Word Confusability - Measuring Hidden Markov Model Similarity

Jia-Yu Chen[1], Peder A. Olsen[2], John R. Hershey[2]

[1]Department of Electrical Engineering, Stanford University,
[2] IBM T. J. Watson Research Center
jiayuc@stanford.edu, {pederao,jrhershe}@us.ibm.com

Abstract

We address the problem of word confusability in speech recognition by measuring the similarity between Hidden Markov Models (HMMs) using a number of recently developed techniques. The focus is on defining a word confusability that is accurate, in the sense of predicting artificial speech recognition errors, and computationally efficient when applied to speech recognition applications. It is shown by using the edit distance framework for HMMs that we can use statistical information measures of distances between probability distribution functions to define similarity or distance measures between HMMs. We use correlation between errors in a real speech recognizer and the HMM similarities to measure how well each technique works. We demonstrate significant improvements relative to traditional phone confusion weighted edit distance measures by use of a Bhattacharyya divergence-based edit distance.

Index Terms: Bayes Error, Bhattacharyya divergence, variational methods, gaussian mixture models, unscented transformation, Kullback–Leibler distance rate.

1. Introduction

The problem of mathematically formulating similarity and distance[1] measures between two HMMs has captured the imagination of scientists since the publication of Juang and Rabiner's paper in 1985, [1]. The two HMMs considered, may differ in topology and transition probabilities, as well as in observation distributions. The Kullback–Leibler distance cannot be directly used because it assigns negative infinity to certain pairs of non-ergodic HMMs whose topology differs. To surmount this problem, Juang and Rabiner defined the Kullback–Leibler Distance Rate (KLDR) as a measure of similarity between ergodic HMM, and they proceded to show how to extend the KLDR to non-ergodic (e.g., left-to-right) HMMs such as occur in speech recognizers. The KLDR has three caveats. First, it is computationally expensive, second it is not a good measure for classification error, and to handle non-ergodic HMMs requires looping them, which is unrealistic.

Many other authors have defined distance measures to compensate for these shortcomings, [2, 3, 4, 5, 6]. In this paper we define some new measures inspired by [7] and [8].

Distance measures between HMMs have been used in areas such as speech recognition, texture image classification, handwriting recognition and machine learning. In speech recognition, HMM distances have been applied to such tasks as vocabulary selection, grammar design, phoneme clustering, measuring language modeling perplexity, locating occurrences of out-of-vocabulary words in an indexed audio database, matching acoustic tags, and pronunciation variation analysis.

This paper discusses the use of distance measures to predict the confusability of two words. Section 2 defines the edit distance and applies it to HMMs. Section 3 shows how to compute distances between GMMs and uses these distances as weights in the HMM weighted distance computation. Finally, Section 4 experimentally compares the distance measures.

[1]The term "distance" is used loosely here and should not be interpreted as a mathematical distance.

2. Edit distance and HMM distances

A word is modeled using an HMM derived from the pronunciation of the word. A word such as **call** may have a pronunciation `K AO L`, and a word such as **dial** a corresponding pronunciation `D AY AX L`. We shall use these two words to exemplify various word confusability measures throughout this paper. Figure 1 shows the HMM for **dial** and **call**. The phonemes are modeled using three-state HMMs, which we have depicted using only one state, for simplicity.

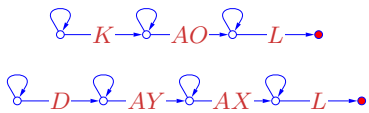

Figure 1: HMMs for **call** with pronunciation `K AO L`, and **dial** with pronunciation `D AY AX L`.

The simplest method to measure the word confusability is to compute the number of corrections, insertions and deletions required to turn one pronunciation into another. This measure is commonly known as the edit, but also as the Levenshtein distance, [9]. For the two example words, the number of edits required is three, as seen in Table 1.

call	**dial**	edit operation	cost
K	D	correction	1
	AY	insertion	1
AO	AX	correction	1
L	L	no operation	0
		total cost	3

Table 1: Edit distance between **call** and **dial**

Computing the edit distance requires finding the minimum number of edits. This can be done by finding the shortest path in the edit graph as shown in Fig. 2.

