A THEORY OF (ALMOST) ZERO RESOURCE SPEECH RECOGNITION

BY

SUJEETH SUBRAMANYA BHARADWAJ

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Mark Hasegawa-Johnson, Chair
Professor Stephen E. Levinson
Associate Professor Feng Liang
Assistant Professor Paris Smaragdis

ABSTRACT

Automatic speech recognition has matured into a commercially successful technology, enabling voice-based interfaces for smartphones, smart TVs, and many other consumer devices. The overwhelming popularity, however, is still limited to languages such as English, Japanese, and German, where vast amounts of labeled training data are available. For most other languages, it is prohibitively expensive to 1) collect and transcribe the speech data required to learn good acoustic models; and 2) acquire adequate text to estimate meaningful language models. A theory of unsupervised and semi-supervised techniques for speech recognition is therefore essential. This thesis focuses on HMM-based sequence clustering and examines acoustic modeling, language modeling, and applications beyond the components of an ASR, such as anomaly detection, from the vantage point of PAC-Bayesian theory.

The first part of this thesis extends standard PAC-Bayesian bounds to address the sequential nature of speech and language signals. A novel algorithm, based on sparsifying the cluster assignment probabilities with a Renyi entropy prior, is shown to provably minimize the generalization error of any probabilistic model (e.g. HMMs).

The second part examines application-specific loss functions such as cluster purity and perplexity. Empirical results on a variety of tasks – acoustic event detection, class-based language modeling, and unsupervised sequence anomaly detection – confirm the practicality of the theory and algorithms developed in this thesis.

"Take up one idea. Make that one idea your life – think of it, dream of it, live on that idea. Let the brain, muscles, nerves, every part of your body be full of that idea, and just leave every other idea alone. This is the way to success, and this is the way great spiritual giants are produced."

–Swami Vivekananda

ACKNOWLEDGMENTS

It would be impossible for me to overstate the respect and gratitude I have for Professor Mark Hasegawa-Johnson – a wonderful teacher, thinker, and human being. He gave me full freedom to craft my own journey along with the support structure to make things happen. Thanks to Mark, I have never had to think about anything other than my own research. This is the greatest gift an adviser can give to his student.

I am also grateful to Professor Stephen Levinson. For the last eight years, he has encouraged me in all of my endeavors, including my first interaction with Mark. His commitment to science and education has been a constant source of inspiration for me.

I am honored to have Professors Feng Liang and Paris Smaragdis on my PhD committee – their feedback has been extremely valuable.

The Statistical Speech Technology (SST) group has an incredible research environment. Thanks to all of my labmates, past and present, for the many discussions we have had on speech recognition.

Most of my PhD work was funded by grants from NSF, Beckman Institute, and NCSA, as well as the ECE Distinguished Fellowship, Joan and Lalit Bahl Fellowship, and James M. Henderson Fellowship. I thank these organizations and donors for their generosity.

I am truly fortunate to have worked with some amazing people at TTIC, IBM, Intel, and Google. I am grateful to all of my industry mentors and colleagues for teaching me a few practical skills.

Thanks to friends and family – both in US and in India – for adding some color to my life. All those random (and stupid) conversations kept me sane.

Above all, I am indebted to my parents and sister. None of this would have been possible without their love, support, guidance, and *ashirvadam.*

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Automatic speech recognition (ASR) has evolved from trivial command-and-control applications to large-vocabulary continuous-speech systems that can easily run on a smartphone. The underlying theory and algorithms, to this day, are based on the hidden Markov model (HMM). What has advanced over the years is therefore not theory, but rather techniques and the computing power required to leverage increasing amounts of labeled data. Popular approaches to ASR are almost exclusively designed for English, Japanese, and other such languages where transcribed speech data is abundant.

Comparable datasets simply do not exist for a majority of the nearly 7000 known languages, and it is prohibitively expensive to 1) collect and transcribe the speech data required to learn good acoustic models; and 2) acquire adequate text to estimate meaningful language models. In this thesis, we extend the theory of speech recognition to a zero-resource setting, in which almost no labeled data are available.

1.1 Motivation

To further motivate the importance of unsupervised and semi-supervised techniques, we briefly examine the linguistic landscape of India. With over a billion people distributed across at least 30 major languages (each spoken by more than a million), India serves as a good case study.

Figure 1.1 provides a ranking of Indian languages by population [1]. Hindi is a widely spoken language with nearly as many native speakers as English; however, commercial ASR solutions do not support Hindi, let alone other significant (by population) languages such as Bengali, Telugu, Marathi, etc. There are two main reasons for the lack of interest in Indian languages: 1) English is a *lingua franca* among the educated Indian elite, and although

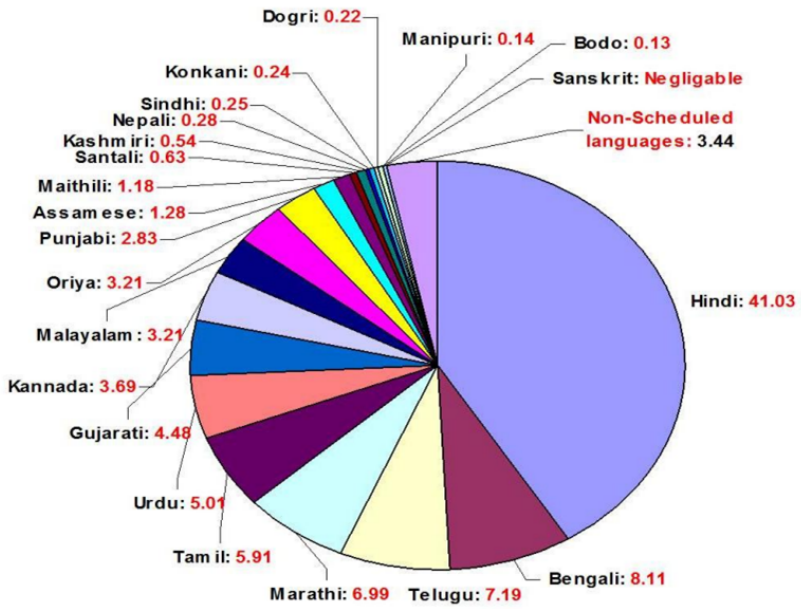

Figure 1.1: Indian languages by population (% of total population)

there are only 85 million Indians who speak English, the segment of the population already includes many of the financially stable and tech savvy people; 2) there are very few transcribed datasets that can support non-trivial applications.

We focus on the second problem. A trivial solution is to collect the necessary data and replicate what is already successful for English; but such an approach is certainly not scalable, and can also be limited in scope. For example, Indian speakers are often bilingual or trilingual, and colloquial speech consists of code-switching across multiple languages. There are several other situations, both within and outside the context of Indian languages, that can benefit from the efficient use of limited and possibly unlabeled data. In this thesis, we extend the theory of unsupervised techniques to not only address zero resource speech recognition, but also other related applications such as audio event detection and anomaly detection.

1.2 Background

The fundamental components of ASR must be revisited if we are to enhance and extend the technology to settings in which we have very little data.

Figure 1.2 represents the architecture of a typical ASR. Language is highly

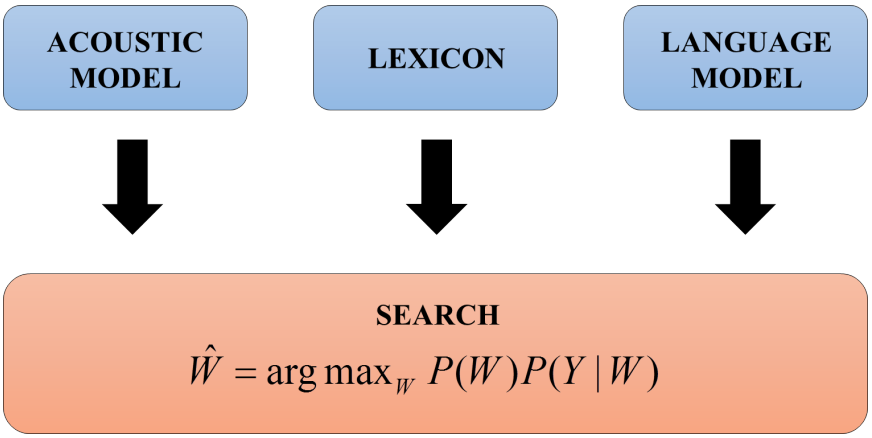

Figure 1.2: Architecture of a typical speech recognition system

structured, as reflected by the architecture: a sentence is made up of words (language model), which is made up of a sequence of sub-word units such as phonemes (lexicon), and the phonemes themselves map to a set of sounds (acoustic model).

1.2.1 Acoustic Model

The acoustic model maps a speech signal into sub-word units such as phones. The mapping can be either deterministic or probabilistic, with the latter being far more popular. Given some acoustic features $Y \ \epsilon \ \mathcal{Y}$ and some symbol $W \ \epsilon \ \mathcal{W}$, the acoustic model estimates $P(Y|W)$. The hidden Markov model (HMM) is the most popular statistical tool for acoustic modeling: it has a history that dates back to the 1960s and is still the *de facto* approach in state-of-the-art systems. Efficient implementation, impressive results, and a natural and intuitive interpretation justify the extensive use of HMMs within the speech recognition community.

HMMs are graphical models that are characterized by a hidden state space and an observation space, as shown in Figure 1.3. In the case of an acoustic model, the hidden states represent symbolic, but acoustically generalizable units such as phones, and the observations are acoustic feature vectors in some fixed length vector space (e.g. 39 mel-frequency cepstral coefficients (MFCCs)). They are fully parameterizable by

$$\lambda = \{P(W_t), P(W_t|W_{t-1}), P(Y_t|W_t)\}$$

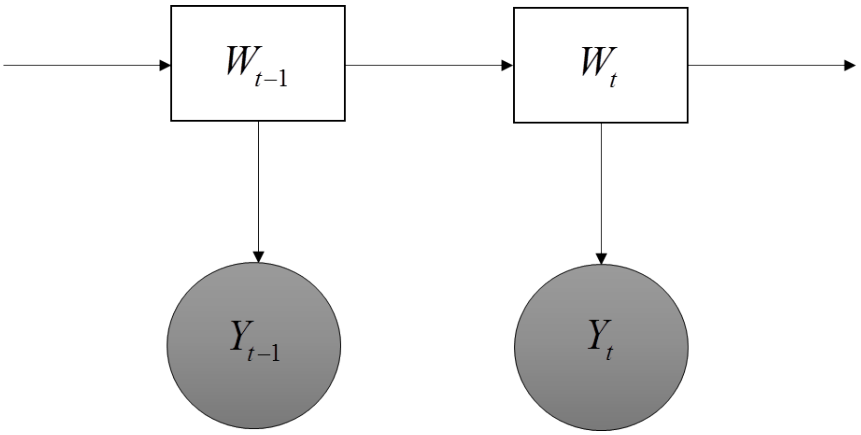

Figure 1.3: Graphical model representation of a HMM

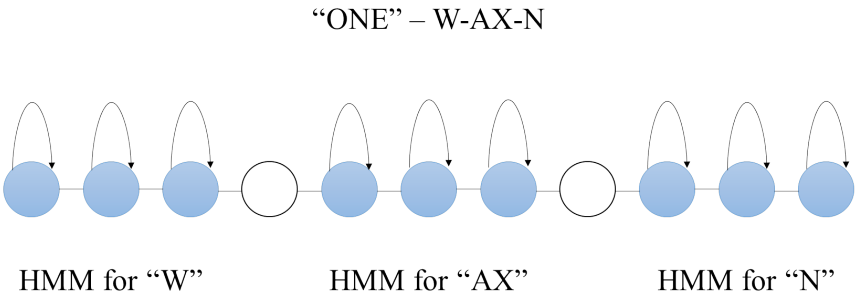

Figure 1.4: HMM representation of the word "ONE"

where $P(W_t)$ is a prior on the state space, $P(W_t|W_{t-1})$ is the probability of transitioning from one state to another, and $P(Y_t|W_t)$ is the probability of observing the acoustic vector Y_t given that the hidden state (phone or triphone) is W_t.

We *train* an acoustic model by estimating λ, the parameters of a HMM. Since Y_t is itself continuous, additional assumptions on $P(Y_t|W_t)$ are required; in most cases, $P(Y_t|W_t)$ is modeled with a mixture of Gaussians. In standard systems, a separate HMM is trained for each phone, and the individual phone HMMs are stitched together to represent a particular word. Figure 1.4 illustrates this for the word "ONE." The success of an acoustic model therefore depends heavily on both the quantity and quality of speech data and their transcriptions.

1.2.2 Language Model

The language modeling literature is as vast as all of speech recognition since it serves many other technologies such as information retrieval and machine translation. We focus on statistical language models (SLMs), which specify a prior on language; the prior captures structure in natural language by learning a distribution over all possible sequences of words (phrases, sentences, etc.). Given a sentence $W = w_1 w_2 ... w_k$, the goal is to estimate $P(W)$.

This is fundamentally a density estimation problem, and therefore unsupervised. However, most natural languages contain large vocabularies, and a good SLM requires enormous amounts of training data. For languages such as English, text corpora are readily available online at sources like the Wall Street Journal, Wikipedia, etc. In other cases, it may be difficult to find written text.

We do not provide a survey of all possible SLM techniques here. In this thesis, we focus on n-grams and their variants, which account for some of the most successful approaches in language modeling. With a Markov assumption, $P(W)$ can be segmented into several joint probabilities over an n-tuple. An n-gram is therefore a histogram that simply counts the number of times a particular word sequence occurs in the training text. It is well understood that there is no such thing as a "large" dataset – there always exist word sequences that are not observed in the training set, but may be present in the test set. Techniques such as clustering, smoothing, and interpolation ameliorate the impact of data sparsity.

1.2.3 Zero Resource Setting

Automatic speech recognition (ASR) is a mature technology when all three components – acoustic model, lexicon, and language model – are well-specified. The purpose of this thesis is to develop rigorous theory and practical algorithms when one or more of these components is missing. In this section, we briefly review existing techniques that cope with the lack of transcribed speech data, and then describe a particularly useful framework that lends itself to the theoretical results presented in subsequent chapters.

Figure 1.5 represents a setting in which there is absolutely no transcribed speech data, no lexicon, and no text to estimate a language model. In fact, we

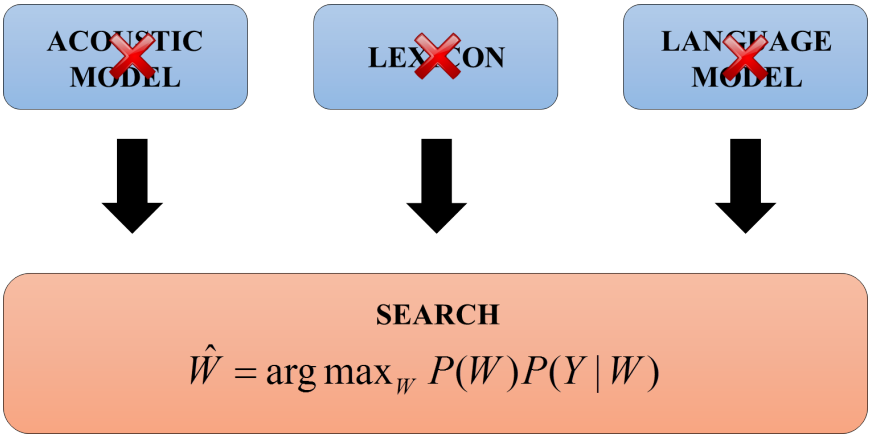

Figure 1.5: A zero resource setting

may not even know what the language is. It is still important to identify recurring acoustic patterns that represent meaningful sentence-like structures. The prevailing approach [2, 3] is based on the acoustic dot plot. A similarity matrix (time-shifted kernel evaluations of the speech signal) is constructed, and key off-diagonal patterns are assumed to be meaningful phrases.

We approach this problem in the context of sequence clustering as it is both intuitive and encapsulates the former approach. The main idea is to cluster a sequence of phones into another sequence of word-like structures; zooming in, we can obtain a sequence of phones by clustering a sequence of acoustic features. The problem as specified, however, is ill-posed; to avoid a trivial solution, we impose the following intuitive constraints:

1. We want the word-like clusters to be as pure as possible.

2. We want the word transitions to be sparse, so that they reflect structure in natural language (e.g. words don't arbitrarily transition between each other).

Rigorous results from PAC-Bayesian theory will later confirm that sparsity is not only intuitive, but also essential for provably minimizing generalization error.

