

## Lecture 14: Feb 23, 2005

Lecturer: Prof. J. Bilmes <bilmes@ee.washington.edu>

Scribe: Kevin Duh

### 14.1 Overview

There are four main problems associated with the hidden Markov model (HMM):

1. Compute the probability of the evidence  $p(x)$  (Forward or Backward algorithm)
2. Compute the most likely path for the hidden states (Viterbi path)
3. Train parameters such that the likelihood of data is maximized (Baum-Welch Re-estimation)
4. Train parameters in a discriminative fashion (e.g. MMIE, MCE)

In this lecture, we will address problems 3 and 4. First, continuing from the previous lecture, we will view Baum-Welch Re-estimation as an instance of the Expectation-Maximization (EM) algorithm and prove why the EM algorithm maximizes data likelihood. Then, we will proceed to discuss discriminative training under the maximum mutual information estimation (MMIE) framework. Specifically, we will discuss the motives for discriminative training, the derivation of the MMIE criteria, and two methods for optimizing MMIE (gradient descent and extended Baum-Welch).

### 14.2 HMM: EM Algorithm

#### 14.2.1 Recap: Update equations for Baum-Welch Re-estimation

Recall from the previous lecture that the update equations for training a discrete HMM are:

$$\begin{aligned} \text{Initial probabilities: } \pi_j &= p(Q_1 = j | \lambda^g, x_{1:T}) \\ \text{Output probabilities: } p_{i,q}^{(\lambda)} &= \frac{\sum_{t=1}^T \mathbf{1}(x_t = x^{(i,q)}) p(x_{1:T}, Q_t = q | \lambda^g)}{\sum_{t=1}^T p(x_{1:T}, Q_t = q | \lambda^g)} \\ \text{Transition probabilities: } a_{ij} &= \frac{\sum_{t=2}^T p(x_{1:T}, Q_{t-1} = i, Q_t = j | \lambda^g)}{\sum_{t=2}^T p(x_{1:T}, Q_{t-1} = i)} \end{aligned}$$

where  $Q_t$  is the random variable representing the hidden state at time  $t$ ,  $\lambda^g$  is the given parameter values of the previous EM iteration, and  $x_{1:T}$  is the entire observation vector.

These update equations are actually quite intuitive. For instance, the output probability  $p_{i,q}$  is simply the count of observations equal to  $i$  at state  $q$ , but weighted by the state  $q$ 's occupation probability for each time  $t$  and normalized by the total state  $q$  occupation probability. Similarly, the transition probability from state  $i$  to  $j$  is simply updated as the *expected* number of transitions from state  $i$  to  $j$  divided by the *expected* number of times in state  $i$ .

## 14.2.2 EM Algorithm interpretation

How did we arrive at the above update equations? Recall we did so by maximizing the auxiliary function with respect to  $\lambda$ . We view  $Q_{1:T}$  as the hidden variable and define the auxiliary function as the expected value of the complete-data likelihood:

$$Q(\lambda, \lambda^g) = E[\log p(X_{1:T}, Q_{1:T}|\lambda) | X_{1:T}, \lambda^g] \quad (14.1)$$

$$= \sum_{q_{1:T}} [\log p(x_{1:T}, q_{1:T}|\lambda)] p(x_{1:T}, q_{1:T}|\lambda^g) \quad (14.2)$$

For each EM iteration, we update the parameter  $\lambda$  by maximizing  $Q(\lambda, \lambda^g)$  w.r.t  $\lambda$ :

$$\begin{aligned} \lambda^i &= \arg \max_{\lambda} Q(\lambda, \lambda^i) \\ i &= i + 1 \end{aligned} \quad (14.3)$$

The (log-)likelihood function  $\log p(x_{1:T}|\lambda)$  can be maximized by iteratively computing  $\lambda$  using Eq. 14.3 until convergence (i.e. until  $\lambda$  changes little from one iteration to the next). This is a very powerful result, as our ultimate goal in training HMMs is to optimize  $\log p(x_{1:T}|\lambda)$ . To repeat: the EM algorithm maximizes the likelihood by iteratively maximizing the auxiliary function. By doing so, it obtains the local optimum of a likelihood function that is otherwise not directly optimizable.