The edit distance was originally introduced to do approximate string matching. Here we use it in the same way. One natural extension is to put weights on the edges in the edit graph. We

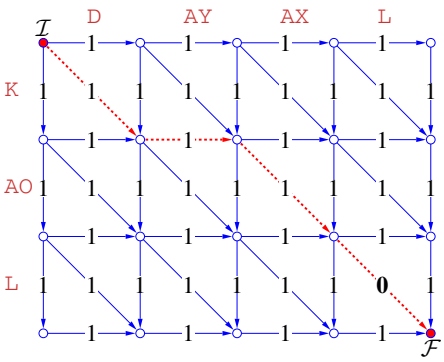

Figure 2: The edit graph to turn the pronunciation of **call** into that of **dial**. The dashed line outlines a path that attains the edit distance. Horizontal lines correspond to insertions, vertical lines to deletions and diagonal lines to corrections.

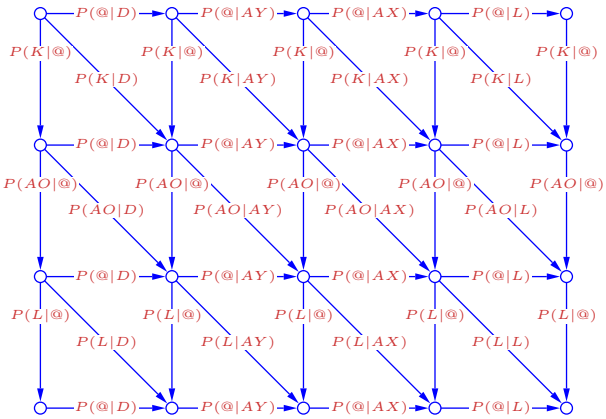

Figure 3: A weighted edit distance between **call** and **dial**. The weights are the negative log of the probabilities in the edit graph.

will use a constant insertion and deletion weight corresponding to the vertical and horizontal edges. For the diagonal correction weights we use, $-\log P(\Phi_1|\Phi_2)$, where Φ_1 and Φ_2 are phonemes in the pronunciations of **call** and **dial**. The product of the probabilities roughly corresponds to the probability of misrecognizing **dial** as **call**. Taking the logarithm and reversing the sign makes the most likely path correspond to the shortest path in the edit distance, where the weights are added instead of multiplied. Figure 3 shows the graph corresponding to the weighted edit distance. We will refer to the edit distance with weights, $-\log P(\Phi_1|\Phi_2)$, as a phoneme–based edit distance.

The phoneme–based distance is purely a function of the pronunciation and does not vary with changes in the acoustic context or the underlying HMM topology for the word. There is another form of edit graph and corresponding edit distance that considers the HMM topology for the two words. The first word is the generating word that synthesizes the acoustic signal, and the second word is the acceptor word, or the recognized word. For our example words, **call** is the generator and **dial** is the acceptor. We define finite state transducers (FSTs) corresponding to the generator and acceptor HMMs as seen in Fig. 4. The cartesian product of the HMMs is the composition of the generator and acceptor that can be seen in Fig. 5. The resulting HMM composition is the state–based edit graph. It differs from the earlier phoneme–based edit graph in two respects. First, it has a number of self–loops that the original edit graph did not have. For computing the shortest (Viterbi) path, the self–loops only add to the overall cost of the path, and so can be ignored. Second, the horizontal and vertical edges are no longer insertions or deletions, but are actually modeled as substitution errors. This is an improvement, as we have no simple method to accurately estimate the insertion and deletion weights in a systematic way. In the new state–based edit graph there are only substitution weights that we can compute from the underlying pair of GMMs.