Given some under-resourced language, it may even be possible to find an acoustic model in another language (e.g. English), that can produce a noisy sequence of phones. Of course, the level of noise depends on the phonetic inventory and similarity between the two languages. Nevertheless, the clustering approach described above can be extended to the almost zero

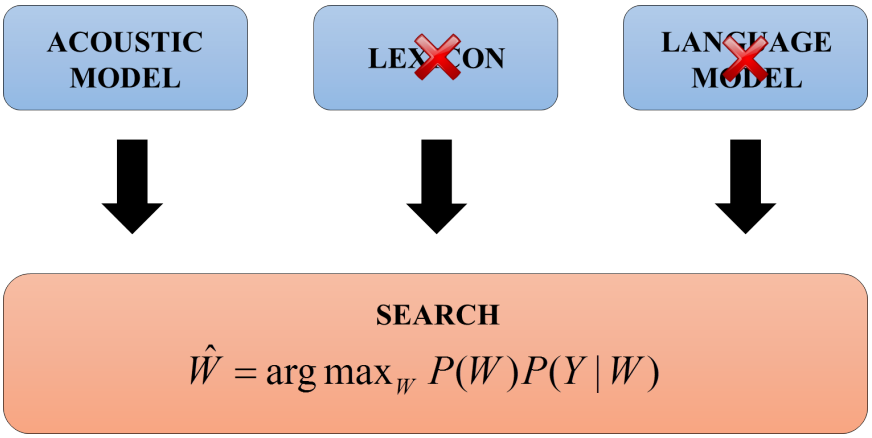

Figure 1.6: An (almost) zero resource setting

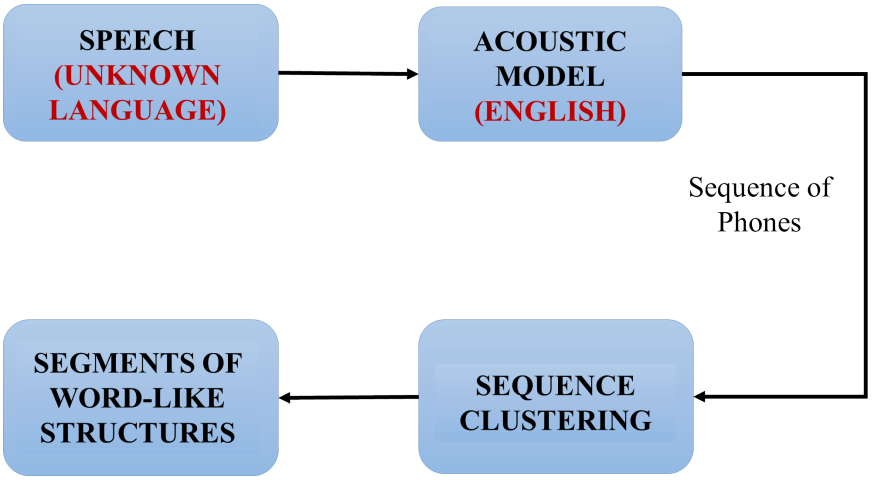

Figure 1.7: A transfer learning architecture

resource setting depicted in Figure 1.6 and the transfer learning approach shown in Figure 1.7.

1.3 Contributions

This thesis introduces PAC-Bayesian analysis to unsupervised problems in audio, speech, and language applications; consequently, we develop novel theory and algorithms that improve upon the state-of-the-art in many settings. Following is a partial list that summarizes the main contributions.

1. Standard PAC-Bayesian bounds are extended to address the sequential nature of speech and language signals.

2. In the case of clustering – the prevailing approach for organizing unlabeled data – a novel regularization technique is shown to provably minimize generalization error.

3. The regularization technique is incorporated into HMM-based sequence clustering algorithms; as a corollary, it is shown, for the first time, that sparsification of any HMM with a Renyi entropy prior minimizes generalization error.

4. The HMM-based sequence clustering algorithm performs remarkably well on tasks such as language modeling and acoustic event detection.

1.4 Organization

The rest of this thesis is organized in two parts. In the first part (Chapters 2 & 3), standard unsupervised approaches to speech recognition, such as clustering, are re-examined within the PAC-Bayesian framework; this provides theoretical guarantees and the insight necessary for a novel HMM-based sequence clustering algorithm. In the second part (Chapters 4-6), applications such as language modeling, acoustic event detection, and anomaly detection are explored as a practical consequence of the theory and algorithms developed in the first part. The following list provides a short description of each chapter.

Chapter 2: PAC-Bayesian Analysis Relevant PAC-Bayesian results for supervised learning problems (e.g. classification) as well as unsupervised learning problems (e.g. clustering and density estimation).

Chapter 3: Sequence Clustering PAC-Bayesian bounds for clustering are extended to sequences. A novel HMM-based sequence clustering algorithm that directly minimizes the bound is introduced.

Chapter 4: Class-Based Language Models The bounds are specialized to the perplexity of a language model, and the clustering algorithm is tested on the resource management corpus.

Chapter 5: Acoustic Event Detection The algorithm is used effectively for clustering non-speech audio into meaningful acoustic events. It is also shown, within the PAC-Bayesian framework, that the algorithm directly maximizes the purity of a cluster.

Chapter 6: Anomaly Detection Additional theory and results are developed for unsupervised sequence anomaly detection.

Chapter 7: Discussion An example of an entire end-to-end unsupervised system for recognizing words in Gujarati.

Chapter 8: Conclusion Summary of key results and possible extensions to other models (e.g. nonparametric) and applications (e.g. mismatch between training and test sets).

CHAPTER 2

PAC-BAYESIAN ANALYSIS

2.1 Introduction

PAC-Bayesian theory is a useful framework for combining frequentist bounds with the notion of a prior. Probably approximately correct (PAC) learning bounds the worst case generalization error of the best hypothesis selected from a hypothesis space – and therefore treats all hypotheses uniformly [4]. PAC-Bayesian bounds, however, place a prior over the hypothesis space while making no explicit assumptions on the data generating distribution [5]. Thus, PAC-Bayesian bounds can both 1) incorporate prior information, and 2) provide frequentist guarantees on the expected performance. They have been successfully applied to classification settings such as the support vector machine (SVM) [6, 7], yielding significantly tighter bounds. Seldin and Tishby [8] extend the framework to include unsupervised learning tasks such as density estimation and clustering.

Given some feature space $\mathcal{X}$, a label space $\mathcal{Y}$, we denote $h : \mathcal{X} \to \mathcal{Y}$ as a hypothesis $h(x)$ on sample x. We assume $h \,\epsilon\, \mathcal{H}$, where $\mathcal{H}$ is the hypothesis space.

For $y, y' \,\epsilon\, \mathcal{Y}$, we define a loss function $l(y, y')$. In the case of classification, this is usually the 0-1 loss, quadratic loss, hinge loss (SVM), etc. For an unsupervised learning task such as density estimation, less intuitive metrics such as cross entropy can be used. Similar to PAC learning, we can define a true loss $L(h)$ and the empirical loss $\hat{L}(h)$ for a hypothesis h.

$$L(h) = \mathbb{E}_{(x,y)}[l(y, h(x))]$$

$$\hat{L}(h) = \frac{1}{N} \sum_{i=1}^{N} l(y_i, h(x_i))$$

By defining a distribution $\mathcal{Q}(h)$ over the hypothesis space $\mathcal{H}$, PAC-Bayesian analysis allows for a second level of averaging. With some notational overload, we can refer to $\mathcal{Q}$ as a random predictor that satisfies the following process:

- Draw $h \, \epsilon \, \mathcal{H}$ according to $\mathcal{Q}(h)$.
- Observe a new sample, x.
- Return $h(x)$.

We can again define loss functions over $\mathcal{Q}$:

$$L(\mathcal{Q}) = \mathbb{E}_{h\sim\mathcal{Q}}[L(h)]$$

$$\hat{L}(\mathcal{Q}) = \mathbb{E}_{h\sim\mathcal{Q}}[\hat{L}(h)]$$

The goal of PAC-Bayesian analysis is to provide guarantees on the difference between the true loss ($L(\mathcal{Q})$) and the empirical loss ($\hat{L}(\mathcal{Q})$) as a function of the number of samples N and the model parameters defined by the hypothesis space $\mathcal{H}$.

The Change of Measure Inequality (CMI) [8] is central to almost every PAC-Bayesian bound, so we briefly state it here. For any measurable function $\phi(h)$ on $\mathcal{H}$ and for any distributions $\mathcal{Q}(h)$ and $\mathcal{P}(h)$:

$$\mathbb{E}_{\mathcal{Q}(h)}[\phi(h)] \leq \mathbb{KL}(\mathcal{Q}||\mathcal{P}) + \ln \mathbb{E}_{\mathcal{P}(h)}\left[e^{\phi(h)}\right] \tag{2.1}$$

where

$$\mathbb{KL}(\mathcal{Q}||\mathcal{P}) = \mathbb{E}_{\mathcal{Q}(h)}\left[\ln \frac{\mathcal{Q}(h)}{\mathcal{P}(h)}\right]$$

is the KL-divergence between $\mathcal{Q}$ and $\mathcal{P}$.

Proof Sketch The proof is surprisingly straightforward.

$$\mathbb{E}_{\mathcal{Q}(h)}[\phi(h)] = \mathbb{E}_{\mathcal{Q}(h)} \ln \left(\frac{\mathcal{Q}(h)}{\mathcal{P}(h)} \frac{\mathcal{P}(h)}{\mathcal{Q}(h)} e^{\phi(h)}\right)$$

$$= \mathbb{E}_{\mathcal{Q}(h)} \ln \left(\frac{\mathcal{Q}(h)}{\mathcal{P}(h)}\right) + \mathbb{E}_{\mathcal{Q}(h)} \ln \left(\frac{\mathcal{P}(h)}{\mathcal{Q}(h)} e^{\phi(h)}\right)$$

and by the definition of KL-divergence and Jensen's inequality

$$\leq \mathbb{KL}(\mathcal{Q}||\mathcal{P}) + \ln \mathbb{E}_{\mathcal{Q}(h)}\left[\frac{\mathcal{P}(h)}{\mathcal{Q}(h)} e^{\phi(h)}\right]$$

The second distribution, $\mathcal{P}(h)$, is usually referred to as a "prior" in the PAC-Bayesian literature. Note that $\mathcal{P}$ is not a prior in the Bayesian sense:

- It indicates preference on the structure of the hypothesis, not an assumption on the data generating distribution, although the latter could be a consequence of the former.
- The inequality holds regardless of $\mathcal{P}$.
- The inequality holds regardless of $\mathcal{Q}$, which is not necessarily the Bayes posterior.

2.2 Classification

We can measure the performance of any classification task with a loss function $l(y, y')$. The 0-1 loss, for example, is the most intuitive one, as it simply counts the number of misclassifications. It is trivial to measure performance on a labeled training set, but the ultimate goal of any prediction problem is to also ensure that the algorithm works well on a previously unobserved test set. PAC-Bayesian analysis provides provable guarantees on the performance of an algorithm/a class of algorithms on test data. In this section, we present one such bound for classification.

Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where (x_i, y_i) are independent and identically distributed, we learn a random classifier $\mathcal{Q}$, which is a distribution over classifiers in some hypothesis space $\mathcal{H}$. Unlike standard PAC analysis, we can also select a prior $\mathcal{P}$ that allows us to favor certain classifiers over others. For example, if $\mathcal{H}$ is defined to be the space of all polynomials, $\mathcal{P}$ can be selected to favor polynomials with degree 1. The following bound for classification is an immediate consequence of CMI in Equation (2.1).

PAC-Bayes-Hoeffding Inequality [8] Assume that $l(y, y')$ is bounded, and fix a prior $\mathcal{P}$ over $\mathcal{H}$. Then, for any $\delta \in (0, 1)$, with probability greater than $1 - \delta$, for all random classifiers $\mathcal{Q}$:

$$L(\mathcal{Q}) \leq \hat{L}(\mathcal{Q}) + \sqrt{\frac{\mathbb{KL}(\mathcal{Q}||\mathcal{P}) + \ln \frac{1}{\delta}}{2N}} \tag{2.2}$$

In words, the test set error of *any* random classifier ($L(\mathcal{Q})$) is bounded by

its training error ($\hat{L}(\mathcal{Q})$) plus an additional term that depends on the model parameters ($\mathbb{KL}(\mathcal{Q}||\mathcal{P})$), confidence level ($\delta$), and the size of the training set (N). The bound above can be specialized and tightened by selecting an appropriate hypothesis space and prior.

The proof follows from the change of measure inequality [8]. By setting $\phi(h) = \lambda(L(h) - \hat{L}(h))$ in Equation (2.1), we obtain

$$\lambda\left(L(\mathcal{Q}) - \hat{L}(\mathcal{Q})\right) \leq \mathbb{KL}(\mathcal{Q}||\mathcal{P}) + \ln \mathbb{E}_{\mathcal{P}(h)}\left[e^{\lambda\left(L(\mathcal{Q}) - \hat{L}(\mathcal{Q})\right)}\right]$$

and the inequality in Equation (2.2) follows after applying the Markov and Hoeffding inequalities to the RHS above, and optimizing over λ. Similar bounds can be derived for unsupervised learning problems such as density estimation and clustering.

2.3 Density Estimation

In unsupervised learning problems, it is difficult to define straightforward loss functions. Seldin and Tishby show that it is still possible to obtain similar generalization results by selecting a clever space. We reproduce the result below and encourage the interested reader to consult Seldin and Tishby [8] for a thorough proof.

PAC-Bayesian Bound for Density Estimation Let $\mathcal{X}$ be the sample space and $p(X)$ be an unknown distribution over $\mathcal{X}$. Let $\mathcal{H}$ be a hypothesis space, in which $h \,\epsilon\, \mathcal{H}$ is a mapping $h : \mathcal{X} \to \mathcal{Z}$ where $\mathcal{Z}$ is a finite set. We define $p_h(Z) = P\{h(X) = Z\}$ as a distribution over $\mathcal{Z}$ induced by the unknown distribution $p(X)$ and the hypothesis h. Again, we assume that $\mathcal{P}$ is some prior over the hypothesis space. As with the loss function, we can have a second level of averaging over $\mathcal{Q}$: $p_{\mathcal{Q}} = \mathbb{E}_{\mathcal{Q}(h)} p_h(Z)$. Given N i.i.d. samples drawn from $p(X)$, let $\hat{p}(X)$ be the empirical distribution. We can define $\hat{p}_{\mathcal{Q}}(Z) = \mathbb{E}_{\mathcal{Q}(h)} \hat{p}_h(Z)$. With probability greater than $1 - \delta$, for all $\mathcal{Q}$:

$$\mathbb{KL}(\hat{p}_{\mathcal{Q}(Z)}||p_{\mathcal{Q}(Z)}) \leq \frac{\mathbb{KL}(\mathcal{Q}||\mathcal{P}) + (|\mathcal{Z}| - 1)\ln(N+1) - \ln\delta}{N} \tag{2.3}$$

This is a specific realization of the CMI, with $\phi(h) = \lambda \mathbb{KL}(\hat{p}_h(Z)||p_h(Z))$. Although we do not have an explicit bound on the loss function as in the

case of classification, Equation (2.3) still bounds the distance between the true distribution $p(X)$ and the empirical estimate $\hat{p}(X)$ as a function of the confidence (δ) and the number of samples (N).

2.4 Clustering

In this section, we present PAC-Bayesian bounds for clustering as a density estimation problem and briefly discuss the case when labels are available.

2.4.1 Clustered Density Estimation

Given a d-dimensional product space $\mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)}$ and a collection of N samples, S, independent and identically distributed (i.i.d.) according to some unknown distribution $p(x_1, ..., x_d)$ over the product space, we want to estimate $p(x_1, ..., x_d)$ with some model $q(x_1, ..., x_d)$. In the case of clustering (e.g. class-based models), we make the following assumption on $q(x_1, ..., x_d)$ [Note: we make no assumptions on the true distribution $p(x_1, ..., x_d)$]:

$$q(x_1, ..., x_d) = \sum_{c_1,...,c_d} q(c_1, ..., c_d) \prod_{i=1}^{d} q(x_i|c_i) \tag{2.4}$$

where $c_i = h_i(x_i)$ for some clustering function $h_i : \mathcal{X}^{(i)} \mapsto \mathcal{C}^{(i)}$. We refer to them collectively as a clustering function h, $h = \{h_i\}_{i=1}^{d}$; hence

$$h : \mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)} \mapsto \mathcal{C}^{(1)} \times ... \times \mathcal{C}^{(d)}$$

We assume that the original space $\mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)}$ has finite cardinality, with $n_i = |\mathcal{X}^{(i)}|$, and likewise for the clustered space $\mathcal{C}^{(1)} \times ... \times \mathcal{C}^{(d)}$, where $m_i = |\mathcal{C}^{(i)}|$ is the number of clusters. We define the hypothesis space, $\mathcal{H}$, to be the space of all possible clustering functions $h \in \mathcal{H}$.

For $h \in \mathcal{H}$, we define distributions $p_h(c_1, ..., c_d)$ and $\hat{p}_h(c_1, ..., c_d)$ that depend on the unknown true distribution $p(x_1, ..., x_d)$ and the empirical (maximum likelihood) estimate $\hat{p}(x_1, ..., x_d)$.

$$p_h(c_1, ..., c_d) = \sum_{x_1,...,x_d} p(x_1, ..., x_d) \prod_{i=1}^{d} \delta(h_i(x_i) = c_i)$$

$$\hat{p}_h(c_1, ..., c_d) = \sum_{x_1,...,x_d} \hat{p}(x_1, ..., x_d) \prod_{i=1}^{d} \delta(h_i(x_i) = c_i)$$

The delta function, $\delta(arg)$, takes a value of 1 only when arg is true, and 0 otherwise. We can extend to the original space with the model assumption in Equation (2.4) as follows:

$$p_h(x_1, ..., x_d) = \sum_{c_1,...,c_d} p_h(c_1, ..., c_d) \prod_{i=1}^{d} q(x_i|c_i)$$

$$\hat{p}_h(x_1, ..., x_d) = \sum_{c_1,...,c_d} \hat{p}_h(c_1, ..., c_d) \prod_{i=1}^{d} q(x_i|c_i)$$

We can apply the CMI with $\phi(h) = N \cdot \mathbb{KL}(\hat{p}_h(x_1, ..., x_d)||p_h(x_1, ..., x_d))$ and simplify the KL-divergence term by recognizing that:

- The set $\{q(c_i|x_i)\}_{i=1}^{d}$ defines a distribution over all possible clusterings, and hence $\mathcal{Q} = \{q(c_i|x_i)\}_{i=1}^{d}$.
- A specific prior $\mathcal{P}$ can be defined without making any explicit assumptions on the true distribution $p(x_1, ..., x_d)$.