To get an intuitive understanding of what the EM algorithm is doing exactly, consider a plot of the likelihood and auxiliary functions in Figure 1. Our goal is to find the  $\lambda^{**}$ , the global maximum of the likelihood function  $L(\lambda) = p(x_{1:T}|\lambda)$ . However, since the likelihood function can be arbitrary, we cannot optimize it easily. Instead, we pick an initial parameter  $\lambda^1$  and construct its corresponding auxiliary function  $Q(\lambda, \lambda^1)$ . Since this auxiliary function is provably concave, we can easily maximize it, getting  $\lambda^2$ . Next we construct  $Q(\lambda, \lambda^2)$  and maximize to find  $\lambda^3$ . By iteratively constructing these auxiliary functions and maximizing them, we can climb the surface of the likelihood function to arrive at a local optima,  $\lambda^*$ .

Note that the EM algorithm only guarantees finding the local optimum  $\lambda^*$ , not the global optimum  $\lambda^{**}$ . In practice, a variety of techniques from optimization theory can be employed to improve the estimate. For example, one can adjust the "width" of the auxiliary function to make  $\lambda$  jump in different magnitudes. One can also start at random initial values  $\lambda^1$  to explore a potentially wider space.

As a side-note, it is worth pointing out that the term  $p(x_{1:T}, q_{1:T}|\lambda^g)$  in Eq. 14.2 would be changed to  $p(q_{1:T}|x_{1:T}, \lambda^g)$  if one uses the standard form of EM, which takes the expectation with respect to the posterior probability of  $Q$ . However, by the chain rule,  $p(q_{1:T}|x_{1:T}, \lambda^g) = p(x_{1:T}, q_{1:T}|\lambda^g)p(x_{1:T}|\lambda^g)$ ; since we eventually maximize the auxiliary function only with respect to  $\lambda$ , the  $p(x_{1:T}|\lambda^g)$  term drops out. We leave the term as a joint probability because it can be efficiently calculated using the alpha and beta recursions.

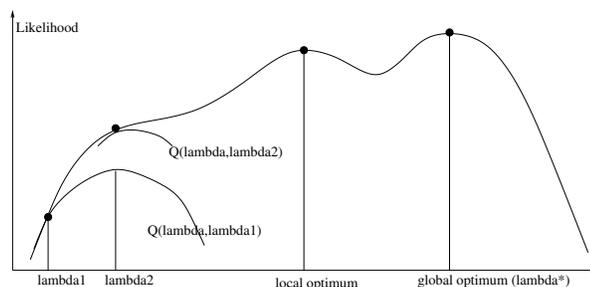


Figure 14.1: Pictorial view of the EM algorithm

### 14.2.3 Proof: Why EM maximizes likelihood

Now we will rigorously prove why the EM algorithm maximizes the likelihood function. Before we do this, we need to discuss the notion of convexity/concavity and present the Jensen's Inequality [CT91].

**Definition 14.1.** A function  $f(x)$  is **convex** over an interval  $(a, b)$  if for every  $x_1, x_2 \in (a, b)$  and  $0 \leq \lambda \leq 1$ ,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2). \quad (14.4)$$

A function is **strictly convex** if equality holds only in the case  $\lambda = 0$  or  $\lambda = 1$ .

**Definition 14.2.** A function  $f$  is **concave** if  $-f$  is convex.

What are some examples of convex and concave functions?

- Strictly convex:  $x^2$ ,  $\exp^x$ , and  $x \log x, \forall x \geq 0$
- Strictly concave:  $\log x$  and  $\sqrt{x}$  for all  $x \geq 0$
- Both convex and concave: Linear function  $ax + b$

**Theorem 14.3. Jensen's Inequality:** If  $f$  is a convex function and  $X$  is a random variable, then

$$Ef(X) \geq f(EX). \quad (14.5)$$

Moreover, if  $f$  is strictly convex, then equality in Eq. 14.5 implies that  $X = EX$  with probability 1, i.e.  $X$  is a constant.

In words, Jensen's Inequality says that the expected value of the function of some random variable is greater than the function of the expected value of that variable, provided that the function is convex. We will sketch the proof here for discrete distributions, using proof by induction.

- Base case: For a two mass point distribution, we can write the following by the definition of convex functions:

$$p_1 f(x_1) + p_2 f(x_2) \geq f(p_1 x_1 + p_2 x_2) \quad (14.6)$$

- Inductive case: Suppose the theorem is true for distributions with  $k - 1$  mass points. Let  $p'_i = \frac{p_i}{1 - p_k}$  for  $i = 1, 2, \dots, k - 1$ . Then we have:

$$\begin{aligned} Ef(X) &= \sum_{i=1}^k p_i f(x_i) = p_k f(x_k) + (1 - p_k) \sum_{i=1}^{k-1} p'_i f(x_i) \\ &\geq p_k f(x_k) + (1 - p_k) f\left(\sum_{i=1}^{k-1} p'_i x_i\right) \end{aligned} \quad (14.7)$$

$$\begin{aligned} &\geq f\left(p_k x_k + (1 - p_k) \sum_{i=1}^{k-1} p'_i x_i\right) \\ &= f\left(\sum_{i=1}^k p_i x_i\right) = f(EX) \end{aligned} \quad (14.8)$$

The inequality in 14.7 follows from the induction hypothesis; the inequality in 14.8 follows because  $f$  is convex.