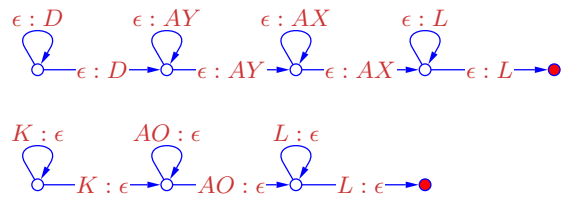

Figure 4: The finite state transducers for **dial** and **call**. ϵ is used as a symbol for the null string.

3. Distances between GMMs

Ultimately we desire to find the classification error or Bayes error when drawing an acoustic sample from one word and finding that the likelihood is larger for the other word. Assuming equal priors on the two word distributions, the Bayes error is defined as

$$\mathrm{B}_e(f,g) \stackrel{\text{def}}{=} \frac{1}{2}\int \min\{f(x),g(x)\}\,\mathrm{d}x. \tag{1}$$

For HMMs the Bayes error is difficult to estimate accurately, but for a pair of GMMs the computational difficulty can be overcome.

In the previous section we reduced the problem to coming up with weights that are functions of the GMM pairs. If we think of the state–dependent edit graph as approximating the likelihood of decoding the acceptor word when given acoustics from the generator word it is natural to use the Kullback–Leibler divergence:

$$D(f\|g) \stackrel{\text{def}}{=} \int f(x)\log(f(x)/g(x))\,\mathrm{d}x. \tag{2}$$

If we wanted to mix the classification approach with the state dependent edit graph approach, we could simply use $-\log \mathrm{B}_e(f,g)$ as weights. It is also possible to use other divergence measures for a weight. In particular we are interested in using the Bhattacharyya measure

$$\mathrm{B}(f,g) \stackrel{\text{def}}{=} \int \sqrt{f(x)g(x)}\,\mathrm{d}x, \tag{3}$$

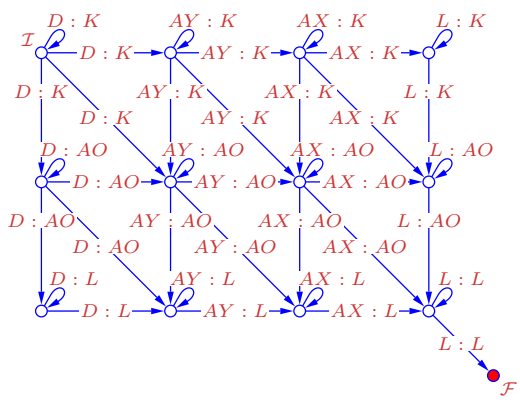

Figure 5: The composition of the FST for **dial** and **call**.

or more specifically $-\log \mathrm{B}(f,g)$ as weights. (Here the Bhattacharyya measure is twice the Bhattacharyya error bound, which includes the priors.)
The Bayes error, Bhattacharyya measure and the KL divergence can not be computed analytically for a GMM pair. We have to resort to Monte Carlo sampling to get approximations to these quantities. The Bhattacharyya and KL divergence can however be computed analytically for a pair of gaussians. This makes it possible to come up with some reasonable analytical approximations to the quantities, [10, 11].
By sampling from the distribution f, we get the following Monte Carlo approximations for the three quantities:

$$\mathrm{MC}_{\mathrm{Bayes}}(f,g) = \frac{1}{n}\sum_{i=1}^{n}\frac{\frac{1}{2}\min(f(x_i),g(x_i))}{f(x_i)} \quad (4)$$

$$\mathrm{MC}_{\mathrm{KL}}(f,g) = \frac{1}{n}\sum_{i=1}^{n}\log(f(x_i)/g(x_i)) \quad (5)$$

$$\mathrm{MC}_{\mathrm{Bhatt}}(f,g) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{g(x_i)}{f(x_i)}}, \quad (6)$$

where $\{x_i\}_{i=1}^n$ are samples from the distribution f.
We will assume that the GMMs f and g have marginal densities that can be written

$$\begin{aligned} f(x) &= \textstyle\sum_a \pi_a \mathcal{N}(x;\mu_a,\Sigma_a) \\ g(x) &= \textstyle\sum_b \omega_b \mathcal{N}(x;\mu_b,\Sigma_b). \end{aligned} \quad (7)$$