The following prior on $\mathcal{H}$ makes no assumptions on $p(x_1, ..., x_d)$. We present the original prior developed by Seldin and Tishby [8] as well as a more simplified version relevant to the rest of this thesis.

$$\mathcal{P}(h) \geq \frac{1}{\exp\left[\sum_{i=1}^{d} (m_i - 1) \ln n_i + n_i H(hist(h|i))\right]} \tag{2.5}$$

and $hist(h|i) = \{|c_{i1}|, ..., |c_{im_i}|\}$ is the size of each of the m_i clusters in the i^{th} dimension, $H(.)$ is the Shannon entropy.

The prior is based on a combinatorial argument. In order to select a clustering function h_i for some i, we first need to pick the cardinality profile, $hist(h|i)$, for the m_i clusters. With the assumption that each cluster has at least one element, we have $n_i - m_i$ remaining elements that can be assigned to m_i clusters, which can be upper bounded by $n_i^{m_i - 1}$. Next, given a cardinality profile $hist(h|i)$, the number of possible ways that the x_i can be assigned to the clusters is upper bounded by $e^{n_i H(hist(h|i))}$. Please refer to Seldin and Tishby [8] for a detailed proof.

From the above prior, it is possible to bound $KL(\mathcal{Q}||\mathcal{P})$ as follows:

$$KL(\mathcal{Q}||\mathcal{P}) \leq \sum_{i=1}^{d} \left[(m_i - 1)\ln n_i + n_i I(\mathcal{X}_i; \mathcal{C}_i)\right] \tag{2.6}$$

where $I(.;.)$ denotes the mutual information. Within the context of an optimization problem, it is evident that n_i dominates this bound. The mutual information term $I(\mathcal{X}_i; \mathcal{C}_i)$ is bounded from above by $\ln m_i$, since $H(\mathcal{C})$ is bounded by $\ln m_i$. In this thesis, we assume that $n_i >> m_i$ and hence loosen the prior for convenience. We therefore obtain:

$$KL(\mathcal{Q}||\mathcal{P}) \leq \sum_{i=1}^{d} \left[(m_i - 1)\ln n_i + n_i \ln m_i\right] \tag{2.7}$$

Alternatively, we can define an entirely new prior that is based on an even simpler argument:

$$\mathcal{P}(h) \geq \frac{1}{\exp\left[\sum_{i=1}^{d} n_i \ln m_i\right]} \tag{2.8}$$

where $n_i = |\mathcal{X}^{(i)}|$, $m_i = |\mathcal{C}^{(i)}|$, and $m_i^{n_i}$ is simply the number of ways in which n_i elements can be assigned to m_i clusters. We can therefore simplify the bound on $KL(\mathcal{Q}||\mathcal{P})$:

$$KL(\mathcal{Q}||\mathcal{P}) \leq \sum_{i=1}^{d} n_i \ln m_i \tag{2.9}$$

The CMI with $\phi(h) = N \cdot \mathbb{KL}(\hat{p}_h(x_1, ..., x_d)||p_h(x_1, ..., x_d))$, our modified prior, and a few information theoretic results lead to the following bound.

PAC-Bayesian Clustering: For any distribution p over $\mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)}$ and an i.i.d. sample S of size N according to p, with probability at least $1-\delta$, for all distributions of cluster functions $\mathcal{Q} = \{q(c_i|x_i)\}_{i=1}^{d}$, the following holds:

$$\mathbb{KL}(\hat{p}_{\mathcal{Q}}(x_1, ..., x_d)||p_{\mathcal{Q}}(x_1, ..., x_d)) \leq \frac{\sum_{i=1}^{d} n_i \ln m_i + K_1}{N} \tag{2.10}$$

where $K_1 = (M-1)\ln(N+1) + \ln\frac{d+1}{\delta}$, and $M = \prod_{i=1}^{d} m_i$.

It is immediately obvious from the bound above that clustering is a useful tool for density estimation. An empirical estimate of $p(x_1, ..., x_d)$ requires N, the number of training examples, to be on the order of $\prod_{i=1}^{d} n_i$; however, in

the case of the bound above, N only needs to be on the order of $\prod_{i=1}^{d} m_i$, which is much smaller since m_i (the number of clusters) is typically smaller than n_i.

We can assume that in a supervised setting, we have an additional label space $\mathcal{Y}$ and the goal is to estimate the joint probability $p(x_1, ..., x_d, y)$ where $y \; \epsilon \; \mathcal{Y}$. The idea is to replace this with $p(c_1, ..., c_d, y)$, and exactly for the reasons discussed above, a clustered estimate requires fewer examples than the empirical estimate.

CHAPTER 3

SEQUENCE CLUSTERING

3.1 Introduction

Unsupervised clustering – grouping the data into clusters – is often a first step in the organization of unlabeled data, with important speech applications such as speaker diarization [9, 10] and speaker adaptation [11], to name a few. Clustering algorithms are useful if the resulting clusters predict the labels that will eventually be assigned. Performance metrics such as cluster purity, entropy, and accuracy attempt to quantify the usefulness of a given algorithm [12]. While many methods (e.g. k-means and spectral clustering) are known to be effective for producing good clusters, they generally only work well when the datapoints lie in some fixed length vector space [12]. Clustering sequences is much more challenging.

Most popular sequence clustering methods tend to be either model-based or distance-based [13, 14, 15, 16]. Model-based approaches make the assumption that the sequences are generated from K different models, each of which represents a cluster [11, 13]. Distance-based methods rely on computing a similarity/distance metric between the sequences [16, 17]; a closely related approach is to extract relevant features and reduce the problem to that of clustering fixed length vectors [15]. There is, however, significant overlap between the two types of sequence clustering algorithms – a large subset of distance-based methods use generative models for obtaining better proximity measures [15, 16, 17]. In this thesis, we focus on generative models and in particular, the HMM.

The popularity of HMMs, especially for describing time-varying signals, is unquestionable. Within the domain of sequence clustering, HMMs have been successfully used in both model-based and distance-based approaches [13, 14, 15, 16, 17, 18], and are quite natural for speech and audio data [18, 19].

Although they allow us to recover structure from sequences and represent observations of varying lengths, they typically do not favor parsimony. We argue that especially for the problem of clustering, parsimony or sparsity in the observation probabilities is essential.

In this chapter, we first extend PAC-Bayesian bounds to a general sequence clustering setting, independent of the clustering model or algorithm. We then present a specific HMM-based sequence clustering algorithm that provably minimizes generalization error.

3.2 PAC-Bayesian Bound for Sequence Clustering

We motivate the sequence clustering problem with an example from language modeling. Suppose our goal is to estimate the probability of a trigram, for example, "the cat sat." Trivially, we can directly estimate the joint probability $p(the, cat, sat)$. Clustering allows us to reduce the number of required training examples and there are four possible clustered (class-based) models:

1. $p(the, cat, sat) = \sum_c p(c)p(the\ cat\ sat|c)$
2. $p(the, cat, sat) = \sum_{c_1,c_2,c_3} p(c_1, c_2, c_3)p(the|c_1)p(cat|c_2)p(sat|c_3)$
3. $p(the, cat, sat) = \sum_{c_1,c_2} p(c_1, c_2)p(the\ cat|c_1)p(sat|c_2)$
4. $p(the, cat, sat) = \sum_{c_1,c_2} p(c_1, c_2)p(the|c_1)p(cat\ sat|c_2)$

In general, an n-gram has 2^{n-1} possible segmentations, as illustrated in the previous example. Suppose $f \in \mathcal{F}$ is a particular segmentation from the space of all possible segmentations, and we explicitly define it as the following mapping:

$$f : \mathcal{V}^n \mapsto \mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)} \tag{3.1}$$

where $\mathcal{V}$ is some vocabulary, $1 \leq d \leq n$, and f is simply a segmentation that does not modify the joint distribution; that is, $p(v_1, ..., v_n) = p(x_1, ..., x_d)$. If f is fixed *a priori*, we can immediately apply the bounds derived in Equation (2.10) over the segmented space $\mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)}$. This is the case where we decide on a model, such as the standard class-based model ($d = 3$), and simply use it.

Extension to a more general sequence clustering paradigm is straightforward. We modify the hypothesis space $\mathcal{H}$ to not only include all possible clusterings, but also all possible segmentations. The new random prediction $\mathcal{Q}$ over $\mathcal{H}$ works as follows:

- Given an n-gram $(v_1, ..., v_n)$, draw a segmentation $f \in \mathcal{F}$ according to the distribution $\pi = (\pi_1, ..., \pi_{2^{n-1}})$, where the segmentations are indexed by $j = 1, ..., 2^{n-1}$ (the ordering does not matter), and π_j is the probability of drawing segmentation j.
- Pick a clustering as in the random classifier described in Equation (2.10) for the new segmented space.
- Estimate $q(v_1, ..., v_n)$ according to the model described by the previous steps.

The bound, in terms of π, is given below.

PAC-Bayesian Sequence Clustering: For any probability measure p over $\mathcal{V}^n$, and an i.i.d. sample S of size N drawn according to p, with probability $1-\delta$ for all distributions of segmentations π and for all distributions of cluster functions $\mathcal{Q}$:

$$\mathbb{KL}(\hat{p}_{\mathcal{Q}}(v_1, ..., v_d)||p_{\mathcal{Q}}(v_1, ..., v_d)) \leq \sum_{j=1}^{2^{n-1}} \left(\frac{\sum_{i=1}^{d(j)} n_i(j) \ln m_i(j) + K_1(j)}{N} \right) \pi_j \tag{3.2}$$

Note that all terms such as $m_i(j)$, the number of clusters corresponding to the space, their product $M(j)$, and additional term $K_1(j)$now depend on the segmentation j since $X^{(i)}$ and $d(j)$ depend on j.

We can favor certain segmentations (e.g. those that require few training examples), but note that the bound above is true regardless of the distribution over possible segmentations, π. Also, the bound is dominated by $n_i(j)$ since it is polynomial in V for all segmentations except the standard class-based setting where $d(j) = n$. For example, if $d(j) = n - 1$ for some segmentation j, there exists some i such that $n_i(j) = V^2$ and hence represents clusters of bigrams. If $d(j) = n - 2$, there exists some segmentation j, and a space i such that $n_i(j) = V^3$, and so on until $d(j) = 1$, and this is the case of word n-grams where $n_1(j) = V^n$.

When a segmentation j and the number of clusters $m_i(j)$ are fixed, $n_i(j)$ is the only term we can attempt to control – it is also the term that dominates the bound in any non-trivial sequence clustering paradigm.

3.2.1 Bound Minimization

Imposing the restriction $\forall j \ \forall i, \ n_i(j) = V$ is simple, and although it can encourage small-sample generalization, it is not a useful strategy for incorporating the constraint. Since $n_i(j)$ corresponds to the original space $\mathcal{X}^{(i)}$ for a given j, restricting $n_i(j)$ would restrict $\mathcal{X}^{(i)}$ to an *a priori*, fixed set of V elements. To learn the best possible set of V elements, however, we need to minimize the *effective* size of $\mathcal{X}^{(i)}$. For example, suppose we are estimating trigrams over $\mathcal{V}^3$ using the following segmentation: $\mathcal{X}^{(1)} = \mathcal{V}$ and $\mathcal{X}^{(2)} = \mathcal{V}^2$ – i.e. a bigram over clusters of words and clusters of word bigrams. The unconstrained bound is dominated by $\mathcal{X}^{(2)}$. We can restrict the *effective* size of $\mathcal{X}^{(2)}$ by assigning zero probability to the vast majority of its elements, by constraining the hypothesis space to consider only cluster assignment functions $q(x_i|c_i)$ in which $n_2 << V^2$ of the elements have nonzero probability. Thus, every word sequence in $\mathcal{V}^d$ can be generated by the $d = n$ segmentation, but every other segmentation is constrained to generate at most a subset of $\mathcal{V}^d$ with nonzero probability.

We achieve this by imposing the restriction on the random predictor $\mathcal{Q}$. By Bayes rule, $q(c_i|x_i) = \frac{q(x_i|c_i)q(c_i)}{q(x_i)}$ and we can alternatively define $\mathcal{Q}$ as $\mathcal{Q} = \{q(c_i), q(x_i), q(x_i|c_i)\}_{i=1}^{d}$. Our goal is to learn a $\mathcal{Q}$ that minimizes the RHS of Equation (3.2), which as discussed above, leads to constraining n_i. As expected, $q(x_i)$ controls the absolute size of $\mathcal{X}^{(i)}$ and $q(x_i|c_i)$ controls the effective size based on the clustering. The dominant term in all of our bounds is n_i, which results from the second term in the prior defined in Equation (2.8), since it bounds the number of ways in which the n_i items can be assigned to the m_i clusters. Alternatively, we can represent this quantity with an upper bound:

$$\left(\sum_{c_i} \|q(x_i|c_i)\|_0 \right) \ln m_i$$

This is because we can write $q(x_i)$ and n_i as follows:

$$q(x_i) = \sum_{c_i} q(x_i|c_i)q(c_i)$$

$$n_i = \|q(x_i)\|_0$$

We therefore have

$$n_i = \|q(x_i)\|_0 = \|\sum_{c_i} q(x_i|c_i)q(c_i)\|_0$$

By the triangle inequality and scale invariance of the l_0 norm, n_i satisfies the inequality

$$n_i \leq \sum_{c_i} \|q(x_i|c_i)\|_0$$

We therefore limit the upper bound, $\sum_{c_i} \|q(x_i|c_i)\|_0$, by sparsifying $q(x_i|c_i)$ for every cluster c_i.

The Optimization Problem: Given some segmentation, we want to find a random predictor $\mathcal{Q}$ – a class-based model over the fixed segmentation – such that the bound in Equation (3.2) is minimized, which is given by the following optimization problem:

$$\begin{aligned} &\underset{\mathcal{Q}}{\text{maximize}} && J(\mathcal{Q}) \\ &\text{subject to} && \|q(x_i|c_i)\|_0 \leq V,\ \forall\ c_i\ \epsilon\ \mathcal{C}^{(i)}, i = 1, \ldots, d \end{aligned} \tag{3.3}$$

where $J(\mathcal{Q})$ represents a likelihood function based on the model parameters.

Since such optimization problems are known to be NP-complete, we need to use a computationally tractable proxy. The standard practice is to use the l_1 norm instead of the l_0 norm; although non-convex, we resort to the l_α norm, $0 < \alpha < 1$, since $q(x_i|c_i)$ is a probability vector with a fixed l_1 norm. We therefore solve the following problem:

$$\begin{aligned} &\underset{\mathcal{Q}}{\text{maximize}} && J(\mathcal{Q}) \\ &\text{subject to} && \|q(x_i|c_i)\|_\alpha \leq V,\ \forall\ c_i\ \epsilon\ \mathcal{C}^{(i)}, i = 1, \ldots, d \end{aligned} \tag{3.4}$$

We have shown that one way to regularize the bound for a non-trivial sequence clustering problem, regardless of whether the segmentation is fixed

or if we are interpolating across all segmentations, is to sparsify the cluster assignment probabilities for every cluster. There are many ways to sparsify a probability vector [20, 21, 22], and we select the l_α norm, $0 < \alpha < 1$, for its simplicity, success in other applications [23], and a useful interpretation in the Bayesian context.

3.3 An Efficient HMM-Based Algorithm

HMMs [19] are parameterized by $\lambda = (q(c), q(c_i|c_j), q(x|c))$, where $q(c)$ is a distribution over the states, $q(c_i|c_j)$ are the state transition probabilities, and $q(x|c)$ are the observation probabilities. When the observation space is finite, we can denote this with a matrix B, where B_{ij} represents the probability of emitting observation j given state i. We would like to group a set of N sequences, $O = \{O^j\}_{j=1}^N$, into K clusters. We assume that K is known; if K is unknown, we can draw from a rich set of model selection methods to estimate it [24].

HMM-based clustering algorithms make some assumptions about the relationship across the K HMMs, each of which generates the samples that belong to its respective cluster. For example, Smyth makes the following mixture model assumption:

$$f_K(O^i) = \sum_{j=1}^{K} f_j(O^i|\lambda_j)p_j$$

where O^i is the i^{th} sequence, and λ_j is the set of model parameters for the j^{th} HMM $f_j(.)$ [13]. The idea in [13] is to construct the following similarity matrix by training a separate HMM on each of the N sequences:

$$S_{ij}^N = P(O^i|\lambda_j)_{i,j=1\ldots N}$$

Given any such matrix S, it is easy to group the sequences into K clusters using some standard method such as spectral clustering [25]. Smyth then proposes to train K new HMMs (one for each cluster) with its corresponding set of sequences [13]. The mixture model assumption allows us to fuse the K HMMs into one big HMM and train on all N sequences [13]. Mixed approaches do not necessarily focus on learning an overall generative model;

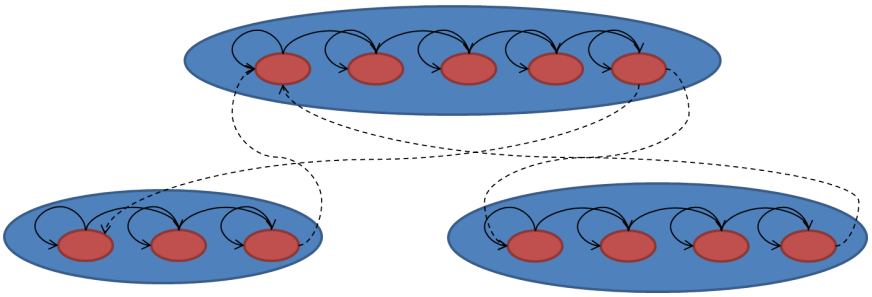

Figure 3.1: HMM topology for 3 sequence clusters

instead, they use S^N or more discriminative estimates of it to directly partition the data into K clusters [15, 16, 17].

