, Redmond, WA as a Senior Researcher, where he is currently a Principal Researcher. Since 2000, he has also been an Affiliate Full Professor and graduate committee member in the Department of Electrical Engineering at University of Washington, Seattle. Prior to MSR, he also worked or taught at Massachusetts Institute of Technology, ATR Interpreting Telecom. Research Lab. (Kyoto, Japan), and HKUST. In the general areas of speech/language technology, machine learning, and signal processing, he has published over 300 refereed papers in leading journals and conferences and 3 books, and has given keynotes, tutorials, and distinguished lectures worldwide. He is a Fellow of the Acoustical Society of America, a Fellow of the IEEE, and a Fellow of ISCA. He served on the Board of Governors of the IEEE Signal Processing Society (2008–2010). More recently, he served as Editor-in-Chief for the *IEEE Signal Processing Magazine* (2009–2011), which earned the highest impact factor among all IEEE publications and for which he received the 2011 IEEE SPS Meritorious Service Award. He currently serves as Editor-in-Chief for the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING. His recent technical work (since 2009) and leadership on industry-scale deep learning with colleagues and collaborators have created significant impact on speech recognition, signal processing, and related applications.

Xiao Li (M'07) received the B.S.E.E degree from Tsinghua University, Beijing, China, in 2001 and the Ph.D. degree from the University of Washington, Seattle, in 2007. In 2007, she joined Microsoft Research, Redmond as a researcher. Her research interests include speech and language understanding, information retrieval, and machine learning. She has published over 30 referred papers in these areas, and is a reviewer of a number of IEEE, ACM, and ACL journals and conferences. At MSR she worked on search engines by detecting and understanding a user's intent with a search query, for which she was honored with MIT Technology Reviews TR35 Award in 2011. After working at Microsoft Research for over four years, she recently embarked on a new adventure at Facebook Inc. as a research scientist.