Now we are ready to show why EM maximizes likelihood:

**Theorem 14.4.** *If  $Q(\lambda, \lambda^g) \geq Q(\lambda^g, \lambda^g)$ , then  $P(\lambda) \geq P(\lambda^g)$ , where  $Q(\cdot)$  is the auxiliary function defined in Eq. 14.2 and  $P(\lambda) = p(x_{1:T}|\lambda)$  is the data likelihood we are trying to maximize.*

The proof is as follows:

$$\begin{aligned} \log P(\lambda) - \log P(\lambda^g) &= \log \left\{ \sum_{q_{1:T}} p(x_{1:T}, q_{1:T}|\lambda) \right\} - \log P(\lambda^g) \\ &= \log \left\{ \sum_{q_{1:T}} p(x_{1:T}, q_{1:T}|\lambda) \frac{p(x_{1:T}, q_{1:T}|\lambda^g)}{p(x_{1:T}, q_{1:T}|\lambda^g)} \right\} - \log P(\lambda^g) \end{aligned} \quad (14.9)$$

$$= \log \left\{ \sum_{q_{1:T}} p(x_{1:T}, q_{1:T}|\lambda^g) \frac{p(x_{1:T}, q_{1:T}|\lambda)}{p(x_{1:T}, q_{1:T}|\lambda^g)} \right\} - \log P(\lambda^g) \quad (14.10)$$

$$\geq \sum_{q_{1:T}} p(x_{1:T}, q_{1:T}|\lambda^g) \log \left\{ \frac{p(x_{1:T}, q_{1:T}|\lambda)}{p(x_{1:T}, q_{1:T}|\lambda^g)} \right\} - \log P(\lambda^g) \quad (14.11)$$

$$\begin{aligned} &= \sum_{q_{1:T}} p(x_{1:T}, q_{1:T}|\lambda^g) \{ \log p(x_{1:T}, q_{1:T}|\lambda) - \log p(x_{1:T}, q_{1:T}|\lambda^g) \} - \log P(\lambda^g) \\ &= \{ Q(\lambda, \lambda^g) - Q(\lambda^g, \lambda^g) \} - \log P(\lambda^g) \geq 0 \end{aligned} \quad (14.12)$$

Eq. 14.9 and 14.10 is a simple multiplication by 1 and rearrangement of terms. The inequality in 14.11 follows from Jensen's Inequality, using the fact that the log function is convex. In the last line, the nonnegativity in 14.12 is true because (1) the difference  $Q(\lambda, \lambda^g) - Q(\lambda^g, \lambda^g) \geq 0$  by definition and (2) the negative log of a probability ( $-\log P(\lambda^g)$ ) is nonnegative. Finally, since log is a monotonic function,  $\log P(\lambda) - \log P(\lambda^g) \geq 0$  implies that  $P(\lambda) - P(\lambda^g) \geq 0$ .

## 14.3 Discriminative Training

### 14.3.1 Motivation

The EM algorithm maximizes the likelihood, but is this the ideal criterion for speech recognition? Recall that we use the Bayes decision rule, which for isolated word recognition of different words  $M$  corresponds to the following:

$$M^* = \arg \max_M p(M|x) = \arg \max_M p(x|M)p(M)$$

This Bayes decision rule achieves minimum classification error by choosing the model  $M$  which maximizes the posterior probability  $p(M|x)$ . However, does maximum likelihood estimation give accurate posterior probability estimates in practice? It turns out that maximum likelihood estimation gives optimal estimates (in the sense of being minimum variance and unbiased) only if three conditions are satisfied:

1. the model correctly represented the stochastic process. (e.g. the HMM was the correct model for speech)
2. we had an infinite amount of training data
3. we found the true global maximum of the likelihood

In practice, none of the above conditions are satisfied, so we must reconsider whether maximum likelihood is the best estimation criterion to employ. This is the motivation for discriminative training.