In speech and language applications, data are not naturally segmented into N different sequences – initialization as described in [13, 15, 16, 17] is difficult. We therefore allow transitions across the HMMs and train a single super-HMM that automatically segments and clusters the data. Figure 3.1 illustrates a HMM topology for 3 clusters of varying lengths. A similar model was used effectively in [11] for the problem of speaker adaptation. In all of the approaches outlined and cited here, HMMs are trained using the maximum likelihood criterion; in this section, we show how we can incorporate sparsity to directly minimize the PAC-Bayesian bound.

To estimate the joint probability $q(x_1, ..., x_d)$, we previously made the following modeling assumption:

$$q(x_1, ..., x_d) = \sum_{c_1,...,c_d} q(c_1, ..., c_d) \prod_{i=1}^{d} q(x_i|c_i) \tag{3.5}$$

For an HMM, we can make a Markov assumption on $q(c_1, ..., c_n)$:

$$q(x_1, ..., x_d) = \sum_{c_1,...,c_d} \prod_{i=1}^{d} q(x_i|c_i) q(c_i|c_{i-1}) \tag{3.6}$$

where $\{x_i\}_{i=1}^{d}$ is some segmentation of $(v_1, ..., v_n) \ \epsilon \ \mathcal{V}^n$. If we consider each state of the HMM to be a cluster, then $q(c_i|x_i) = q(x_i|c_i)\frac{q(c_i)}{q(x_i)}$ is a distribution over all possible clustering functions. To solve the optimization problem described in Equation (3.4), we need to maximize the likelihood function $J(\mathcal{Q})$ while satisfying the constraint $\|q(x_i|c_i)\|_\alpha \leq V$. We can rewrite the constrained optimization problem as an unconstrained problem using a La-

grangian:

$$\begin{aligned} \underset{\mathcal{Q}}{\text{maximize}} \quad & J(\mathcal{Q}) - \eta \|q(x_i|c_i)\|_\alpha \\ \text{subject to} \quad & \eta \geq 0 \end{aligned} \tag{3.7}$$

and solve for $q(x_i|c_i)$ with an l_α regularized version of the expectation maximization (EM) algorithm, similar to Bharadwaj et al. [26].

3.3.1 MAP Estimation to Encourage Sparsity

A popular approach for learning the HMM parameters is Baum Welch estimation based on the expectation maximization (EM) algorithm [27]. The idea is to iterate between computing the expectation (the Q function) and maximizing it. $Q(\lambda, \lambda')$ is given by

$$Q(\lambda, \lambda') = \sum_{q \epsilon S} \log P(O, q|\lambda) P(O, q|\lambda') \tag{3.8}$$

where S is the space of all state sequences, $O = \{O_t\}_{t=1}^T$ is the observation sequence, and λ' is the previous estimate of the parameters. It is easy to see that $Q(\lambda, \lambda')$ can be written as a sum of functions of the three types of parameters: the initial distribution of states ($q(c)$), the state transition matrix ($A_{ij} = q(c_i|c_j)$), and the matrix of observation probabilities (B) [27]. We can independently optimize over each of the three sets of parameters (at a given iteration). To incorporate prior knowledge/constraints (sparsity or otherwise), we use maximum a posteriori (MAP) estimation. Here, we present the update equations for B with some general prior, $g(B)$. Extension to other sets of parameters ($q(c)$ and A) is straightforward, but not necessary for our problem. We maximize

$$\sum_{i=1}^{N} \sum_{t=1}^{T} \log b_i(O_t) P(O, q_t = i|\lambda') - \eta g(B) \tag{3.9}$$

where $b_i(O_t)$ is the probability that the i^{th} state emits the t^{th} observation in the sequence $\{O_t\}_{t=1}^T$. By setting the gradient to zero and satisfying the

usual constraints that for each i, $\sum_j B_{ij} = 1$ and $B_{ij} \geq 0$, we get

$$B_{ij} = \frac{(\sum_{t=1}^{T} P(O, q_t = i|\lambda')\mathbf{1}\{O_t = j\} - \eta S_{ij})^+}{\sum_{t=1}^{T} P(O, q_t = i|\lambda') - \eta S_{ij})^+} \tag{3.10}$$

where $(x)^+ = \max(x, 0)$, $S_{ij} = B_{ij}\nabla_{B_{ij}} g(B)$, and $\mathbf{1}\{arg\}$ is an indicator function that is 1 if arg is true and 0 otherwise.

Equation (3.10) is a fixed point equation which can be shown to converge to a local optimum whenever $g(B)$ is convex (making $-g(B)$ concave). The overall function (likelihood + prior) can be shown to increase irrespective of how many additional terms we introduce to the likelihood function, as long as they satisfy Jensen's inequality [28].

3.3.2 The Appropriate $g(B)$

Given a vector x in the N-dimensional euclidean space, $\|x\|_\alpha$, the l_α norm of x is given by $\|x\|_\alpha = (\sum_{i=1}^{N} x_i^\alpha)^{\frac{1}{\alpha}}$. The l_2 norm is the most commonly used metric for regularization [24]. The intractable l_0 norm and its relaxation, the l_1 norm (lasso) encourage sparsity [24]. We, however, cannot directly use the l_1 norm since B is a stochastic matrix – the entries are non-negative and each row sums to 1; the l_1 norm of each row is also 1, thus $l_1$regularization is meaningless. A few approaches to sparsifying probability vectors (or simplex) exist [21, 22]; for example, the authors in [21] present a new convex optimization problem that does not use l_1. We use the l_α norm because it can be easily integrated into the Baum Welch algorithm.

Intuitively, the l_α norm for $0 < \alpha < 1$ also encourages sparsity and its use is theoretically justified in [29]. We minimize the sum of the l_α norm of each row of B. In this work, $g(B) = \|B\|_{1,\alpha} = \sum_i (\sum_j B_{ij}^\alpha)^{\frac{1}{\alpha}}$. When $\alpha < 1$, $g(B)$ is not convex and convergence of Equation (3.10) is not guaranteed; however, our experiments demonstrate good convergence properties in practice.

CHAPTER 4

CLASS-BASED LANGUAGE MODELS

4.1 Introduction

The ability to predict unseen events from a few training examples is the holy grail of statistical language modeling (SLM). Although the final test for any language model is its contribution to the performance of a real system, task-independent metrics such as perplexity are popular for evaluating the general quality of a model. Standard algorithms therefore attempt to minimize perplexity on some previously unobserved test set, assumed to be drawn from the same distribution as the training set. This begets the question of how the test set perplexity is related to training set perplexity – every paper on SLM has an answer, with varying levels of theoretical and empirical justification.

The problem of data sparsity and generalization can be traced back to at least as early as Good [30], and possibly Laplace, who recognizes that the maximum likelihood (ML) estimate of event frequencies (n-grams) cannot handle unseen events. Smoothing techniques such as the add-one estimator [31] and the Good-Turing estimator [30] assign a non-zero probability to events that have never been observed in the training set. Recently, Ohannessian and Dahleh [32] strengthened the theory by showing that Good-Turing estimation is consistent when the data generating process is heavy-tailed. In the context of this work, smoothing was perhaps the first attempt to bound generalization error, in that it successfully guarantees a finite test set perplexity.

It is evident that smoothing of the n-gram estimate alone is not sufficient. Techniques that incorporate lower and higher order n-grams, such as Katz [33] smoothing, Jelinek-Mercer [34] interpolation, and Kneser-Ney [35] smoothing, have become standard [36]. Chen and Goodman [37] provide a thorough empirical comparison of smoothing methods and uncover use-

ful relationships between the test set cross-entropy (log perplexity) and the size of the training set, model order, etc. A Bayesian interpretation further explains why some of the techniques (don't) work. Teh [38] discusses fundamental limitations of the Dirichlet process [39] and proposes the hierarchical Pitman-Yor language model as a better way of generating the heavy-tailed (power law) distributions exhibited in natural language.

Instead of directly modeling a heavy-tailed distribution over words, class-based models address data sparsity by estimating n-grams over clusters of words. Intuitively, clustering is a transformation of the event space from the space of word n-grams, in which most events are rare, to the space of class n-grams, which is more densely measured and therefore requires fewer training examples. Brown et al. [40] show that the clustering function that maximizes the training data likelihood must also maximize mutual information between adjacent clusters; although several useful clustering algorithms are based on this principle, no provable guarantees currently exist. Moreover, word transitions are never completely captured by the underlying class transitions, and some tradeoff between accurate estimation of frequent events (word n-grams) and generalization to unseen events (class n-grams) is desired – class-based models are therefore often interpolated with word n-grams using some of the previously described Bayesian methods [36].

Our survey of SLM techniques and their treatment of generalization error has been rather brief and certainly not comprehensive. We focus primarily on n-grams and related models since they have dominated SLM over the last several decades [36], and therefore serve as a good starting point for further analysis. The existing literature suggests that apart from empirical validation and intuition, no provable guarantees exist on the generalization error of language models. Bayesian techniques work well only to the extent the prior assumptions are valid; in this thesis, we present theoretical guarantees that hold irrespective of the correctness of the prior.

Model selection approaches such as the Akaike Information Criterion (AIC) [41] and its variants [42] quantify the tradeoff between complexity and goodness of fit. In the context of a language model, it can be shown that test set cross entropy is approximately the training set cross entropy plus the number of model parameters. Unfortunately, such bounds are loose and do not provide significant algorithmic insight – at best, they recommend the smallest model that works well on the training set. Chen [43] obtained a very accu-

rate relationship for exponential language models by estimating the test set performance with linear regression. Although empirical, his approximation leads to better models based on $l_1 + l_2^2$ regularization. Exponential models are often motivated with the minimum discrimination information (MDI) principle, which roughly states that of all distributions satisfying a particular set of features, the exponential family is the centroid (minimizes distortion relative to the farthest possible true distribution) [44]. This does not bound the generalization error in the manner we wish to, but it is nevertheless a useful property that complements Chen's observations.

In this thesis, we strive for the best of both worlds – we present PAC-Bayesian theory as a powerful tool for deriving high probability guarantees as well as efficient and well-motivated algorithms. We apply the previously described PAC-Bayesian bounds to n-grams, class-based n-grams, and also sequence clustering, where classes represent longer context such as phrases. We show that for sequence clustering, the bound is dominated by the maximum number of sequences represented by each cluster, and consequently requires many more training examples than a class-based model over words. We address this issue by sparsifying the cluster assignment probabilities using the l_α norm, $0 < \alpha < 1$, an effective proxy for the intractable l_0 norm. We validate the theory developed in earlier parts with empirical results on the resource management corpus.

4.2 PAC-Bayesian Bound for Minimizing Perplexity

In applications such as language modeling, we are interested in directly bounding the test set perplexity or cross-entropy. Seldin and Tishby [8] smooth $\hat{p}_{\mathcal{Q}}(x_1, ..., x_d)$ to bound $\mathbb{E}_{p(x_1,...,x_d)}[-\ln \hat{p}_{\mathcal{Q}(x_1,...,x_d)}]$ and provide the following useful result based on Equation (2.10).

Bound on Cross-Entropy: For any probability measure p over $\mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)}$ and an i.i.d. sample S of size N according to p, with probability $1 - \delta$ for all distributions of cluster functions $\mathcal{Q} = \{q(c_i|x_i)\}_{i=1}^{d}$:

$$\mathbb{E}_{p(x_1,...,x_d)}[-\ln \hat{p}_{\mathcal{Q}}(x_1, ..., x_d)] \leq -I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d)) + Eq.(2.10) + K_2 \quad (4.1)$$

where $\hat{p}_{\mathcal{Q}}(x_1, ..., x_d)$ is now the *smoothed* empirical estimate induced by $\mathcal{Q}$

and $I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d))$ is the multi-information given by

$$I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d)) = \sum_{i=1}^{d} H(\hat{p}_{\mathcal{Q}}(c_i)) - H(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d))$$

Eq. (2.10) refers to the bound derived in Equation (2.10), and K_2 is an additional term, $K_2 \geq I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d))$, and the bound is non-negative.

4.3 Interpolated Models

Since language modeling is yet another density estimation problem in which we want to minimize the test set perplexity, the bound in Equation (4.1) readily applies to both word n-grams and class-based n-grams. Note that the bounds are on cross-entropy, which is log perplexity, but we use the two terms almost interchangeably. We are now interested in estimating the unknown true distribution $p(v_1, ..., v_n)$ over the space $\mathcal{V}^n$, where $\mathcal{V}$ is some vocabulary consisting of $V = |\mathcal{V}|$ words. The degenerate case, $d = 1$, $\mathcal{X}^{(1)} = \mathcal{V}^n$, is the case of word n-grams and results in a bound that is dominated by $n_1 = |\mathcal{X}^{(1)}| = V^n$. This suggests that the number of training samples, N, must be on the same order as V^n for the bound (and hence the estimate) to be meaningful.

It is also clear why class-based models are favored whenever they work. In this case, $d = n$, $\mathcal{X}^{(i)} = \mathcal{V}$ for all $1 \leq i \leq d$, and the bound in Equation (4.1) reduces to something linear in V (since $\forall i$, $n_i = |\mathcal{X}^{(i)}| = V$). Moreover, the clustering function is the same for all i – that is, word clusters do not depend on the position in the n-gram. Assuming K word clusters, the number of training examples, N, only needs to be on the order of $K^n + nV$, achieving effective small sample generalization especially when $K << V$. In the following subsections, we extend the bound to sequences and present a unique approach to regularize the bound.

4.3.1 Sequence Clustering

We have discussed two extreme cases, namely $d = 1$ and $d = n$, that correspond to word n-grams and class-based n-grams, respectively. In practice,

they are often interpolated to retain the advantages of both, as shown in the following model:

$$q(v_1, ..., v_n) = \gamma q(v_1, ..., v_n) + (1 - \gamma) \sum_{c_1,...,c_n} q(c_1, ..., c_n) \prod_{i=1}^{n} q(v_i|c_i) \quad (4.2)$$

for some $0 < \gamma < 1$. A Bayesian interpretation of the above model is to select between the n-gram and the class-based model with probabilities γ and $1-\gamma$, respectively. In other words, for each n-gram $(v_1, ..., v_n)$, we simply flip an γ-biased coin to decide on one of the two models. We interpolate across the entire spectrum, $1 \leq d \leq n$, instead of just the extreme cases – that is, we capture clusters over not just words, but also sequences of words (phrases). Previous results by Deligne and Bimbot [45], Ries et al. [46], and Justo and Torres [47] indicate that clustering over phrases is practically useful and leads to significant improvements.

We re-examine the "the cat sat" example from before. In the case of $d = 1$, we directly estimate the joint probability $p(the, cat, sat)$. In the standard class-based model, where $d = 3$, we estimate with the model

$$p(the, cat, sat) = \sum_{c_1,c_2,c_3} p(c_1, c_2, c_3)p(the|c_1)p(cat|c_2)p(sat|c_3)$$

The intermediate cases, such as $d = 2$ in this example, are often neglected. The theory we subsequently develop interpolates over all four segmentations, including the missing ones:

$$p(the, cat, sat) = \sum_{c_1,c_2} p(c_1, c_2)p(the\ cat|c_1)p(sat|c_2)$$

as well as

$$p(the, cat, sat) = \sum_{c_1,c_2} p(c_1, c_2)p(the|c_1)p(cat\ sat|c_2)$$

As discussed earlier, an n-gram has 2^{n-1} possible segmentations, as illustrated in the previous example. Suppose $f \in \mathcal{F}$ is a particular segmentation from the space of all possible segmentations, and we explicitly define it as

the following mapping:

$$f : \mathcal{V}^n \mapsto \mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)} \tag{4.3}$$

where $1 \leq d \leq n$ and f is simply a segmentation that does not modify the joint distribution; that is, $p(v_1, ..., v_n) = p(x_1, ..., x_d)$. If f is fixed *a priori*, we can immediately apply the bounds derived in Equation (4.1) over the segmented space $\mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)}$. This is the case where we decide on a model, such as the standard class-based model ($d = n$), and simply use it.

An extension to the case of interpolated models is straightforward. We modify the hypothesis space $\mathcal{H}$ to not only include all possible clusterings, but also all possible segmentations. The new random prediction $\mathcal{Q}$ over $\mathcal{H}$ works as follows: given an n-gram $(v_1, ..., v_n)$, draw a segmentation $f \, \epsilon \, \mathcal{F}$ according to the distribution $\pi = (\pi_1, ..., \pi_{2^{n-1}})$, where the segmentations are indexed by $j = 1, ..., 2^{n-1}$ (the ordering does not matter), and π_j is the probability of drawing segmentation j; pick a clustering as in the random classifier described in Equation (4.1) for the new segmented space; and estimate $q(v_1, ..., v_n)$ according to the model described by the previous steps. The bound, in terms of π, is given below.

PAC-Bayes Language Modeling: For any probability measure p over $\mathcal{V}^n$, and an i.i.d. sample S of size N drawn according to p, with probability $1 - \delta$ for all distributions of segmentations π and for all distributions of cluster functions $\mathcal{Q}$:

$$\mathbb{E}_{p(v_1,...,v_n)}[-\ln \hat{p}_{\mathcal{Q}}(v_1, ..., v_n)] \leq \sum_{j=1}^{2^{n-1}} \left(-I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d)) + Eq.(3.2) + K_2(j)\right) \pi_j \tag{4.4}$$

where $Eq.(3.2)$ refers to the bound in Equation (3.2), and the dependence on j, as before, is due to the additional segmentation process. As discussed earlier, the bound is polynomial in V for all segmentations except the standard class-based setting where $d(j) = n$.

