1

Introduction

Samy Bengio[1] and Joseph Keshet[2]

[1] Google Inc., Mountain View, CA, USA
[2] IDIAP Research Institute, Martigny, Switzerland

One of the most natural communication tool used by humans is their voice. It is hence natural that a lot of research has been devoted to analyze and understand human uttered speech for various applications. The most obvious one is **automatic speech recognition**, where the goal is to transcribe a recorded speech utterance into its corresponding sequence of words. Other applications include **speaker recognition**, where the goal is to either determine the claimed identity of the speaker (verification) or who is speaking (identification), and speaker segmentation or diarization, where the goal is to segment an acoustic sequence in terms of the underlying speakers (such as during a dialog).

Although enormous amount of research has been devoted to speech processing, there appear to be some form of local optimum in terms of the fundamental tools used to approach these problems. The aim of this book is to introduce the speech researcher community with radically different approaches based on more recent kernel based machine learning approaches. In this introduction, we first briefly remind the main speech processing approach, based on hidden Markov models, as well as its known problems, then introduce the most well known kernel based approach, the Support Vector Machine (SVM), and finally opens to the various contributions of this book.

Speech and Speaker Recognition: Large Margin and Kernel Methods. Edited by J. Keshet and S. Bengio

1.1 The Traditional Approach to Speech Processing

Most speech processing problems, including speech recognition, speaker verification, speaker segmentation, etc., proceed with basically the same general approach, which is described here in the context of speech recognition, as this is the field that has attracted most of the research in the last 40 years. The approach is based on the following statistical framework.

A sequence of acoustic feature vectors is extracted from a spoken utterance by a front-end signal processor. We denote the sequence of acoustic feature vectors by $\bar{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathcal{X}$ and $\mathcal{X} \subset \mathbb{R}^d$ is the domain of the acoustic vectors. Each vector is a compact representation of the short-time spectrum. Typically, each vector covers a period of 10 msec and there are approximately $T = 300$ acoustic vectors in a 10 word utterance. The spoken utterance consists of a sequence of words $\bar{v} = (v_1, \ldots, v_N)$. Each of the words belongs to a fixed and known vocabulary $\mathcal{V}$, that is, $v_i \in \mathcal{V}$. The task of the speech recognizer is to predict the most probable word sequence $\bar{v}'$ given the acoustic signal $\bar{\mathbf{x}}$. Speech recognition is formulated as a *maximum a posteriori* (MAP) decoding problem as follows

$$\bar{v}' = \arg\max_{\bar{v}} P(\bar{v}|\bar{\mathbf{x}}) = \arg\max_{\bar{v}} \frac{p(\bar{\mathbf{x}}|\bar{v})P(\bar{v})}{p(\bar{\mathbf{x}})} \,, \tag{1.1}$$

where we used Bayes' rule to decompose the posterior probability in the last equation. The term $p(\bar{\mathbf{x}}|\bar{v})$ is the probability of observing the acoustic vector sequence $\bar{\mathbf{x}}$ given a specified word sequence $\bar{v}$ and it is known as *the acoustic model*. The term $P(\bar{v})$ is the probability of observing a word sequence $\bar{v}$ and it is known as *the language model*. The term $p(\bar{\mathbf{x}})$ can be disregarded, since it is constant under the $\max$ operation.

The acoustic model is usually estimated by a Hidden Markov Model (HMM) (Rabiner and Juang 1993), a kind of graphical model (Jordan 1999) that represents the joint probability of an observed variable and a hidden (or latent) variable. In order to understand the acoustic model, we now describe the basic HMM decoding process. By decoding we mean the calculation of the $\arg\max_{\bar{v}}$ in Equation (1.1). The process starts with an assumed word sequence $\bar{v}$. Each word in this sequence is converted into a sequence of basic spoken units called *phones*[1] using a pronunciation dictionary. Each phone is represented by a single HMM, where the HMM is a probabilistic state machine typically composed of three states (which are the hidden or latent variables) in a left-to-right topology. Assume that $\mathcal{Q}$ is the set of all states, and let $\bar{q}$ be a sequence of states, that is $\bar{q} = (q_1, q_2, \ldots, q_T)$, where it is assumed there exists some latent random variable $q_t \in \mathcal{Q}$ for each frame $\mathbf{x}_t$ of $\bar{\mathbf{x}}$. Wrapping up, the sequence of words $\bar{v}$ is converted into a sequence of phones $\bar{p}$ using a pronunciation dictionary, and the sequence of phones is converted to a sequence of states, with in general at least 3 states per phone. The goal now is to find the most probable sequence of states.