In this lecture, we discuss discriminative training based on the Maximum Mutual Information criterion, which corresponds to modeling the posterior probability  $p(M|x)$  directly (as opposed to modeling the likelihood  $p(x|M)$ ). The idea is to maximize the "information flow" between acoustics and words. We would like the conditional entropy of words given acoustics to be minimized, so that we would have a better chance of choosing the correct word. As we will show later, this is achieved by simultaneously increasing the likelihood of a correct model (as done in maximum likelihood estimation) and decreasing the likelihoods of all incorrect models. The result is that models become more discriminatory in nature, which should lead to lower recognition error. There are also other discriminative training methods, such as minimum classification error (MCE) training, that directly attempts to minimize error rate.

Bear in mind that discriminative training is not the cure-all. In practice, discriminative training requires significantly more computation. Even when computation is not an issue, the amount of data or complexity of model also affects the parameter estimation process. For instance, Ng and Jordan [NJ02] showed that generatively trained models (i.e. maximum likelihood estimation) and discriminatively trained models perform differently depending on the amount of data. As shown in Figure 2, the accuracy of discriminative training outperforms generative training only for large data sizes. Further, Nadas [N83] showed that if the assumptions for prior and likelihood are correct, then both MLE and MMIE produce consistent estimates, but MMIE has a greater variance. In fact, when Bahl et. al. [B86] introduced MMIE training in 1986, the results were mixed. It is not until recently that discriminative training has shown consistent gains, thanks to the availability of significantly more data.

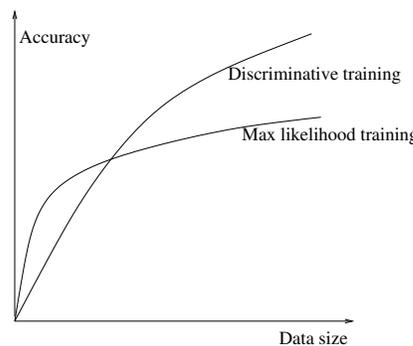


Figure 14.2: Accuracy vs. Data size for Likelihood-based and Discriminatory training

### 14.3.2 Quick Review of Information Theory

Before we present the MMIE criterion, it would be beneficial to refresh our minds on some basic information theory concepts. For more detail, refer to [CT91] or [HAH01].

The **entropy** of a discrete random variable  $X$  is defined as  $H(X) = -\sum_x p(x) \log p(x)$ , where  $p(x)$  is the probability mass function of  $X$ . Entropy measures the amount of uncertainty or information of  $X$ . Intuitively, if  $p(x)$  is a uniform distribution, then all values of  $x$  are equally likely, giving rise to the maximum amount of uncertainty (maximum entropy). Knowing the value of  $X$  in this case gives the greatest information. Since  $0 \leq p(x) \leq 1$ ,  $H(X)$  is strictly non-negative. Further, entropy is a concave function of the distribution.

The **joint entropy** of a pair of discrete random variables  $(X, Y)$  with joint distribution  $p(x, y)$  is defined as  $H(X, Y) = -\sum_{x,y} p(x, y) \log p(x, y)$ . The **conditional entropy** is the uncertainty of a random variable given another variable: i.e.  $H(Y|X) = -\sum_{x,y} p(x, y) \log p(y|x)$ . The **chain rule of entropy** states that  $H(X, Y) = H(X) + H(Y|X)$ , which means that the joint uncertainty of two variables is the entropy of one variable plus the condition entropy.

The **Kullback-Liebler (KL) Divergence** (a.k.a relative entropy) between two probability mass functions  $p(x)$  and  $q(x)$  is defined as  $D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ . KL divergence is non-negative and is zero if and only if  $p = q$ . It measures the "distance" between two distributions but is not a true distance metric due to the lack of symmetry and

triangle inequality properties.

The **mutual information** of two discrete random variables (X,Y) is the KL divergence between the joint distribution  $p(x, y)$  and the product distribution  $p(x)p(y)$ . i.e.,

$$I(X; Y) = D(p(x, y) || p(x)p(y)) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

This definition of mutual information can be rewritten in terms of entropy:  $I(X; Y) = H(X) - H(X|Y)$ . This can be interpreted as saying "the mutual information that is shared between X and Y is the reduction of X's uncertainty due to the knowledge of Y." If knowing Y tells us a lot about X, then  $H(X|Y)$  is small and  $I(X; Y)$  becomes large; conversely, if Y and X exhibit little dependency between each other, then  $H(X|Y)$  remains large and  $I(X; Y)$  becomes small.  $I(X; Y)$  is lower-bounded by zero and upper-bounded by  $\min H(X), H(Y)$ .