The main difference between Equations (3.2) and (4.4) is that the latter directly bounds the cross-entropy of an interpolated language model. Because of this, we now have a multi-information term in the bound that we need to minimize, in addition to controlling n_i as before.

4.3.2 Bound Minimization

We need to find a random classifier $\mathcal{Q}$ that maximizes the likelihood $J(\mathcal{Q})$ *as well as* the multi-information $I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d))$. We can rewrite our optimization problems as a Lagrangian.

The Optimization Problem: Given some segmentation, we want to find a random predictor $\mathcal{Q}$ – a class-based model over the fixed segmentation – such that the bound in Equation (4.4) is minimized, which is given by the following optimization problems:

$$\begin{aligned} \underset{\mathcal{Q}}{\text{maximize}} \quad & J(\mathcal{Q}) + \eta I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d)) \\ \text{subject to} \quad & \|q(x_i|c_i)\|_0 \leq V, \ \forall \ c_i \ \epsilon \ \mathcal{C}^{(i)}, i = 1, \ldots, d \end{aligned} \tag{4.5}$$

$$\begin{aligned} \underset{\mathcal{Q}}{\text{maximize}} \quad & J(\mathcal{Q}) + \eta I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d)) \\ \text{subject to} \quad & \|q(x_i|c_i)\|_\alpha \leq V, \ \forall \ c_i \ \epsilon \ \mathcal{C}^{(i)}, i = 1, \ldots, d \end{aligned} \tag{4.6}$$

Within the context of HMMs, we find a clever trick to maximize the multi-information term $I(\hat{p}_{\mathcal{Q}}(c_1, ..., c_d))$. Intuitively, sparsifying the state transition probabilities $q(c_i|c_{i-1})$ should achieve this. This provably works when we use l_α regularization, $0 < \alpha < 1$ for sparsifying $q(c_i|c_{i-1})$. The Renyi α-entropy of a random variable with some probability distribution q is defined to be

$$H_\alpha(q) = \frac{\alpha}{1-\alpha} \log \|q\|_\alpha$$

and there is a useful result we use [48]

$$\lim_{\alpha \to 1} H_\alpha(q) = H(q)$$

where $H(q)$ is the Shannon entropy. This, coupled with the fact that $H_\alpha(q)$ is non-increasing in α ensures that multi-information is maximized. For $\alpha < 1$, $H_\alpha(q)$ is an upper bound on the Shannon entropy. Since l_α regularization minimizes the Renyi α-entropy, which for $0 < \alpha < 1$ is an upper bound on the Shannon entropy, it effectively maximizes the mutual information between c_i and c_{i-1}, given that

$$I(\hat{q}_{\mathcal{Q}}(c_i, c_{i-1})) = H(\hat{q}_{\mathcal{Q}}(c_i)) - H(\hat{q}_{\mathcal{Q}}(c_i|c_{i-1}))$$

We can again reduce Equation (4.7) to an unconstrained optimization problem using the Lagrangian:

$$\begin{aligned} \underset{\mathcal{Q}}{\text{maximize}} \quad & J(\mathcal{Q}) - \eta_1 \|q(x_i|c_i)\|_{\alpha_1} - \eta_2 \|q(c_i|c_j)\|_{\alpha_2} \\ \text{subject to} \quad & 0 < \alpha_1, \alpha_2 < 1, \quad \eta_1, \eta_2 \geq 0 \end{aligned} \tag{4.7}$$

Thus, we have shown that at least in the context of clustering, sparsifying both the observation probabilities and the state transition probabilities of an HMM using the l_α prior directly minimizes generalization error.

A Bayesian interpretation of our regularization provides additional insight into other successful models, such as the hierarchical Pitman-Yor language model (HPYLM). In our approach, we impose the restriction $\|q(x_i|c_i)\|_\alpha \leq V$, $0 < \alpha < 1$, for every cluster c_i. It can be shown that this is equivalent to a sub-exponential prior on $q(x_i|c_i)$ [24]. Since $q(x_i) = \sum_{c_i} q(x_i|c_i)q(c_i)$ and we make the assumption that $q(x_i|c_i)$ is sub-exponential for every c_i, we are consequently assuming that $q(x_i)$ is also sub-exponential. Although PAC-Bayesian bounds hold regardless of the true distribution, our regularization technique implicitly assumes that it is heavy-tailed.

The key to HPYLM's success within the Bayesian setting is a better prior that matches the heavy-tailed distribution of natural language [38] – the regularization approach developed in this thesis reassuringly corresponds to the assumption that the true distribution is heavy-tailed (Figure 4.1 illustrates this for various values of α). On the other hand, it may be possible to derive provable guarantees for HPYLM within the context of our clustering model. The main difference between HPYLM and the less successful Dirichlet process (DP) is the Chinese restaurant process, which assigns new tables (clusters) to customers (samples) much more aggressively in the former model than in the latter [38]. HPYLM therefore has far fewer customers (samples) per table (cluster) than DP, resulting in significantly sparser $q(x_i|c_i)$.

4.3.3 Alternate Justification

We can confirm that our regularization technique for minimizing the PAC-Bayesian bound is well-grounded – at least for the case of class-based language modeling using HMMs – with an alternate analysis.

The standard class-based model – in which word transitions are completely

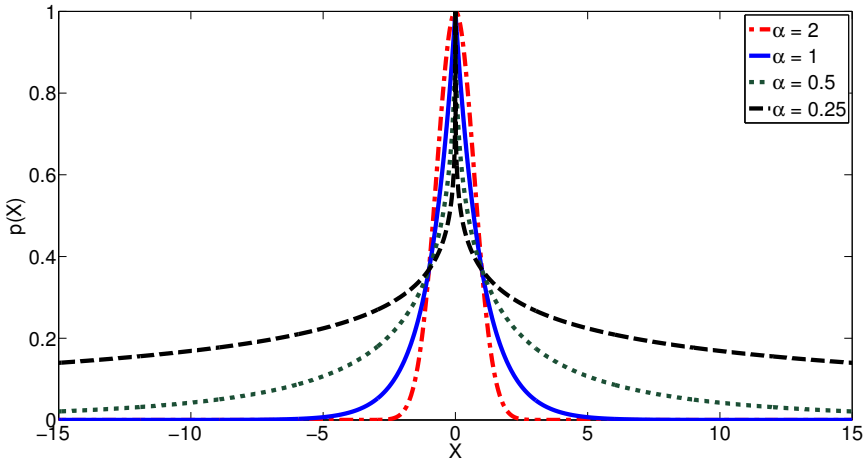

Figure 4.1: $p(X) \propto e^{-\|X\|_\alpha}$ for various α

governed by the underlying class transitions – is motivated with the intuition that words are often related with respect to their use in sentences. When we have limited data and cannot possibly observe all realizations of a particular structure, we resort to counting occurrences of the underlying classes instead, and hope that the estimates generalize to similar, but unseen, events [40]. If the class model is second order (bigram), we can write the probability of a sentence $w_1 w_2 ... w_N$ as

$$P(w_1, w_2, ..., w_N) = \prod_{i=1}^{N} P(w_i|c_i)P(c_i|c_{i-1}) \tag{4.8}$$

where c_j is the cluster assigned to word w_j. This is equivalent to the probability of observing $\{w_1 w_2 ... w_N\}$ given the sequence of hidden states $\{c_1 c_2 ... c_N\}$ in a HMM where $P(w_i|c_i)$ are the observation probabilities and $P(c_i|c_{i-1})$ are the state transition probabilities [19]. A natural extension is to consider classes that also represent word sequences. When the number of classes is small, sequences provide the added benefit of modeling phrases that would otherwise not be well-represented by words [45].

Suppose $\{w_1 w_2 ... w_N\}$ is segmented into M subsequences $S = \{s_1 s_2 ... s_M\}$; HMM-based sequence clustering algorithms such as [26] make the following Markov assumption:

$$P(S, C) = \prod_{i=1}^{M} P(s_i|c_i)P(c_i|c_{i-1}) \tag{4.9}$$

where c_i, the unknown true cluster corresponding to s_i, is drawn from a set of clusters with unknown cardinality, and $C = \{c_1 c_2 ... c_M\}$. When it is further assumed that the c_i are HMMs – that is, the sequences in each cluster are generated by some HMM – Equation (4.9) can be reduced to a single HMM with some structural restrictions on the larger state transition matrix. Such an HMM can automatically segment and cluster the training sequences; Bharadwaj et al. further show that sparsifying the observation probabilities leads to purer clusters [26]. In [26], a sparsifying prior (e.g. Renyi entropy) was used to learn maximum a posteriori (MAP) estimates of the HMM parameters. We adopt this technique to cluster word sequences for a language model.

However, purity of the clusters alone does not imply good generalization in a language model; in the following subsection, we show that to guarantee low perplexity, we need to further minimize the Renyi entropy of transitions between clusters.

Minimizing Perplexity

Define $\pi_i \in \{\Lambda_1, \ldots, \Lambda_K\}$ to be the estimated cluster of sequence s_i, drawn from a set of K HMMs; thus, π_i is an estimate of c_i. Our goal is to find a clustering function $\pi(.)$, $\pi_i = \pi(s_i)$, that minimizes the perplexity per sentence of our model: $L(\pi) = -\frac{1}{M} \log P(S, \Pi)$, where $P(S, \Pi)$ denotes the probability of S and the estimated cluster alignment Π, $\Pi = \{\pi_1 \pi_2 ... \pi_M\}$. We use the same notation as Brown [40], but $\pi(.)$ here is more general than their partition function – it maps sequences of arbitrary length, and does not have to be deterministic. Such a function clearly exists; for example, $\pi(s_i) = \arg\max_{\Lambda \epsilon \ \{\Lambda_1, \ldots, \Lambda_K\}} P(s_i|\Lambda)$ does the trick.

We conveniently rewrite $L(\pi)$ as $L(\pi) = L_A(\pi) + L_B(\pi)$, where

$$L_A(\pi) = -\frac{1}{M} \sum_{i=1}^{M} \log P(\pi_i|\pi_{i-1}) \tag{4.10}$$

$$L_B(\pi) = -\frac{1}{M} \sum_{i=1}^{M} \log P(s_i|\pi_i) \tag{4.11}$$

Let $N_\pi(k, l)$ denote the number of sentences that transition from cluster Λ_k to Λ_l, according to the estimated state alignment Π; likewise, we can define

$N_\pi(k)$ to denote the number of sentences that are assigned to cluster Λ_k. The subscript π is used to emphasize the dependence of these quantities on the clustering algorithm $\pi(.)$. We introduce $P_\pi(l|k) = \frac{N_\pi(k,l)}{N_\pi(k)}$ and $P_\pi(k) = \frac{N_\pi(k)}{M}$ – the ML estimates of the conditional and marginal probabilities, respectively. $L_A(\pi)$ is then an average (weighted by $P_\pi(k)$) of the cross entropy between our estimate $P_\pi(l|k)$ and the true transition probability $P(l|k)$.

$$L_A(\pi) = \sum_{k=1}^{K} P_\pi(k) \sum_{l=1}^{K} -P_\pi(l|k) \log P(l|k) \tag{4.12}$$

$$L_A(\pi) = \sum_{k=1}^{K} P_\pi(k) \left(H_\pi(l|k) + D_\pi(l|k)\right) \tag{4.13}$$

where $H_\pi(l|k)$ is the conditional entropy of the estimate $P_\pi(l|k)$, and $D_\pi(l|k) = D_{KL}(P_\pi(l|k)||P(l|k))$ denotes the KL-divergence between our estimate $P_\pi(l|k)$ and the true distribution $P(l|k)$.

Clearly, minimizing the conditional entropy $H_\pi(l|k)$ for each k minimizes $L_A(\pi)$; the KL-divergence term depends on the true distribution and can be reduced by selecting an appropriate prior for our estimator. The analysis thus far has been independent of the nature of $\{\Lambda_k\}$ and trivially holds for clusters that represent words. In fact, the Brown algorithm [40], in which the mutual information between adjacent clusters Λ_l and Λ_k is maximized, is based on similar analysis. Note that $I_\pi(l;k) = H_\pi(l) - H_\pi(l|k)$. We minimize $H_\pi(l|k)$, which consequently also maximizes $I_\pi(l;k)$; but the converse is not true – maximizing the mutual information can sometimes maximize only the entropy of a cluster $H_\pi(l)$, which does not necessarily help in minimizing perplexity.

If we now consider $\{\Lambda_k\}$ to be HMMs, we can construct a single HMM with T hidden states $Q = \{q_1 ... q_T\}$ that emit O words $W = \{w_1 ... w_O\}$. By defining $N_\pi^{ob}(w,q)$ to be the number of times $\pi(.)$ assigns word w to state q, $N_\pi^{tr}(r,q)$ to be the number of times q transitions to r, and $N_\pi(q)$ to be the number of words assigned to q, we can introduce their counterparts $P_\pi^{ob}(w|q)$, $P_\pi^{tr}(r|q)$, and $P_\pi(q)$. $L_B(\pi)$ reduces to

$$L_B(\pi) = \sum_{t=1}^{T} P_\pi(q_t) \left(H_\pi^{ob}(w|q_t) + H_\pi^{tr}(q|q_t) + D_\pi\right) \tag{4.14}$$

where $H_\pi^{ob}(w|q_t)$ is the conditional entropy of the observation probabilities given state q_t, $H_\pi^{tr}(q|q_t)$ is the conditional entropy of the transition probabilities, and D_π simply refers to the KL-divergence between the estimated probabilities, P_π^{ob} and P_π^{tr}, and their corresponding unknown true values P^{ob} and P^{tr}. It is now clear that minimizing the conditional entropies $H_\pi^{ob}(w|q_t)$ and $H_\pi^{tr}(q|q_t)$ for each t minimizes $L_B(\pi)$. Our strategy for minimizing perplexity is to therefore minimize entropy in both the observation and the state transition probabilities of a suitable HMM, and sparsity clearly achieves this.

4.4 Experiments

We test our approach on a subset of the resource management (RM) corpus [49], which consists of naval commands that span approximately $V = 1000$ words. First, we show that l_α regularization works. Figure 4.2 shows the estimated test set cross-entropy of an unregularized HMM and of an l_α-regularized HMM as a function of the number of training sentences. We vary the training set size from 10 to 2000 sentences and test the models on 800 sentences; Figure 4.2 reports the average cross-entropy on brackets of training sizes – 10-100, 110-200, and so on. The l_α-regularized HMM requires additional tunable parameters such as the value of α. To simplify the search on a separate 300 sentence development set, we make a (rather restrictive) assumption that α for both the transition and observation probabilities is the same, and that α is independent of the size of the training set. Our solutions are therefore not optimal, but adequate to demonstrate our claims. To ensure that the cross-entropy is bounded, we smooth all estimates with add-one smoothing. For small training datasets, the unregularized HMM learns models that assign near-zero likelihood to some of the test sentences; hence, we only present results for training set sizes greater than 500 sentences.

Like many other model selection results, Figure 4.2 suggests that model sparsity is essential when training datasets are small. In this example, about 900 sentences are required for the unregularized HMM to outperform the sparse HMM. In the context of the theory developed in earlier sections, it was shown that test set cross-entropy is proportional to $\frac{n_i}{N}$, where N is the number of training examples. In practical settings, N is fixed; hence, the only strategy for minimizing cross-entropy is to minimize n_i. Figure 4.2

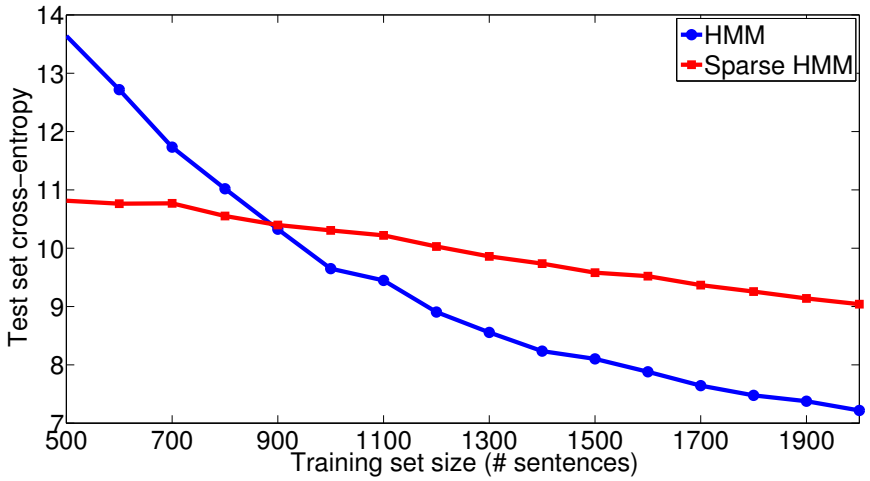

Figure 4.2: Test set cross-entropy of HMM vs l_α-regularized (sparse) HMM as a function of the number of training sentences

confirms that l_α regularization successfully sparsifies $q(x_i|c_i)$, the observation probabilities of the HMM, thereby minimizing n_i.

We also compare how the test set cross-entropy improves as a function of the training set size for four different models: 1) a baseline bigram model estimated over words; 2) a baseline class-based model using Brown's algorithm [40] with $K = 20$ clusters, learned over the entire dataset so that it is also representative of knowledge-based approaches in which the true clusters are known *a priori*; 3) l_α-regularized HMM with 20 ergodic states; and 4) a special case of 3) in which the state transitions are constrained to artificially form $m_1 = 10$ word clusters (10 states) and $m_2 = 5$ clusters that represent word bigrams (10 states, where the 5 clusters are modeled with 2 left-to-right states each); therefore, the model represents an interpolation between the standard class-based model and word bigrams, but is of the exact same complexity as 2) and 3).