Formally, the HMM is defined as a pair of random processes $\bar{q}$ and $\bar{\mathbf{x}}$, where the following first order Markov assumptions are made:

I. $P(q_t|q_1, q_2, \ldots, q_{t-1}) = P(q_t|q_{t-1})$; and

II. $p(\mathbf{x}_t|\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_T, q_1, \ldots, q_T) = p(\mathbf{x}_t|q_t)$.

[1] A *phone* is a consonant or vowel speech sound. A *phoneme* is any equivalent set of phones which leaves a word meaning invariant (Allen 2005).

The HMM is a *generative model* and can be thought of as a generator of acoustic vector sequences. During each time unit (frame), the model can change a state with probability $P(q_t|q_{t-1})$, also known as the *transition probability*. Then, at every time step, an acoustic vector is emitted with probability $p(\mathbf{x}_t|q_t)$, sometimes referred to as the *emission probability*. In practice the sequence of states is not observable; hence the model is called hidden. The probability of the state sequence $\bar{q}$ given the observation sequence $\bar{\mathbf{x}}$ can be found using Bayes' rule as follows,

$$P(\bar{q}|\bar{\mathbf{x}}) = \frac{p(\bar{\mathbf{x}}, \bar{q})}{p(\bar{\mathbf{x}})} ,$$

where the joint probability of a vector sequence $\bar{\mathbf{x}}$ and a state sequence $\bar{q}$ is calculated simply as a product of the transition probabilities and the output probabilities,

$$p(\bar{\mathbf{x}}, \bar{q}) = P(q_0) \prod_{t=1}^{T} P(q_t|q_{t-1})\, p(\mathbf{x}_t|q_t) , \tag{1.2}$$

where we assumed that q_0 is constrained to be a non-emitting initial state. The emission density distributions $p(\mathbf{x}_t|q_t)$ are often estimated using diagonal covariance Gaussian Mixture Models (GMMs) for each state q_t, which model the density of a d-dimensional vector $\mathbf{x}$ as follows:

$$p(\mathbf{x}) = \sum_i w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i); \tag{1.3}$$

where $w_i \in \mathbb{R}$ is positive with $\sum_i w_i = 1$, and $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\sigma})$ is a Gaussian with mean $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and standard deviation $\boldsymbol{\sigma}_i \in \mathbb{R}^d$. Given the HMM parameters in the form of the transition probability and emission probability (as GMMs), the problem of finding the most probable state sequence is found by maximizing $p(\bar{\mathbf{x}}, \bar{q})$ over all possible state sequences using the *Viterbi algorithm* (Rabiner and Juang 1993).

In the training phase, the model parameters are estimated. Assume one has access to a training set of m examples $\mathcal{T}_{\text{train}} = \{(\bar{\mathbf{x}}^i, \bar{v}^i)\}_{i=1}^m$. Training of the acoustic model and the language model can be done in two separate steps. The acoustic model parameters include the transition probabilities and the emission probabilities, and they are estimated by a procedure known as the *Baum-Welch algorithm* (Baum et al. 1970), which is a special case of the expectation-maximization (EM) algorithm, when applied to HMMs. This algorithm provides a very efficient procedure to estimate these probabilities iteratively. The parameters of the HMMs are chosen to maximize the probability of the acoustic vector sequence $p(\bar{\mathbf{x}})$ given a virtual HMM composed as the concatenation of the phone HMMs that correspond to the underlying sequence of words $\bar{v}$. The Baum-Welch algorithm monotonically converges in polynomial time (with respect to the number of states and the length of the acoustic sequences) to local stationary points of the likelihood function.

Language models are used to estimate the probability of a given sequence of words, $P(\bar{v})$. The language model is often estimated by n-grams (Manning and Schutze 1999), where the probability of a sequence of N words $(\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_N)$ is estimated as follows:

$$p(\bar{v}) \approx \prod_t p(v_t|v_{t-1}, v_{t-2}, \ldots, v_{t-N}) \tag{1.4}$$

where each term can be estimated on a large corpus of written document by simply counting the occurrences of each n-gram. Various smoothing and back-off strategies have been developed in the case of large n where most n-grams would be poorly estimated even using very large text corpora.

1.2 Potential Problems of the Probabilistic Approach

Although most state-of-the-art approaches to speech recognition are based on the use of HMMs and GMMs, also called continuous-density HMMs (or CD-HMMs) they have several drawbacks, some of which we discuss hereafter.