### 14.3.3 The Maximum Mutual Information Estimation (MMIE) Criterion

In MMIE, we attempt to directly model the posterior probability  $p(M|x)$ . Therefore, we find the parameters  $\lambda$  that minimize the KL divergence between the true  $p(M|x)$  and the estimated  $p(M|x, \lambda)$ :

$$\lambda^* = \arg \min_{\lambda} D(p(M|x) || p(M|x, \lambda)) \quad (14.13)$$

Writing out the equations, we discover that minimizing KL divergence is equivalent to maximizing the mutual information. To see this, first we re-write the KL divergence as:

$$\begin{aligned} D(p(M|x) || p(M|x, \lambda)) &= \sum_{m,x} \log \frac{p(m|x)}{p(m|x, \lambda)} \\ &= \sum_{m,x} p(m, x) \log p(m|x) - \sum_{m,x} p(m, x) \log p(m|x, \lambda) \\ &= -H(M|X) - \sum_{m,x} p(m, x) \log p(m|x, \lambda) \end{aligned} \quad (14.14)$$

Now, minimizing Eq. 14.14 w.r.t  $\lambda$  becomes maximizing its second term:

$$\begin{aligned} \lambda^* &= \arg \max_{\lambda} \sum_{m,x} p(m, x) \log p(m|x, \lambda) \\ &= \arg \max_{\lambda} \sum_{m,x} p(m, x) \log \frac{p(m|x, \lambda)}{p(m)} \\ &= \arg \max_{\lambda} I_{\lambda}(M; X) \end{aligned} \quad (14.15)$$

Eq. 14.15 leads to the following important equation:

$$\begin{aligned} \arg \max_{\lambda} I_{\lambda}(M; X) &= \arg \max_{\lambda} \sum_{m,x} p(m, x) \log \frac{p(m|x, \lambda)}{p(m)} \\ &= \arg \max_{\lambda} \sum_{m,x} p(m, x) \log \frac{p(x|m, \lambda)}{\sum_{m'} p(x|m', \lambda)p(m')} \end{aligned} \quad (14.16)$$

What this MMIE criterion (Eq. 14.16) is saying is essentially this: we want to find the  $\lambda$  that maximizes the likelihood  $p(x|m, \lambda)$  of the correct model M while at the same time minimizing the total of all other likelihoods

$\sum_{m'} p(x|m', \lambda)p(m')$ . This results in discriminative training because we are considering both data for correct and incorrect models in the training process. Parameters will not be tuned to maximize the likelihoods of both the correct and incorrect models, since that is a waste of modeling effort.

From Eq. 14.16, we can also see two practical challenges for implementing MMIE. First, the term  $\sum_{m'} p(x|m', \lambda)p(m')$  is the sum over *all* possible models. For large vocabulary continuous speech recognition (LVCSR),  $m'$  is not only all words but all possible sentences, making this summation an intractable task. Second, the term  $p(m, x)$  is the true joint distribution, so it is unknown in practice. Therefore, we require large training data size in order to invoke the Law of Large Numbers:

$$\begin{aligned} I_\lambda(M; X) &= \sum_{m,x} p(m, x) \log \frac{p(x|m, \lambda)}{\sum_{m'} p(x|m', \lambda)p(m')} \\ &= \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{i=1}^K \log \frac{p(x_i|m_i, \lambda)}{\sum_{m'} p(x_i|m', \lambda)p(m')} \end{aligned}$$

where  $K$  is the size of the training data. The empirical criterion function we wish to maximize is thus:

$$I_\lambda(M; X) = \frac{1}{K} \sum_{i=1}^K \log \frac{p(x_i|m_i, \lambda)}{\sum_{m'} p(x_i|m', \lambda)p(m')}$$

### 14.3.4 MMIE by Gradient Descent

One method for MMIE training is to use gradient descent. Without loss of generality in the following derivations, we assume either that  $K = 1$  or all the data are i.i.d. and wrapped within the variables  $(m, x)$ , so that our treatment of Eq. 14.3.3 can be simplified by disregarding the summation over  $K$ .