Figure 4.3 shows the estimated test set cross-entropy for each of the four models. The values of α used in our experiments are $\alpha = 0.7$ for the words only case and $\alpha = 0.9$ for sequences. It is clear from Figure 4.3 that l_α regularization helps even in the case of a standard class-based model, the bound for which is already linear in V. With fewer than 100 sentences, l_α regularization can both learn the clusters and estimate their transitions reasonably well, and surpasses Brown for training set sizes of $N \geq 800$ sentences. Brown's algorithm in 2) finds clusters such that pairwise mutual information

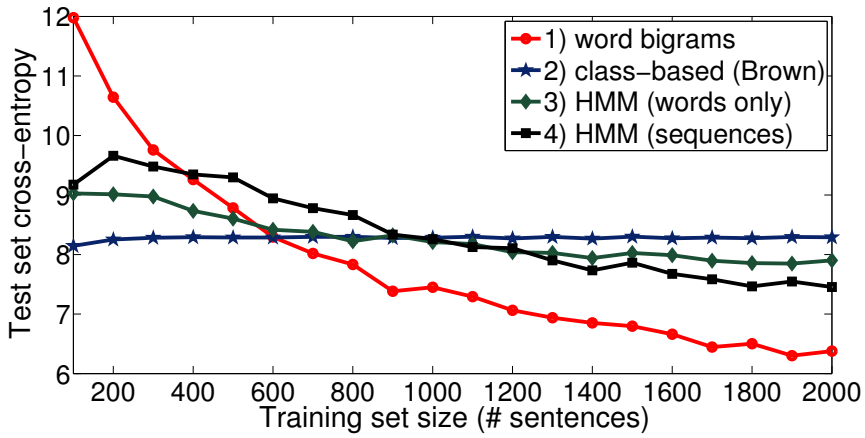

Figure 4.3: Test set cross-entropy as a function of the number of training sentences for the four settings

terms are maximized; in 3), we not only maximize the mutual information, but we also reduce the effective V by ensuring that each cluster (or state) specializes and represents as few words as possible. As the number of training examples increases, estimates of class transitions indeed improve, but the class-based assumption itself becomes too restrictive. In 4), which represents an interpolated model, we see the tradeoff achieved by incorporating sequences: for small training sets, the model achieves better generalization than word bigrams, but is worse than the class-based model; and for larger training sets, the interpolated model learns better representations of high frequency events and outperforms the class-based models represented by 2) and 3).

The value of α in 3) is 0.7, whereas α in 4) is 0.9; this seems counter-intuitive at first, but note that a smaller α does not necessarily imply sparser observation probabilities; however, it implies a heavier distribution in a Bayesian setting. A Bayesian interpretation therefore suggests that in 4), the model itself is better equipped to cope with heavy tails, whereas a more aggressive α is required in 3).

CHAPTER 5

ACOUSTIC EVENT DETECTION

We treat acoustic event detection as yet another sequence clustering problem, and so all the previously developed bounds and algorithms naturally extend. In this chapter, we describe the intuition and show some experimental results on the BBC sound effects corpus.

Let us consider just one cluster and take purity to be the measure of its goodness. The purity of a cluster C is given by

$$purity(C) = \frac{1}{|C|} \max_i \left(|C|_{class=i} \right)$$

where $|C|_{class=i}$ denotes the number of items of class i in the cluster, and $|C|$ is the total size of the cluster. This definition requires us to have access to ground truth labels. In some applications, however, it is difficult to predefine a fixed number of classes and even more difficult to assign labels to all of the datapoints. In such cases, the majority class associated with any given cluster can be reasonably defined to be the most frequently produced sequence of symbols, after deleting repetitions [9]. Cluster purity can then be defined as the fraction of tokens assigned to a cluster that share the same symbol sequence.

We can maximize purity, as defined above, by minimizing the total number of *different* sequences that belong to a particular cluster. Let us consider a simpler case by making the assumption that the state transition matrix is left-to-right; note that our argument can be extended to more general cases and we make this assumption for the sake of simplicity. This structure allows us to view the observation sequence as a set of symbols emitted by the first state, followed by a set of symbols emitted by the second state, and so on. Encouraging sparsity in the observation probabilities allows us to directly minimize the number of symbols emitted by each state and therefore also reduces the total number of possible observations generated by the HMM.

Consequently, this minimizes n_i in the PAC-Bayesian bounds and encourages small-sample generalization.

We test our method on clustering non-speech audio events from the BBC sound effects corpus [50]. The dataset contains 48 files ranging from 15 seconds to 5 minutes in length. The files consist of common events such as rain, waterfall, gunshot, birds, dog, baby crying, etc. We assume that the events can vary drastically in length; for example, a typical gunshot is much shorter than a baby crying. We hypothesize that there are 35 clusters uniformly distributed across 7 event lengths, ranging from 3 states per HMM to 9 states per HMM. In order to detect multiple events per file, we allow transitions from the last state of one HMM to the first state of another and we refer to the resulting HMM as super-HMM. Viterbi decoding is used to segment each audio file into sequences, and to assign each sequence to one of the 35 cluster HMMs. We discretize the observation space by computing 13 mel-frequency cepstral coefficients (MFCCs) with a window of 250 ms and an overlap of 100 ms over all 48 files, and group them into 70 clusters using the k-means algorithm. Each event can then be approximated by a sequence of integers.

Figure 5.1 shows the observation probability matrices of the super-HMM for two cases: no sparsity (top) and some sparsity encouraged (bottom). The exact choice of the parameters ($\alpha = 0.4, \eta = 0.09$) is arbitrary and simply illustrates that our proposed algorithm indeed sparsifies the observation probabilities.

Although we previously defined majority class to be the most frequently produced sequence of symbols, we report results on the more realistic and practical situation in which there are exactly 48 sound classes, each corresponding to a particular file in the dataset. We report results on frame-wise clustering of the data since it allows for a much easier comparison with k-means clustering. We use two measures of average purity: unweighted and weighted.

If we partition the dataset D into K clusters, $\{C_j\}_{j=1}^{K}$,

$$P_{unweighted} = \frac{1}{K} \sum_{j=1}^{K} purity(C_j)$$

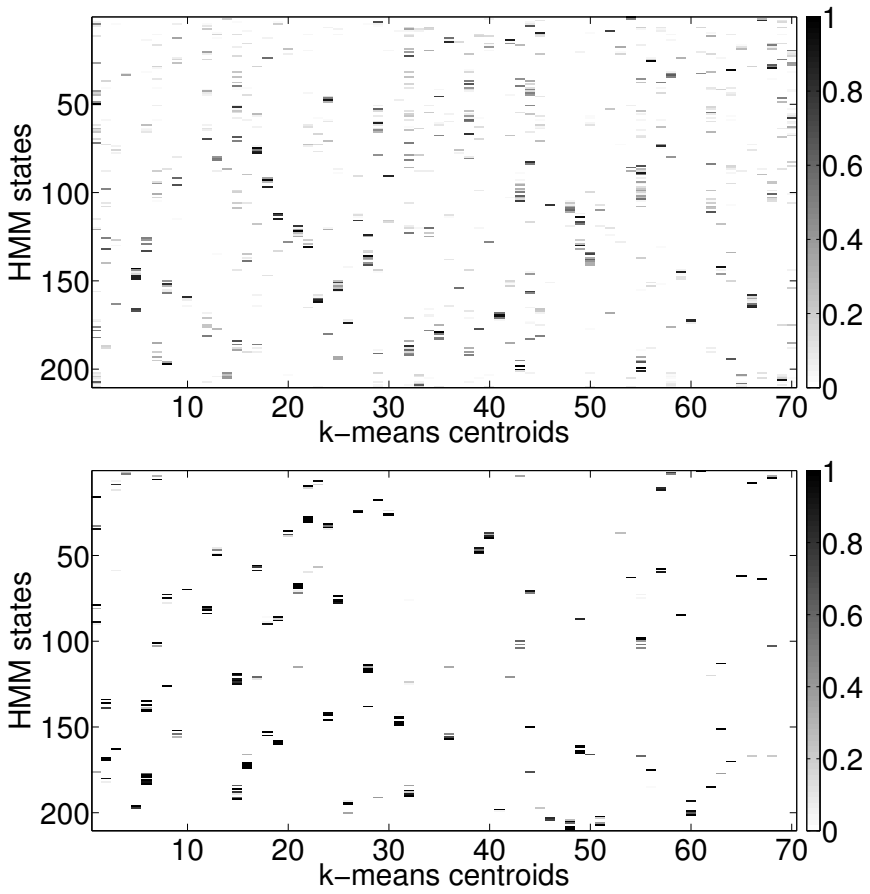

Figure 5.1: Observation matrices B_{ij} (displayed as images with i = row index and j = column index) for $\eta = 0$ (top) and $\alpha = 0.4, \eta = 0.09$ (bottom)

$$P_{weighted} = \sum_{j=1}^{K} \frac{|C_j|}{|D|} purity(C_j)$$

A high value of $P_{unweighted}$ implies that most of the individual clusters are very pure and only a few are impure; $P_{weighted}$, however, also takes into account the number of samples in each cluster – it acts as a check against trivial solutions such as one in which $K-1$ clusters contain one sample each and the K^{th} cluster contains everything else in D.

Figure 5.2 shows the dependence of $P_{unweighted}$ on the regularization parameter η (top) and on α (bottom). It supports our claim (and intuition) that sparsifying the observation probabilities within each HMM purifies the cluster and on average, leads to many more pure clusters. The best values of η (0.05) and α (0.3) indicate that B is neither too sparse nor too dense. It is intuitively clear that when the observation matrix is dense, clusters are bound to be less pure; but why does a little more sparsity lead to relatively less pure clusters? The parameters η and α explicitly control some tradeoff between likelihood and sparsity, and in extreme situations the model is heavily constrained and learning becomes no more than just randomly picking a few (sparse) observations for each state.

Table 5.1 contains the best results for all three methods and the two notions of average purity. The values of (α, η) that maximize $P_{unweighted}$ and $P_{weighted}$ are $(0.3, 0.044)$ and $(0.3, 0.009)$, respectively, which is in line with our intuition – as discussed above, the observation matrix cannot be arbitrarily sparse when trying to maximize $P_{weighted}$. We see that in both cases, sparse HMMs do significantly better than the baseline HMM and k-means. A considerably higher value of $P_{weighted}$ (0.75) especially indicates that when the parameters are chosen appropriately, sparse HMMs do not just focus on a handful of samples and dump the rest into highly impure "garbage" clusters; sparsity is indeed an effective tool for learning purer clusters.

Table 5.1: Purity results

Method	$P_{unweighted}$	$P_{weighted}$
k-means clustering	0.69	0.66
HMM	0.72	0.57
Sparse HMM	**0.88**	**0.75**

These results confirm that l_α-regularized Baum Welch algorithm can be used to learn clusters that are considerably more pure than those obtained

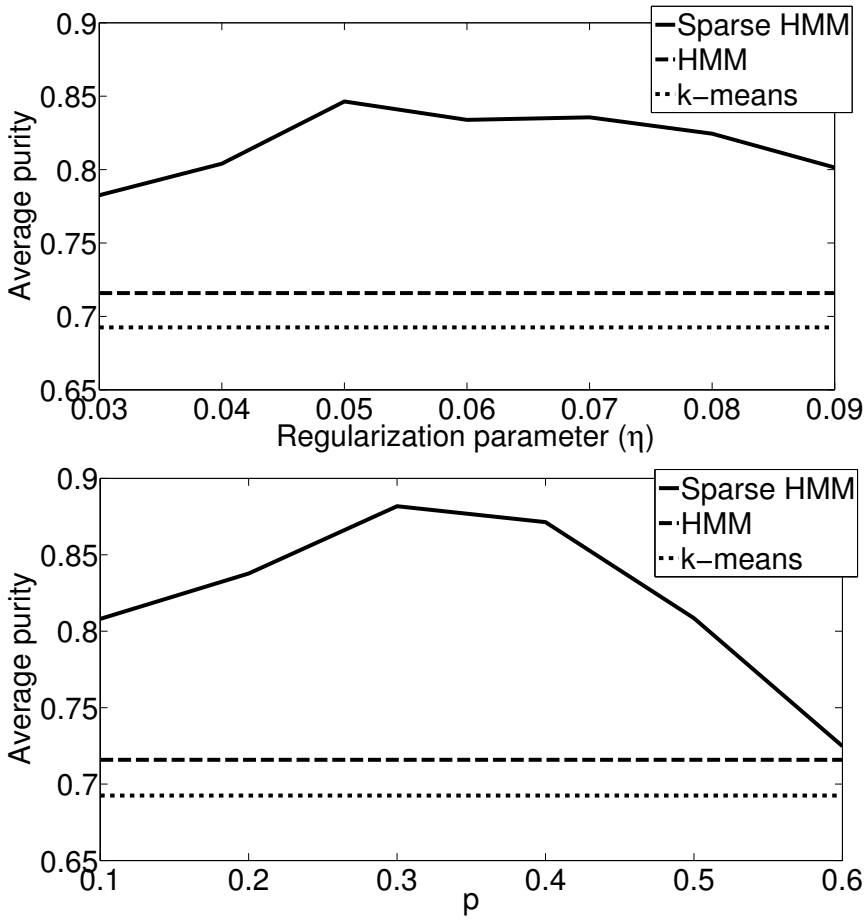

Figure 5.2: Average (unweighted) purity as a function of η with $\alpha = 0.4$ (top) and as a function of α with the best η for each α (bottom)

by standard methods such as the baseline HMM or k-means clustering. Although we restrict our experiments to discrete HMMs in a generative framework, our approach can be extended to more general cases. Methods that use HMM as a tool for learning good distance metrics can also benefit from our algorithm; intuitively, sparse observation probabilities must lead to more discriminative (sparse) similarity matrices and, naturally, to purer clusters. Our interpretation of Renyi α-entropy also provides for an extension to more general HMMs; to maximize purity, we can directly minimize the Renyi α-entropy, $0 < \alpha < 1$, of each state.

CHAPTER 6

ANOMALY DETECTION

6.1 Introduction

Anomaly detection generally refers to a broad class of methods designed to identify unusual events – events that deviate from their normal or expected behavior. Despite this simple definition, there are nearly as many different notions of *anomaly* as there are applications; hence, it is challenging to formulate a unifying theory that is also practically relevant. In this chapter, we adopt the PAC-Bayesian framework to derive efficient algorithms as well as provable bounds in the most general setting, wherein: 1) the algorithms can be fully unsupervised, which allows us to detect anomalies directly from test data; 2) the bounds can be extended to sequential data, in which the anomaly itself can be a sequence of events; and 3) the definition of anomaly is extended to incorporate a hidden class structure, augmenting the usual notion of low probability or rareness of an event. All three extensions have significant value in a number of different application domains (e.g. event detection, speech and natural language processing), and therefore deserve a more rigorous treatment than what is currently available in the anomaly detection literature.

Within the context of learning theory, we can group anomaly detection algorithms into *supervised*, *semi-supervised*, or *unsupervised*. Supervised settings can be reduced to binary classification, where the classes refer to either a *background* event or an *anomalous* event. This allows us to draw from a rich set of existing theory for classification problems. For example, Vapnik-Chervonenkis (VC) theory and probably approximately correct (PAC) learning provide bounds on the generalization error of an algorithm as well as methods for provably minimizing them. It is slightly more challenging to obtain PAC bounds in the semi-supervised case, where we assume that all

of the training examples belong to the *background* class, but still possible for methods like the one-class support vector machine (SVM). The unsupervised setting is most challenging. Although there are many practical approaches to unsupervised anomaly detection (e.g. techniques based on mixture models), there exist almost no guarantees akin to PAC bounds for supervised learning – we show that PAC-Bayesian analysis is especially useful for obtaining provable guarantees in the unsupervised case.

We show that these bounds are also practical. We have already seen that in the case of sequences, generalization error grows multiplicatively as a function of the length of the sequence. This is undesirable for many applications such as acoustic event detection, where long sequences are plausible but we still need to detect anomalies given limited training data. Our main strategy is a form of regularization that constrains generalization error to grow at-most linearly with respect to sequence length. Coupled with the notion that an "anomaly" is not just any rare event, but an event associated with some hidden class that is infrequent, we successfully extract meaningful anomalies from the BBC sound effects corpus, a collection of several non-speech acoustic events.

We treat unsupervised anomaly detection as a sequence clustering problem, where each cluster is highly pure and representative of either an anomalous event or a background event. By using PAC-Bayesian results for sequence clustering, we show that sparsifying the HMM observation probabilities with an ℓ_α prior, $0 < \alpha < 1$, minimizes the false alarm in an anomaly detection problem.

Previously, we introduced PAC-Bayesian bounds for sequences and developed an HMM-based algorithm for minimizing the perplexity of a language model. In this chapter, we extend the theory and algorithms to anomaly detection. Our approach is general and can be applied to any anomaly detection problem, but we present experiments for non-speech acoustic event detection from the BBC sound effects corpus.

Our work is naturally related to a host of anomaly detection techniques outlined in [51, 52], and we highlight deviation from standard methods whenever appropriate. Since we take a sequence clustering approach to anomaly detection, our work is also related to general HMM-based clustering techniques such as [11, 13, 14, 15, 16, 17, 18] as well as sparse HMMs [26, 53, 54].

In the next section, we introduce an HMM-based approach for unsuper-

vised sequence anomaly detection.