- Consider the logarithmic form of Equation (1.2),

$$\log p(\bar{\mathbf{x}}, \bar{q}) = \log P(q_0) + \sum_{t=1}^{T} \log P(q_t|q_{t-1}) + \sum_{t=1}^{T} \log p(\mathbf{x}_t|q_t) \; . \tag{1.5}$$

  There is a known structural problem when mixing densities $p(\mathbf{x}_t|q_t)$ and probabilities $P(q_t|q_{t-1})$: the global likelihood is mostly influenced by the emission distributions and almost not by the transition probabilities, hence temporal aspects are poorly taken into account (Bourlard et al. 1996; Young 1996). This happens mainly because the variance of densities of the emission distribution depends on d the actual dimension of the acoustic features: the higher d, the higher the expected variance of $p(\bar{\mathbf{x}}|\bar{q})$, while the variance of the transition distributions mainly depend on the number of states of the HMM. In practice, one can observe a ratio of about 100 between these variances, hence when selecting the best sequence of words for a given acoustic sequence, only the emission distributions are taken into account. Although the latter may well be very well estimated using GMMs, they do not take into account most temporal dependencies between them (which are supposed to be modeled by transitions).

- While the EM algorithm is very well known and efficiently implemented for HMMs, it can only converge to local optima, and hence optimization may greatly vary according to initial parameter settings. For CD-HMMs, the Gaussian means and variances are often initialized using K-Means, which is itself also known to be very sensitive to initialization.

- Not only EM is known to be prone to local optimal, it is basically used to maximize the likelihood of the observed acoustic sequence, in the context of the expected sequence of words. Note however that the performance of most speech recognizers are estimated using other measures than the likelihood. In general, one is interested in minimizing the number of errors in the generated word sequence. This is often done by computing the Levenshtein distance between the expected and the obtained word sequences, and is often known as the *word error rate*. There might be a significant difference between the best HMM models according to the maximum likelihood criterion and the word error rate criterion.

Hence, throughout the years, various alternatives have been proposed. One line of research has been centered around proposing more discriminative training algorithms for

HMMs. That includes Maximum Mutual Information Estimation (MMIE) (Bahl et al. 1986), Minimum Classification Error (MCE) (Juang and Katagiri 1992), Minimum Phone Error (MPE) and Minimum Word Error (MWE) (Povey and Woodland 2002). All these approaches, although proposing better training criteria, still suffer from most of the drawbacks described earlier (local minima, useless transitions).

The last 15 years of research in the machine learning community has welcomed the introduction of so-called large margin and kernel approaches, of which the Support Vector Machine (SVM) is its best known example. An important topic of this book is to show how these recent effort from the machine learning community can be used to improve research in the speech processing domain. Hence, the next section is devoted to a brief introduction to SVMs.

1.3 Support Vector Machines for Binary Classification

The most well known kernel based machine learning approach is the Support Vector Machine (SVM) (Vapnik 2000). While it was not developed in particular for speech processing, most of the chapters in this book propose kernel methods that are in one way or another inspired by the SVM.

Let us assume we are given a training set of m examples $\mathcal{T}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional input vector and $y_i \in \{-1, 1\}$ is the target class. The simplest binary classifier one can think of is the linear classifier, where we are looking for parameters $(\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R})$ such that

$$\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \ . \tag{1.6}$$

When the training set is said to be linearly separable, there is potentially an infinite number of solutions $(\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R})$ that satisfy (1.6). Hence, the SVM approach looks for the one that maximizes the *margin* between the two classes, where the margin can be defined as the sum of the smallest distances between the separating hyper-plane and points of each class. This concept is illustrated in Figure 1.1.

This can be expressed by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } \forall i \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \ . \end{aligned} \tag{1.7}$$

While this is difficult to solve, its following dual formulation is computationally more efficient:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^n y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to } & \begin{cases} \forall i \quad \alpha_i \geq 0 \\ \displaystyle\sum_{i=1}^m \alpha_i y_i = 0 \ . \end{cases} \end{aligned} \tag{1.8}$$

One problem with this formulation is that if the problem is not linearly separable, there might be no solution to it. Hence one can relax the constraints by allowing errors with an

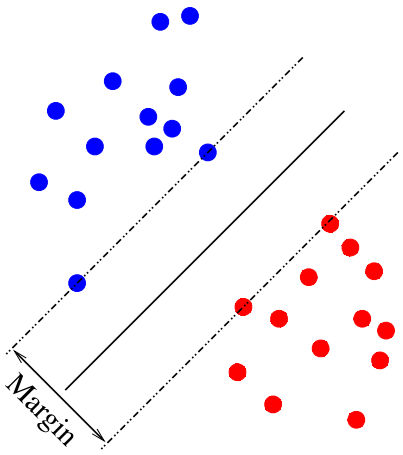

Figure 1.1 Illustration of the notion of margin.

additional hyper-parameter C that controls the trade-off between maximizing the margin and minimizing the number of training errors, as follows:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \tag{1.9}$$

$$\text{subject to} \begin{cases} \forall i & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \\ \forall i & \xi_i \geq 0 \end{cases}$$

which dual becomes

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{1.10}$$

$$\text{subject to} \begin{cases} \forall i & 0 \leq \alpha_i \leq C \\ \displaystyle\sum_{i=1}^{m} \alpha_i y_i = 0 \ . \end{cases}$$