In maximum likelihood estimation, we define the criterion function  $f_{MLE} \triangleq \log p(x|m, \lambda)$ , so the gradient is:  $f'_{MLE} \triangleq \frac{\partial \log p(x|m, \lambda)}{\partial \lambda_i} = \frac{1}{p(x|m, \lambda)} \frac{\partial p(x|m, \lambda)}{\partial \lambda_i}$ . Now we can express the MMIE gradient in terms of the MLE gradient:

$$\begin{aligned} f_{MMIE}(m|x) &\triangleq \log p(x|m, \lambda) - \log p(x|\lambda) \\ \frac{\partial f_{MMIE}(m|x)}{\partial \lambda_i} &= f'_{MLE}(x|m) - \sum_{m'} f'_{MLE}(x|m') p(x|m', \lambda) \frac{p(m'|\lambda)}{p(x|\lambda)} \\ &= f'_{MLE}(x|m) [1 - p(x|m, \lambda) \frac{p(m|\lambda)}{p(x|\lambda)}] - \sum_{m' \neq m} f'_{MLE}(x|m') p(x|m', \lambda) \frac{p(m'|\lambda)}{p(x|\lambda)} \\ &= f'_{MLE}(x|m) [1 - p(m|x, \lambda)] - \sum_{m' \neq m} f'_{MLE}(x|m') p(m'|x, \lambda) \end{aligned} \quad (14.17)$$

As seen from Eq. 14.17, the gradient of the MMIE objective function is the weighted sum of several MLE gradients, one in the direction of the correct model  $m$  and others in the direction of incorrect models  $m'$ . In particular, if the posterior for the correct model is high, then it is no longer necessary to proceed in the direction of the correct MLE gradient. On the other hand, if the posterior for any incorrect model is large, then we would want to strongly go in the opposite direction of the incorrect model's MLE gradient. The result is that MMIE will not give high scores to incorrect models. Gradient descent is therefore a reasonable and theoretically beautiful way to optimize the parameters.

Nevertheless, gradient descent for MMIE does have some issues:

1. No guaranteed convergence

2. Computationally expensive
3. It is only first order, so higher order methods like Newton's method may train parameters faster.

Since the introduction of the Extended Baum-Welch algorithm in the early 1990's, gradient descent algorithms for MMIE training have fallen out of popularity. However, there may still be some issues that are worth re-inspecting, such as the performance of gradient-descent trained LVCSR.

### 14.3.5 MMIE by Extended Baum-Welch

An alternative strategy for MMIE training is to extend the Baum-Welch update equations [G91]. The basic idea is to generalize the conditions in Baum's theorem so that the equations in MMIE can be maximized in a similar fashion. Baum's theorem is stated as follows:

**Theorem 14.5. Baum's Theorem (1967):**  $P(\Lambda)$  is a homogeneous polynomial with non-negative coefficients and degree  $d$ , defined on multiple simplexes  $D$ , such that

$$\sum_{j=1}^{q_i} \lambda_{ij} \frac{\partial P(\lambda_{ij})}{\partial \Lambda_{ij}} \neq 0, \forall i \quad (14.18)$$

If the transformaton  $\xi = T(\lambda)$  is defined as

$$\xi_{ij} = \frac{\lambda_{ij} \frac{\partial P(\lambda_{ij})}{\partial \Lambda_{ij}}}{\sum_{j=1}^{q_i} \lambda_{ij} \frac{\partial P(\lambda_{ij})}{\partial \Lambda_{ij}}} \quad (14.19)$$

then  $P(T(\lambda)) \geq P(\lambda)$ , with equality only if  $T(\lambda) = \lambda$ .

To extend the above theorem to MMIE training, we need to deal with three things:

1. Having a rational function of polynomials rather than only a polynomial. Compared to MLE, MMIE results in a rational function due to the addition of the denominator term in the objective function (see Eq. 14.16).
2. Having negative coefficients for the polynomials
3. Having non-homogeneous polynomials

Before we delve into how to deal with the above three issues, let us give some basic definitions and examples:

- Example of a rational function of homogeneous polynomial of probabilities  $p_i$  with nonnegative coefficients:

$$R(p_1, p_2, p_3) = \frac{3p_1^2}{p_1^2 + p_2^2 + 2p_3^2}$$

- Example of a rational function of non-homogeneous polynomial of probabilities  $p_i$  with negative coefficients:

$$R(p_1, p_2, p_3) = \frac{3p_1^2}{p_1^4 + p_2^2 - 2p_3^2}$$

- General rational functions of polynomials of probabilities with nonnegative coefficients:

$$P(\Lambda) = P(\{\Lambda_{ij}\}), i = 1..p, j = 1..q_i$$

where  $\Lambda$  are polynomials defined over multiple simplexes  $D$ :

$$\Lambda \in D = \{\lambda_{ij} : 0 \leq \lambda_{ij} \leq 1, \sum_{j=1}^{q_i} \lambda_{ij} = 1\}$$