6.2 Unsupervised Anomaly Detection

In many real world applications, it is difficult to obtain adequate training data for the anomalous class, making supervised approaches impractical. Likewise, semi-supervised methods also require clean data. For example, if there is an anomaly within the training dataset, a semi-supervised approach would assume it belongs to the background class and fail to detect future occurrences as anomalous. A fully unsupervised approach is therefore essential for detecting anomalies directly within a dataset, without access to separate, clean training data.

Eskin [55] first proposed a mixture model approach for unsupervised anomaly detection. The idea is to make the generative assumption that the data are drawn from $\mathbf{D}$, a mixture of $\mathbf{B}$ and $\mathbf{A}$ – the distributions corresponding to the background data and anomalous data, respectively.

$$\mathbf{D} = \lambda\mathbf{A} + (1 - \lambda)\mathbf{B}$$

where λ is the mixture weight. Since λ determines exactly how rare an anomaly is, it is assumed to be very small. Instead of using the expectation maximization (EM) algorithm to learn $\mathbf{A}$ and $\mathbf{B}$, Eskin exploits the fact that λ is small and uses a more efficient iterative algorithm. Subsequent approaches reviewed in [51, 52] rely on a similar model.

In structurally rich datasets, it is difficult to cluster the data into only two distinct groups – instead, it is much more meaningful to assume that the data are generated by a few *background* clusters and a few *anomaly* clusters. In this work, we make the following more natural assumption on $\mathbf{D}$.

$$\mathbf{D} = \sum_{k=1}^{K} \lambda_k \mathbf{A}_k$$

We can expect λ_k to be small if $\mathbf{A}_k$ is an anomaly cluster, and large when $\mathbf{A}_k$ is a background cluster. When training such a model, we use the EM algorithm with the assumption that $\{\lambda_k\}_{k=1}^{K}$ is sparse.

6.2.1 Sequential Data

There are two different notions of anomaly in sequential data:

1. When a sequence *among* a set of sequences is anomalous.

2. When a subsequence *within* a sequence is anomalous.

In the first scenario, each event is a sequence (as opposed to a fixed length vector), but the events themselves are not temporally related. Unsupervised anomaly detection is therefore equivalent to representing each sequence with some statistical model (e.g. HMM), and clustering the models with a mixture approach as discussed in the previous section. That is, a mixture of HMMs is a reasonable model for this setting.

We focus mostly on the second scenario, where the events transition among one another and are therefore temporally related. We can make a Markov assumption for these transitions, and use a nested HMM of smaller HMMs as shown in Figure 6.1.

6.2.2 HMMs

HMMs have been successfully used in sequence anomaly detection, where the hidden states are generally found to be more stable and expressive in detecting anomalous events [52, 56, 57, 58]. In the case of unsupervised anomaly detection, HMMs can be used as an effective tool for clustering sequences.

As discussed earlier, the key idea is to group a set of N unlabeled sequences into K clusters, with the implicit assumption that clusters are much more representative of the underlying true class. Each cluster is modeled with a separate HMM, and they can be combined to form a single HMM, as shown in Figure 6.1 for $K = 3$. Training a single HMM allows us to automatically segment the sequences; a similar approach was used in [11, 26, 53] for speaker adaptation, event detection, and language modeling, respectively.

When the data are fully unsupervised – i.e. contain examples of both background and anomalous events – the best we can do is to cluster them in some meaningful way. We need to discriminate between highly pure clusters that contain anomalous events from those that contain background events.

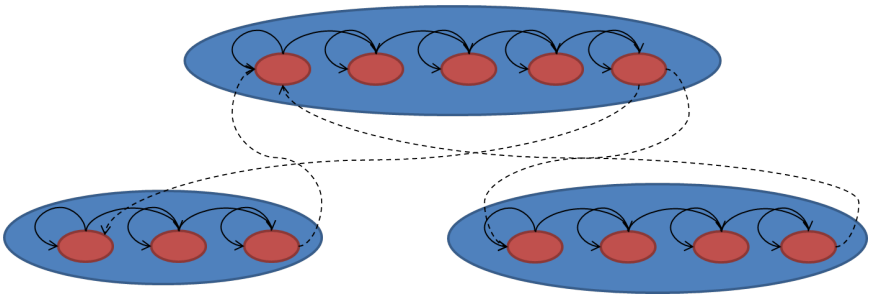

Figure 6.1: HMM-based sequence clustering model

6.3 Theoretical Formulation

Sequential signals such as speech and audio tend to have subsequences that exhibit drastically different characteristics. In many cases, it is possible to ascribe some structure to these anomalies. We examine the setting where there exists an underlying class structure from which the subsequences are drawn. Let us consider a simple example from the BBC sound effects corpus. Figure 6.2 illustrates time-varying spectral bins for six different acoustic events; it is evident that the third class (marked with a *) is drastically different. Unlike the rest of the events, it mostly contains long tones with a few short bursts. Given the true identity of these six events, it would have been easy to isolate the third class (e.g. doorbell sounds a lot different from bird chirping or baby crying). Our definition of anomaly is therefore closely related to the notion of an underlying class structure.

Let us assume that we have some original feature space $\mathcal{X}^{(1)} \times ... \times \mathcal{X}^{(d)}$ from which an acoustic event x is drawn. Our clustering algorithm, $f(.)$, assigns x to one of K clusters, denoted by $\mathcal{C} = \{C_1, ..., C_K\}$; we further assume that x corresponds to one of L sound classes, denoted by $\mathcal{Y} = \{Y_1, ..., Y_L\}$ and use $g(.)$ to refer to the true class. We say that x is an anomaly if it belongs to the class that is least frequently assigned to a background cluster. For some clustering function $f(.)$, we can write average purity as the following expectation:

$$\mathbb{E}_k \left[\max_i P(f(x) = C_k | g(x) = Y_i) \right] \tag{6.1}$$

In the previous section, we described a model in which the clusters can be grouped into anomalous or background events; we use C_A and C_B to denote the sets of clusters that correspond to the two groups, respectively. Similarly, we can define sets of classes Y_A and Y_B. A very simple rule for partitioning

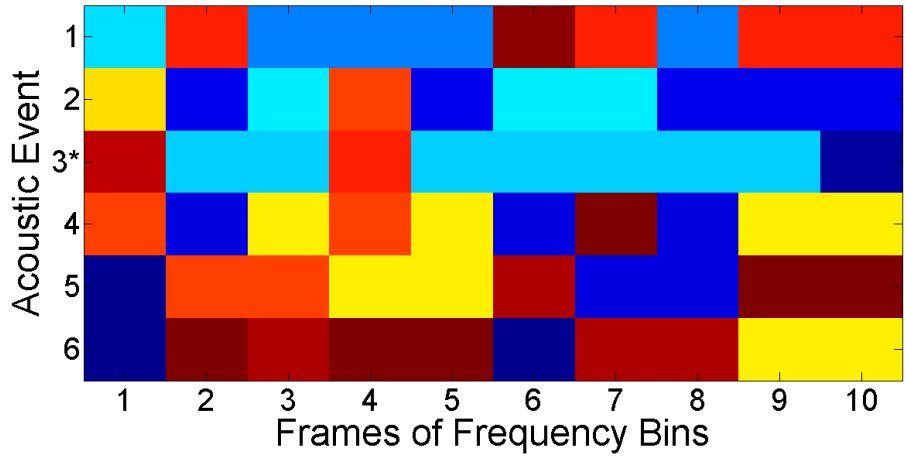

Figure 6.2: Time-varying spectral bins for six different acoustic events. Each color represents a different cluster centroid of the MFCC coefficients.

the clusters (as background or anomalous) is ordering the number of tokens assigned to each cluster. We define $C_B = \{C_k \ : \ |C_k| \geq a\}$, where a is a predetermined threshold, and of course $C_A = \mathcal{C} - C_B$. To simplify our argument, we consider the simplest setting, in which the largest cluster is assumed to be representative of background events. That is,

$$C_B = \arg\max_{C_k} |C_k| \tag{6.2}$$

In this case, Y_B corresponds to the class labels (of tokens) assigned to C_B.

$$Y_B = \{g(x) \ : \ \forall x, \ \ P(f(x) \ \epsilon \ C_B) \geq P(f(x) \ \epsilon \ C_A)\} \tag{6.3}$$

The false alarm (FA) and missed detection (MD) rates are given by the following equations:

$$FA: \ P(f(x) \ \epsilon \ C_A | g(x) \ \epsilon \ Y_B) \tag{6.4}$$

$$MD: \ P(f(x) \ \epsilon \ C_B | g(x) \ \epsilon \ Y_A) \tag{6.5}$$

We have already seen in the previous chapter that sparsifying the cluster assignment probabilities increases purity. Our goal is to show why sparsity also leads to a useful anomaly detection algorithm that minimizes false alarm.

Since cluster purity is a quantity we maximize, let us consider $1 - FA$ and

$1 - MD$, respectively:

$$1 - FA = P(f(x) \in C_B | g(x) \in Y_B) \tag{6.6}$$

$$1 - MD = P(f(x) \in C_A | g(x) \in Y_A) \tag{6.7}$$

To minimize false alarm, we need to maximize $1-FA$, and therefore $P(f(x) = C_B | g(x) \in Y_B)$; this seems to suggest that we would like cluster assignment probabilities to be dense, thereby contradicting our PAC-Bayesian results. Anomaly detection requires a slight modification to the theory – we want the cluster assignment probabilities to be highly sparse for *most* clusters, and dense for the one (or few) background cluster(s). We therefore solve the following optimization problem, where the sparsification terms are summed:

$$\begin{aligned} \underset{\mathcal{Q}}{\text{maximize}} \quad & J(\mathcal{Q}) \\ \text{subject to} \quad & \sum_{c_i} \|q(x_i|c_i)\|_0 \leq V \end{aligned} \tag{6.8}$$

In the context of PAC-Bayesian theory, this restriction still limits each n_i to V and therefore regularizes the bound; however, it is much more restrictive than our original constraint of limiting $\|q(x_i|c_i)\|_0$ for each c_i. In practice, we solve the following relaxation:

$$\begin{aligned} \underset{\mathcal{Q}}{\text{maximize}} \quad & J(\mathcal{Q}) - \eta \sum_{c_i} \|q(x_i|c_i)\|_\alpha \\ \text{subject to} \quad & \eta \geq 0 \end{aligned} \tag{6.9}$$

The key advantage is that by aggressively selecting α and η, we can distribute observation tokens across the clusters to achieve the exact false alarm and missed detection rates we desire, while still ensuring that the small-sample generalization results of PAC-Bayesian theory hold. Note that maximizing $1-FA$ such that $1-MD$ is fixed (and maximizing $1-MD$ such that $1-FA$ is fixed) leads to maximizing average purity as defined in Equation (6.1). However, maximizing average purity does not necessarily guarantee that $1-FA$ and $1-MD$ are maximized. Intuitively, this seems like a saddle-point type situation that warrants further study.

6.4 Results

We test our results on the BBC sound effects corpus – please refer to Chapter 5 for a detailed description of the dataset. To identify a token as an anomaly, we use the simple thresholding strategy described in Equation (6.2). That is, we take the largest cluster as representative of background events, and assume that the rest of the clusters represent anomalous events.

Figure 6.3 is a matrix in which entry (i, j) represents the number of tokens that were assigned to cluster i (of the 35 possible rows), given that their true class is j (of the 48 possible columns).

It is clear that in the first subplot – where we do not encourage any sparsity – it is difficult to isolate any particular cluster as one that is anomalous. Regardless of the threshold we may select, there are always clusters that represent more than one class; hence, to distinguish an anomaly from a background event in a fully unsupervised manner is challenging.

In the second subplot, where we encourage some sparsity, we see one background cluster that represents several classes, and several clusters that represent only one or two classes.

In the third subplot, we see that there is a huge background cluster to which almost every token is assigned, and another cluster that contains all of the tokens from a *specific class*. In this case, sparsity makes it both visually and algorithmically obvious as to what exactly an anomaly is.

To quantify the improvement achieved by sparsity, we use the following decision rule: a token is anomalous if and only if it belongs to the class that is least frequently assigned to the background cluster, i.e., the conditional probability of background cluster assignment given reference class label is smallest. Figure 6.4 is again a matrix in which entry (i, j) represents the number of tokens assigned to cluster i, given that their true class is j (in this case, either an anomaly or a regular event). Note that the number of clusters in both plots is different, and smaller than 35 – as we encourage sparsity, there are more clusters to which no tokens are assigned.

As expected, sparsity and cluster purity significantly reduce the false alarm rate, from **0.10** in the case of medium sparsity ($\alpha = 0.4$) to **0.02** when the observation probabilities are extremely sparse ($\alpha = 0.2$). Of course, such an improvement does not come without a cost – in this case, missed detection (MD) increases from **0** to **0.02**. The advantages and disadvantages

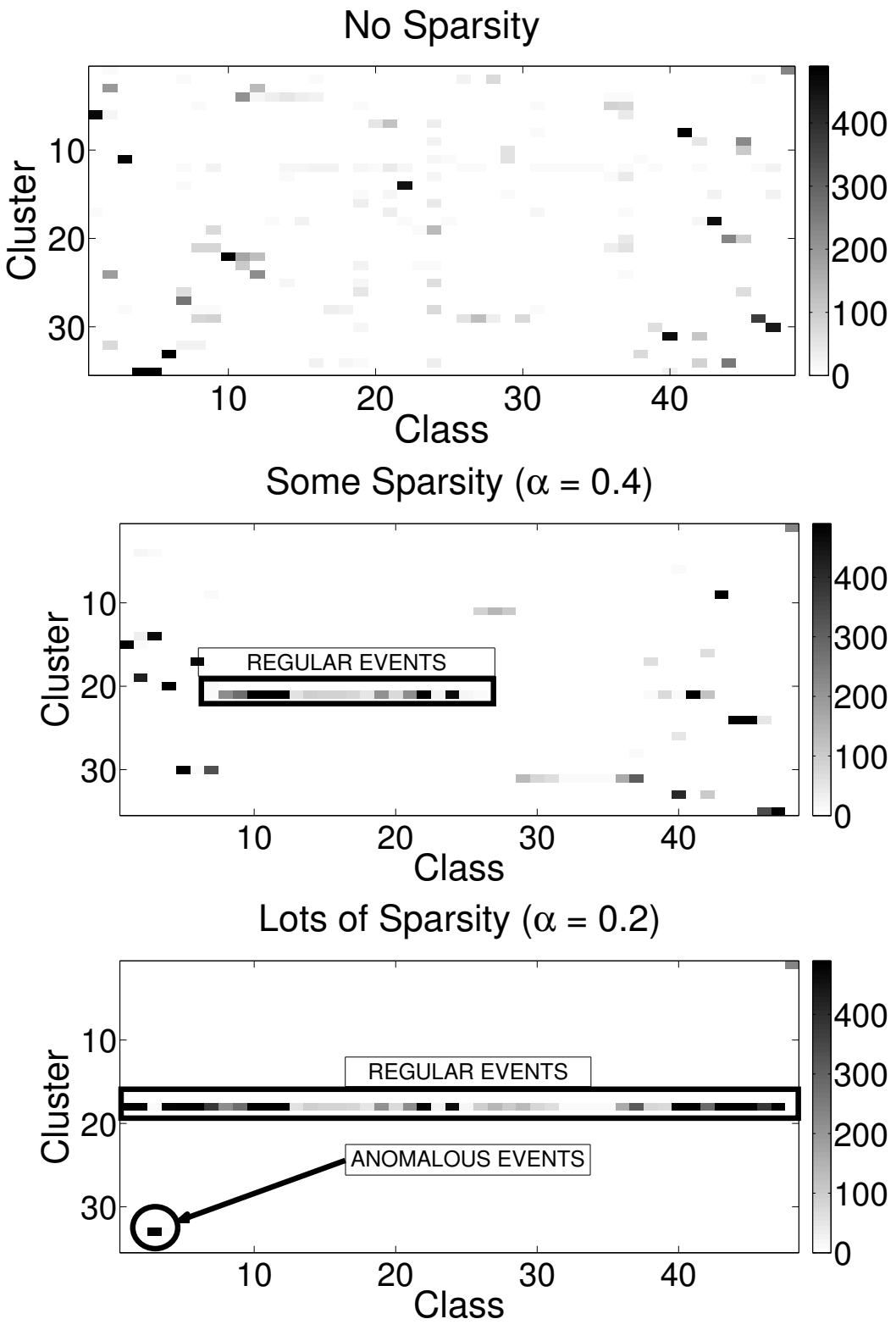

Figure 6.3: A matrix of cluster assignments (y-axis) and their true class (x-axis) for various α

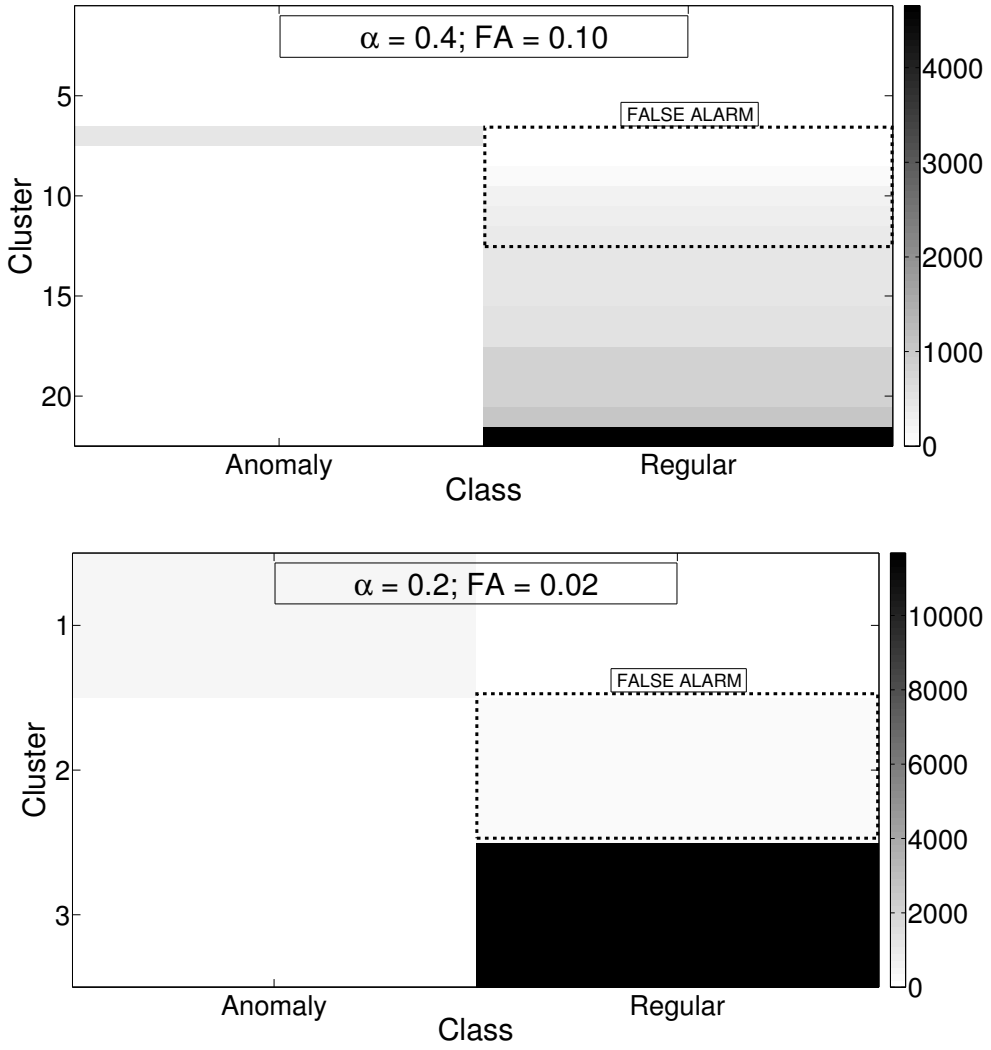

Figure 6.4: False alarm (FA) for $\alpha = 0.4$ (top) and $\alpha = 0.2$ (bottom)

of this tradeoff depend entirely on the application, but our algorithm and experiments show that we can provably control the false alarm rate of an algorithm by adjusting α and η, should we choose to do so.