In order to look for non-linear solutions, one can easily replace $\mathbf{x}$ by some non-linear function $\boldsymbol{\phi}(\mathbf{x})$. It is interesting to note that $\mathbf{x}$ only appears in dot products in (1.10). It has thus been proposed to replace all occurrences of $\boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j)$ by some kernel function $k(\mathbf{x}_i,\mathbf{x}_j)$. As long as $k(\cdot,\cdot)$ lives in a reproducing kernel Hilbert space (RKHS), one can guarantee that there exists some function $\boldsymbol{\phi}(\cdot)$ such that

$$k(\mathbf{x}_i,\mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)\cdot\boldsymbol{\phi}(\mathbf{x}_j) \ .$$

Thus, even if $\boldsymbol{\phi}(\mathbf{x})$ projects $\mathbf{x}$ in a very high (possibly infinite) dimensional space, $k(\mathbf{x}_i,\mathbf{x}_j)$ can still be efficiently computed.

Problem (1.10) can be solved using off-the-shelf quadratic optimization tools. Note however that the underlying computational complexity is at least quadratic in the number of training examples, which can often be a serious limit for most speech processing applications.

After solving (1.10), the resulting SVM solution takes the form of

$$\hat{y}(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{1.11}$$

where most α_i are zero except those corresponding to examples in the margin or misclassified, often called *support vectors* (hence the name of SVMs).

1.4 Outline

The book has four parts. The first part, **Foundations**, covers important aspects of extending the binary support vector machine to speech and speaker recognition applications. Chapter 1 provides a detailed review on efficient and practical solutions to large scale convex optimization problems one encounters when using large margin and kernel methods with the enormous datasets used in speech applications. Chapter 2 presents an extension of the binary support vector machine to multiclass, hierarchical and categorical classification. Specifically, the chapter presents a more complex setting in which the possible labels or categories are many and organized.

The second part, **Acoustic Modeling**, deals with large margin and kernel method algorithms for sequence prediction required for acoustic modeling. Chapter 4 presents a large margin algorithm for forced alignment of a phoneme sequence to a corresponding speech signal, that is, proper positioning of a sequence of phonemes in relation to a corresponding continuous speech signal. Chapter 5 describes a kernel wrapper for the task of phoneme recognition, which is based on the Gaussian kernel. This chapter also presents a kernel-based iterative algorithm aims at minimizing the Levenshtein distance between the predicted phoneme sequence and the true one. Chapter 6 reviews the use of dynamic kernels for acoustic models and especially describes the augmented statistical models, resulted from the generative kernel, a generalization of the Fisher kernel. Chapter 7 investigates a framework for large margin parameter estimation for continuous-density HMMs.

The third part of the book is devoted to **Language Modeling**. Chapter 8 reviews past and present work on discriminative training of language models, and focuses on three key issues: training data, learning algorithms, and features. Chapter 9 describes different large margin algorithms for the application of part-of-speech tagging. Chapter 10 presents a proposal for large vocabulary continuous speech recognition, which is solely based on large margin and kernel methods, incorporating the acoustic models described in Part II and the discriminative language models.

The last part is dedicated to **Applications**. Chapter 11 covers a discriminative keyword spotting algorithm, based on a large margin approach, which aims at maximizing the area under the ROC curve, the most common measure to evaluate keyword spotters. Chapter 12 surveys recent work on the use of kernel approaches to text-independent speaker verification. Finally, Chapter 13 introduces the main concepts and algorithms together with recent advances in learning a similarity matrix from data. The techniques in the chapter are illustrated on the blind one-microphone speech separation problem, by casting the problem as one of segmentation of the spectrogram.

References

Allen JB 2005 *Articulation and Intelligibility*. Morgan & Claypool.

Bahl LR, Brown PF, de Souza PV and Mercer RL 1986 Maximum mutual information of hidden Markov model parameters for speech recognition *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 49–53.

Baum LE, Petrie T, Soules G and Weiss N 1970 A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics* **41**(1), 164–171.

Bourlard H, Hermansky H and Morgan N 1996 Towards increasing speech recognition error rates. *Speech Communication* **18**, 205–231.

(ed. Jordan MI) 1999 *Learning in Graphical Models*. MIT Press.

Juang BH and Katagiri S 1992 Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*.

Manning CD and Schutze H 1999 *Foundations of Statistical Natural Language Processing*. MIT Press.

Povey D and Woodland PC 2002 Minimum phone error and I-smoothing for improved discriminative training *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Rabiner L and Juang BH 1993 *Fundamentals of speech recognition* first edn. Prentice All.

Vapnik VN 2000 *The nature of statistical learning theory* second edn. Springer.

Young S 1996 A review of large-vocabulary continuous speech recognition. *IEEE Signal Processing Mag.* pp. 45–57.