One can imagine that  $\Lambda$  is a "matrix" that has unequal length rows. Each row represents a discrete probability mass function of a specific conditional probability table in model (e.g. transition probability table, emission probability table). Each instance of this lies on a  $N$ -dimensional simplex, where  $N$  is the cardinality of the random variable. The purpose of parameter estimation is to adjust the point in each of the simplexes such that some objective function is optimized. For MMIE, we have a ratio of these polynomials:

$$R(\Lambda) = \frac{S_1(\Lambda)}{S_2(\Lambda)}$$

Now we are ready to extend Baum's theorem to the MMIE case. We will do this for discrete distributions (following Gopalakrishnan [G91]). For extensions to Gaussian densities, see Normadin's paper [N91]. We define a **growth transformation**  $T$  of  $D$  for  $R(\Lambda)$  to be a transformation such that

$$\forall \lambda \in D \text{ and } \xi = T(\lambda)$$

$$\text{then } R(\xi) > R(\lambda) \text{ if } \lambda \neq \xi$$

In words, the growth transformation updates the probability distributions of  $\lambda$  so that the  $R(\cdot)$  increases. Our goal is to find a growth transformation for  $R(\cdot)$ . We do this by (1) showing that the growth transformation of a particular polynomial  $P$  is also the growth transformation of the ratio of polynomials  $R$ , and (2) changing this polynomial  $P$  to a form that satisfies the conditions of Baum's theorem, which gives us a growth transformation  $T$ . We define a three-step procedure to go from a  $R$  to a  $P$  that satisfies Baum's conditions:

1. Change ratio of polynomial  $R(\Lambda)$  into a polynomial  $P(\Lambda)$
2. Add a constant to  $P(\Lambda)$  so that the new polynomial  $P'(\Lambda) \triangleq P(\Lambda) + C(\Lambda)$  has only nonnegative coefficients.
3. Do a variable substitution to ensure that the resulting polynomial  $P''(\Lambda)$  is homogeneous.

**STEP 1:** Change the ratio of polynomials  $R(\lambda) = \frac{S_1(\lambda)}{S_2(\lambda)}$  into a polynomial by defining the polynomial as:

$$P_\lambda(\Lambda) \triangleq S_1(\Lambda) - R(\lambda)S_2(\Lambda) \tag{14.20}$$

A growth transformation for  $P_\lambda(\lambda)$  is also a growth transformation for  $R(\lambda)$  since if  $P_\lambda(\xi) > P_\lambda(\lambda) = 0$ , then  $R(\xi) > R(\lambda)$ . To see this:

$$\begin{aligned} P_\lambda(\xi) &= S_1(\xi) - R(\lambda)S_2(\xi) > 0 \\ R(\lambda) &< \frac{S_1(\xi)}{S_2(\xi)} = R(\xi) \end{aligned}$$

Thus, if we have a growth transformation for  $P(\lambda) \triangleq P_\lambda(\lambda)$ , then we also have a growth transformation for our ratio of polynomial  $R(\lambda)$ .

**STEP 2:** Add a constant to  $P(\lambda)$  to ensure nonnegative coefficients. We define a new polynomial:

$$P'(\Lambda) \triangleq P(\Lambda) + C(\Lambda) \quad (14.21)$$

$$\text{where } C(\Lambda) = -a \left( \sum_{i=1}^p \sum_{j=1}^{p_i} \Lambda_{ij} + 1 \right)^d = -a(p+1)^d$$

The values  $a$  and  $d$  are the minimal negative coefficient and degree of  $P$ , respectively. Adding the constant  $a$  to each monomial term ensures that the polynomial  $P'(\Lambda)$  has nonnegative coefficients. At the same time, since  $C(\Lambda)$  can also be seen as a constant of value  $-a(p+1)^d$ , adding it preserves the growth transformation; i.e. the growth transformation of  $P'(\lambda)$  is also the growth transformation of  $P(\lambda)$ .

**STEP 3:** Perform a change of variables to deal with non-homogeneous polynomials. To do this, we form a new polynomial:

$$P''(\Psi) = \Psi_{p+1,1}^d P'(\{\Psi_{ij}/\Psi_{p+1,1}\}) \quad (14.22)$$

where the variable substitution is

$$\Lambda_{ij} = \Psi_{ij}/\Psi_{p+1,1} \text{ and } \Psi_{p+1,1} = 1$$

The result is a new set of constrained simplexes:

$$\Psi \in D' = \psi_{ij} : \psi_{ij} \geq 0, \sum_{j=1}^{q_i} \psi_{ij} = 1$$

$$\text{for } i = 1 \dots p+1, j = 1 \dots q_i, \psi_{p+1,1} = 1$$

To see why this variable substitution creates homogeneous polynomials, consider the example:

$$P'(x, y) = x^2 + xy + x + y + 1$$

$$\text{let } x' = x/z, y' = y/z$$

$$\text{then } P'' = z^2 P'(x', y') = x^2 + xy + zx + zy + z^2$$

Since  $D$  and  $D'$  are isomorphic and there is a bijection between  $\lambda \in D$  and  $\psi \in D'$ , any growth function in  $D'$  for  $P''$  is a growth function in  $D$  for  $P'$ . Finally, since  $P''$  satisfies Baum's conditions, we get our desired growth function. By combining steps 1,2,3 and Baum's theorem, we arrive at Gopalakrishnan's theorem:

**Theorem 14.6. Gopalakrishnan's Theorem (1991):**  $R(\Lambda)$  is a rational function of polynomials in  $\Lambda_{ij}$ . Then there exists a  $a_R$  such that for  $C > a_R$ , the following function  $T^C()$  is a growth transformation in  $D$  for  $R$ :

$$(T^C(\lambda))_{ij} = \frac{\lambda_{ij} \left( \frac{\partial P_\lambda(\lambda)}{\partial \Lambda_{ij}} + C \right)}{\sum_{j=1}^{q_i} \lambda_{ij} \left( \frac{\partial P_\lambda(\lambda)}{\partial \Lambda_{ij}} + C \right)} \quad (14.23)$$

Here  $a_R = ad(p+1)^{d-1}$ ,  $a = \max_\lambda a_\lambda$ , and  $a_\lambda$  is the minimal negative coefficient for all polynomials over all  $\lambda$ .

Note that we have a lower bound on  $C$ . We can prove convergence if  $C$  is large enough, but as  $C$  gets larger, convergence takes a longer time. A heuristic is to choose the least possible value of  $C$  and double it.

Applying Gopalakrishnan's Theorem to MMIE training of discrete HMMs using uniform priors, we get the following update equation for state transition:

$$a_{ij}^{t+1} = \frac{a_{ij}^t \left( \frac{\partial \log Z_\lambda(\lambda)}{\partial a_{ij}} + C(\lambda) \right)}{\sum_{j=1}^{q_i} a_{ij}^t \left( \frac{\partial \log Z_\lambda(\lambda)}{\partial a_{ij}} + C(\lambda) \right)} \quad (14.24)$$

where

$$Z_\lambda = 2^{I_\lambda(m;x)} = \frac{p(x|m, \lambda)}{\sum_{m'} p(x|m', \lambda)}$$

In practice, much of MMIE research has gone into how to efficiently compute the sum in the denominator of  $Z_\lambda$ . The update equations for the other HMM parameters are similar.

## References

- [B86] L. R. BAHL, P. F. BROWN, P. V. DE SOUZA and R. L. MERCER, "Maximum Mutual Information Estimation of HMM Parameters for Speech Recognition" *ICASSP*, 1986
- [BE67] L. E. BAUM and J. A. EAGON, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bulletin of the American Mathematical Society*, (73), pp. 360-363, 1967.
- [B98] J. BILMES "A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," *ICSI Technical Report*, 1998.
- [B87] P. F. BROWN "The acoustic-modeling problem in automatic speech recognition," IBM Research Center, Yorktown Heights, NY, Research Rep. RC-12750, May 1987.
- [CT91] T. COVER and J. THOMAS "Elements of Information Theory", *Wiley-Interscience*, 1991
- [G91] P. S. GOPALAKRISHNAN, D. KANEVSKY, A. NÁDAS and D. NAHAMOO "An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems" *IEEE Trans. on Information Theory*, Vol 37, No. 1, Jan 1991
- [HAH01] X. HUANG, A. ACERO and H. HON, "Spoken Language Processing," *Prentice Hall PTR*, 2001
- [N83] A. NADAS "A Decision-Theoretic Formulation of a Training Problem in Speech Recognition and a Comparison of Training by Unconditional versus Conditional Maximum Likelihood", *IEEE Trans. on Acoustics, Speech and Signal Processing*, **4**, 1983.
- [NJ01] A. Y. NG and M. I. JORDAN "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," *Advances in Neural Information Processing Systems (NIPS) 14*, 2002.
- [NM91] Y. NORMANDIN and S. D. MORGERA, "An Improved MMIE Training Algorithm for Speaker-independent, small vocabulary, continuous speech recognition," *Prentice Hall*, New Jersey, 1978.