CHAPTER 7

DISCUSSION

The theory and experiments from previous chapters indicate that a PAC-Bayesian approach is promising for learning languages in a fully unsupervised fashion. In this chapter we show, by example on an artificial dataset, that this may be possible.

Figure 7.1 describes the architecture. We pass speech samples in Gujarati to an acoustic model trained on Indian English, under the assumption that the resulting phone sequence is a noisy approximation of the true Gujarati phone sequence. This sequence is then passed through our HMM-based clustering algorithm. Figure 7.2 illustrates the purest clusters, as well as a mock test scenario in which a clean signal is segmented and clustered into meaningful Gujarati words.

While we realize that such a problem is far fetched, it is nevertheless theoretically intriguing. The following list summarizes some of the key issues to address:

- The success of our method needs to be empirically verified on other corpora. Since our original motivation is small-sample generalization, we need to run the same experiment on the 22-language dataset, which has an equally limited vocabulary.

- We need an evaluation metric for this task. It is not clear if purity is necessarily the most appropriate metric, or if we need to account for both precision and recall with something like the F-score.

- Regardless of the evaluation metric, the success of our approach depends on the similarity between the unknown language and English. It is useful to tighten the PAC-Bayes bound by incorporating this relationship.

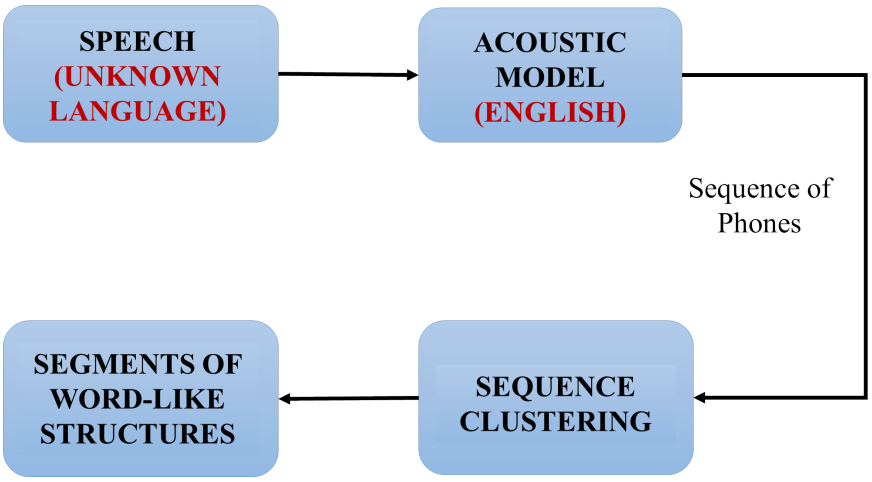

Figure 7.1: An architecture for unsupervised language learning

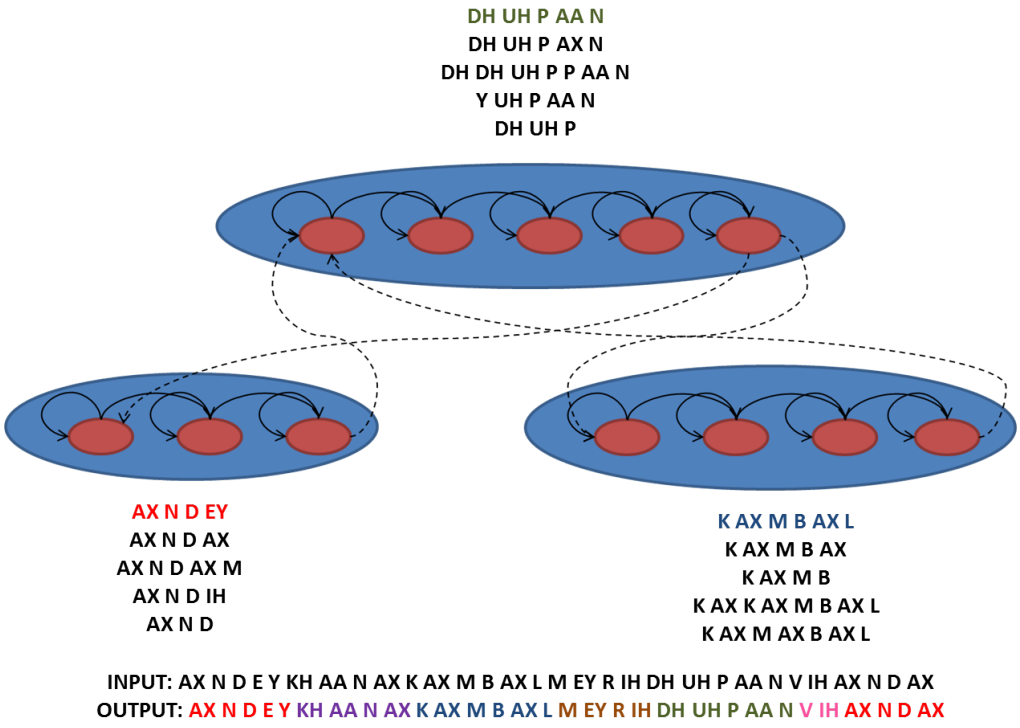

Figure 7.2: A Gujarati example

CHAPTER 8

CONCLUSION

8.1 Summary

We have shown that there is some merit in tackling acoustic, speech, and language processing problems from a learning theoretic point of view. By extending PAC-Bayesian bounds to sequences, we were able to derive provably efficient and practical algorithms that achieve good performance on a variety of tasks – class-based language modeling, acoustic event detection, as well as anomaly detection. Finally, we showed that on a very small and artificial dataset, it may even be possible to learn words from unknown languages in an (almost) unsupervised fashion. Naturally, closing one door opens several more, and quite a few extensions of this work are worth exploring. We summarize some of them below.

8.2 Future Directions

8.2.1 Language Models

The theory developed in this thesis applies to n-grams and class-based language models, which form only a small class of all possible language models. Although n-grams have been popular for several decades, they are now used more as an academic example rather than a practical solution. State-of-the-art methods based on Bayesian nonparametrics (e.g. the hierarchical Pitman-Yor process) as well as neural networks (e.g. recurrent nets) work exceptionally well in practice, but have limited theoretical justification. We believe it may be possible to extend our PAC-Bayesian results to also encapsulate such models. One technique might be to simply view these models as

some form of a clustering algorithm, and apply our bounds.

8.2.2 Anomaly Detection

We have merely modified the sparsification technique of our PAC-Bayesian bound. Anomaly detection has some scope for more rigorous theoretical results. For example, given some $x \in \mathcal{X}$, our goal is to learn some decision function $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{background, anomaly\}$. This would require training examples for both classes, which is usually not possible. However, from a theoretical standpoint, it is interesting to decouple N, the number of training examples, into N_1 (background) and N_2 (anomalous). This allows us to characterize exactly how few examples from the anomalous class are required for good performance.

8.2.3 Zero Resource Speech Recognition

This is the most ambiguously defined problem of the three, and therefore lends itself to several possibilities. The most interesting (and impactful) is the problem of mismatch between training and test sets. For example, clustering a sequence of phones to extract words in Gujarati should benefit much more from using a Hindi acoustic model rather than an English one. We can abstract this problem to some notion of a similarity between training and test signals – perhaps an additional mutual information term in the bounds will do the trick.

REFERENCES

[1] B. Mallikarjun, "An exploration into linguistic majority-minority relations in India," *Language in India*, vol. 4, 2004.

[2] A. Jansen, K. Church, and H. Hermansky, "Towards spoken term discovery at scale with zero resources," in *Proceedings of 11th Annual Conference of the International Speech Communication Association*, 2010, pp. 1676 – 1679.

[3] D. F. Harwath, T. J. Hazen, and J. R. Glass, "Zero resource spoken audio corpus analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 8555–8559.

[4] L. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[5] D. McAllester, "Some PAC-Bayesian theorems," in *COLT' 98 Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998, pp. 230–234.

[6] D. McAllester, "Simplified PAC-Bayesian margin bounds," in *COLT' 03 Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, 2003, pp. 202–215.

[7] J. Langford, "Tutorial on practical prediction theory for classification," *The Journal of Machine Learning Research*, vol. 6, pp. 273–306, 2005.

[8] Y. Seldin and N. Tishby, "PAC-Bayesian analysis of co-clustering and beyond," *The Journal of Machine Learning Research*, vol. 11, pp. 3595–3646, 2010.

[9] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding (ASRU)*, 2003, pp. 411–416.

[10] J. Ajmera, H. Bourlard, I. Lapidot, and I. McCowan, "Unknown-multiple speaker clustering using HMM," in *Intl. Conf. Spoken Language Proc.*, 2002.

[11] M. Padmanabhan, L. Bahl, D. Nahamoo, and M. A. Picheny, "Speaker clustering and transformation for speaker adaptation in speech recognition systems," *IEEE Trans. on Speech and Audio Processing*, vol. 6, pp. 71–77, 1998.

[12] R. Xu, "Survey of clustering algorithms," *IEEE Trans. on Neural Networks*, vol. 16, pp. 645–678, 2005.

[13] P. Smyth, "Clustering sequences with hidden Markov models," in *Advances in Neural Information Processing (NIPS)*, vol. 9, 1997, pp. 648–654.

[14] C. Li and G. Biswas, "Clustering sequence data using hidden Markov model representation," in *Proceedings of the SPIE '99 Conference on Data Mining and Knowledge Discovery*, 1999, pp. 14–21.

[15] M. Bicego, V. Murino, and M. Figueiredo, "Similarity-based clustering of sequences using hidden Markov models," in *Proc. of Intl. Conf. on Machine Learning and Data Mining in Pattern Recognition*, 2003, pp. 86–95.

[16] D. Garcia-Garcia, E. Parrado-Hernandez, and F. Diaz-de Maria, "A new distance measure for model-based sequence clustering," *IEEE Trans. on PAMI*, vol. 31, no. 7, pp. 1325–1331, 2009.

[17] A. Panuccio, M. Bicego, and V. Murino, "A hidden Markov model-based approach to sequential data clustering," in *Proceedings of the International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp. 734–742.

[18] L. Rabiner, C. Lee, B. Juang, and J. Wilpon, "HMM clustering for connected word recognition," in *Proc. of ICASSP*, vol. 1, 1989, pp. 405–408.

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

[20] P. Smaragdis, "Approximate nearest-subspace representations for sound mixtures," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011, pp. 5892–5895.

[21] M. Pilanci, L. E. Ghaoui, and V. Chandrasekaran, "Recovery of sparse probability measures via convex programming," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012, pp. 2429–2437.

[22] A. Kyrillidis, S. Becker, V. Cevher, and C. Koch, "Sparse projections onto the simplex," *JMLR: Workshop and Conference Proceedings, Proceedings of the 30th International Conference on Machine Learning*, vol. 28, no. 2, pp. 235–243, 2013.

[23] R. Chartrand and V. Staneva, "Restricted isometry properties and nonconvex compressive sensing," *Inverse Problems*, vol. 24, pp. 1–14, 2008.

[24] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[25] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing (NIPS)*, vol. 14, 2002.

[26] S. Bharadwaj, M. Hasegawa-Johnson, J. Ajmera, O. Deshmukh, and A. Verma, "Sparse hidden Markov models for purer clusters," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 3098–3102.

[27] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," International Computer Science Institute, Berkeley, California, Tech. Rep. TR-97-021, 1998.

[28] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & sons, Inc., 2008.

[29] R. Chartrand and V. Staneva, "Restricted isometry properties and nonconvex compressive sensing," *Inverse Problems*, vol. 24, pp. 1–14, 2008.

[30] I. Good, "The population frequencies of species and the estimation of population parameters," *Biometrica*, vol. 40, no. 3, pp. 237–264, 1953.

[31] G. Lidstone, "Note on the general case of the Bayes-Laplace formula for inductive or *a posteriori* probabilities," *Transactions of the Faculty of Actuaries*, vol. 8, pp. 182–192, 1920.

[32] M. I. Ohannessian and M. A. Dahleh, "Rare probability estimation under regularly varying heavy tails," *JMLR: Workshop and Conference Proceedings, 25th Annual Conference on Learning Theory*, vol. 23, no. 21, pp. 1–24, 2012.

[33] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, 1987.

[34] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Proceedings of Workshop on Pattern Recognition in Practice*, 1980, pp. 381–397.

[35] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 181–184.

[36] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, 2000.

[37] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 13, pp. 359–394, 1999.

[38] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006, pp. 985–992.

[39] D. J. Mackay and L. C. B. Peto, "A hierarchical Dirichlet language model," *Natural Language Engineering*, vol. 1, no. 03, pp. 289–308, 1995.

[40] P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[41] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proceedings of the Second International Symposium on Information Theory*, 1973, pp. 267–281.

[42] K. P. Burnham and D. R. Anderson, *Model Selection and Multi-model Inference: A Practical Information-Theoretic Approach.* Springer, 2002.

[43] S. F. Chen, "Performance prediction for exponential language models," in *Proceedings of NAACL HLT*, 2009, pp. 450–458.

[44] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," *Computer, Speech, and Language*, vol. 10, pp. 187–228, 1996.

[45] S. Deligne and F. Bimbot, "Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 169–172.

[46] K. Ries, F. D. Buo, and A. Waibel, "Class phrase models for language modeling," in *ICSLP '96 Proceedings of the Fourth International Conference on Spoken Language*, 1996, pp. 398–401.

[47] R. Justo and M. I. Torres, "Different approaches to class-based language models using word segments," *Computer Recognition Systems 2, Advances in Soft Computing*, vol. 45, pp. 421–428, 2007.

[48] J. C. Principe, *Information Theoretic Learning*. Springer, 2010.

[49] P. Price, W. Fisher, J. Bernstein, and D. Pallett, "Resource Management RM 2.0," in *Linguistic Data Consortium, Philadelphia*, 1993.

[50] M. Slaney, "Semantic-audio retrieval," in *ICASSP*, 2002, pp. 1408–1411.

[51] V. Chandola., A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, 2009.

[52] V. Chandola., A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: a survey," *IEEE Trans. on Knowledge and Data Engineering*, vol. 24, pp. 823–839, 2012.

[53] S. Bharadwaj and M. Hasegawa-Johnson, "A PAC-Bayesian approach to minimum perplexity language modeling," in *Proc. of. Intl. Conf. on Computational Linguistics (COLING)*, pp. 130–140.

[54] M. Bicego, M. Cristani, and V. Murino, "Sparseness achievement in hidden Markov models," in *Proceedings of the International Conference on Image Analysis and Processing*, 2007, pp. 67–72.

[55] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 255–262.

[56] Y. Qiao, X. Xin, Y. Bin, and S. Ge, "Anomaly intrusion detection method based on HMM," *Electronic Letters*, vol. 38, pp. 663–664, 2002.

[57] X. Zhang, P. Fan, and Z. Zhu, "A new anomaly detection method based on hierarchical HMM," in *Proc. of the Intl. Conf. on Parallel and Distributed Computing, Applications and Technologies*, 2003, pp. 249–252.

[58] G. Florez-Larrahondo, S. Bridges, and R. Vaughn, "Efficient modeling of discrete events for anomaly detection using hidden Markov models," *Information Security*, vol. 3650, pp. 506–514, 2005.