For the Kullback–Leibler divergence we have the following variational approximation, [10],

$$D_{\mathrm{var}}(f\|g) = \sum_a \pi_a \log \frac{\sum_{a'}\pi_{a'}\exp(-D(f_a\|f_{a'}))}{\sum_b \omega_b \exp(-D(f_a\|g_b))}. \quad (8)$$

For the Bhattacharyya divergence we have the variational approximation

$$B_{\mathrm{var}}(f,g) = \sum_{ab}\sqrt{\phi_{b|a}\psi_{a|b}}\sqrt{\pi_a\omega_b}B(f_a,g_b), \quad (9)$$

where ϕ and ψ satisfies the constraints $\sum_a \phi_{a|b} = \sum_b \psi_{b|a} = 1$ and are the result of iterating the equations

$$\phi_{b|a} = \frac{\psi_{a|b}\omega_b B(f_a,g_b)^2}{\sum_{b'}\psi_{a|b'}\omega_{b'}B(f_a,g_{b'})^2} \quad (10)$$

and

$$\psi_{a|b} = \frac{\phi_{b|a}\pi_a B(f_a,g_b)^2}{\sum_{a'}\phi_{a'|b}\pi_{a'}B(f_{a'},g_b)^2} \quad (11)$$

until convergence. The details can be found in [11].
Additionally we provide an accelerated Monte Carlo method for estimating the Bhattacharyya measure. By drawing samples from the distribution

$$h = \frac{\sum_{ab}\sqrt{\phi_{b|a}\psi_{a|b}\pi_a\omega_b}\sqrt{f_a g_b}}{\int\sum_{ab}\sqrt{\phi_{b|a}\psi_{a|b}\pi_a\omega_b}\sqrt{f_a g_b}}, \quad (12)$$

we have the variational importance sampling (VISa) estimate

$$\mathrm{VIC}_{\mathrm{Bhatt}}(f,g) = \frac{1}{n}\sum_{i=1}^{n}\frac{\sqrt{f(x_i)g(x_i)}}{h(x_i)}. \quad (13)$$

3.1. Loopy estimates

We defined the Kullback–Leibler divergence and Bhattacharyya measure for GMMs in the previous paragraphs. We can similarly define these for probability density functions (pdfs) on sequences. In particular sequence pdfs F and G defined by the single state HMM in Fig. 6 and Fig. 7 are of particular interest as the KL divergence and Bhattacharyya measure can be computed analytically.

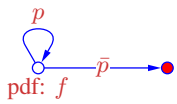

Figure 6: A single state HMM with output distribution f and state transition probabilities p and $\bar{p} = 1 - p$.

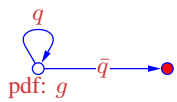

Figure 7: A single state HMM with output distribution g and state transition probabilities q and $\bar{q} = 1 - q$.

The pdfs F and G are implicitly defined on sequences $x = (x_1,\ldots,x_k)$ of length $k = 1$ or greater. For the pdf F the probability for obtaining a sequence of length k is $p(k) \stackrel{def}{=} \bar{p}p^{k-1}$ and the specific probability for the sequence $x = (x_1,\ldots,x_k)$ is:

$$F(x) = p(k)\prod_{i=1}^{k} f(x_i). \quad (14)$$

Similarly for G the probability is $q(k) \stackrel{def}{=} \bar{q}q^{k-1}$ for obtaining a sequence of length k and for the specific sequence the likelihood is

$$G(x) = q(k)\prod_{i=1}^{k} g(x_i). \quad (15)$$

The divergence between F and G can be derived using (2) as follows:

$$\begin{aligned}
D(F\|G) &= \int F(x)\log\frac{F(x)}{G(x)}\mathrm{d}x \\
&= \sum_{k=1}^{\infty}\int p(k)\prod_{i=1}^{k} f(x_i)\log\left(\frac{p(k)\prod_{j=1}^{k} f(x_j)}{q(k)\prod_{j=1}^{k} g(x_j)}\right)\prod_{i=1}^{k}\mathrm{d}x_i \\
&= \sum_{k=1}^{\infty}\int p(k)\prod_{i=1}^{k} f(x_i)\log\left(\frac{p(k)}{q(k)}\right)\prod_{i=1}^{k}\mathrm{d}x_i \\
&\quad + \sum_{k=1}^{\infty}\sum_{j=1}^{k}\int p(k)\prod_{i=1}^{k} f(x_i)\log\left(\frac{f(x_j)}{g(x_j)}\right)\prod_{i=1}^{k}\mathrm{d}x_i \\
&= \sum_{k=1}^{\infty} p(k)\log\left(\frac{p(k)}{q(k)}\right)\prod_{i=1}^{k}\int f(x_i)\mathrm{d}x_i \\
&\quad + \sum_{k=1}^{\infty}\sum_{j=1}^{k} p(k)\int f(x_j)\log\left(\frac{f(x_j)}{g(x_j)}\right)\mathrm{d}x_j \\
&= \sum_{k=1}^{\infty} p(k)\log\left(\frac{p(k)}{q(k)}\right) + \sum_{k=1}^{\infty} kp(k)D(f\|g) \\
&= D(p\|q) + D(f\|g)/\bar{p}.
\end{aligned}$$

A similar computation for the Bhattacharyya measure yields the equation

$$B(F,G) = \frac{\sqrt{\bar{p}\bar{q}}B(f,g)}{1-\sqrt{pq}B(f,g)}. \quad (16)$$

4. Experiments

To measure how well each method predicts recognition errors we used a test suite consisting of short words. For this we chose a spelling task, for which there were 38,921 spelling words (a-z) in the test suite with an average word error rate of about 19.3%. A total of 7,500 spelling errors were detected. Given the errors we estimated the probability of correct recognition $\mathrm{P}(w_1|w_2) = C(w_1,w_2)/C(w_2)$. We discarded cases where the error count was low, the total count was low, or the probability was 1.

For the remaining errors, we compared the various methods, as seen in Figure 8. The figure shows that adding the KL divergence loop estimate to account for the self–loop transition is uniformly better. The Bhattacharyya loop estimate gave a small gain, but not as much as for the KL divergence loop estimate. The best method was the Bhattacharyya VISa estimate with the KL divergence loop estimate.

Figure 9 shows a scatter-plot of the Bhattacharyya VISa score for each pair of letters, versus the empirical measurement. Note that similar-sounding combinations of letters appear on the lower left (e.g. "c·z"), and dissimilar combinations appear in the upper right (e.g. "a·p").

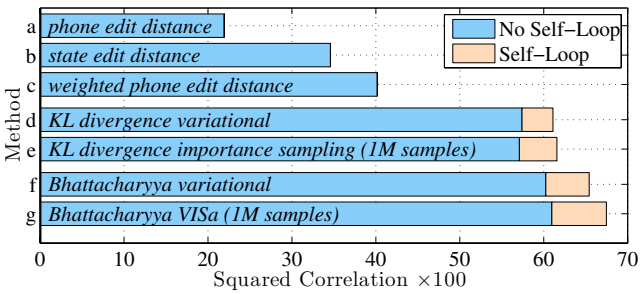

Figure 8: Squared correlation coefficient between the empirical negative log error rate, and each of the confusability scores. The squared correlation represents the percent of the empirical variance that is explained by each of the scores.

5. Conclusion

We have shown in this paper how we can apply the edit distance framework to HMMs and use GMM based divergence or distance measures to define an HMM based divergence score that correlates well with the type of errors the speech recognizer makes. Overall the best measure used the Bhattacharyya divergence together with the KL–based self–loop transition. This system significantly outperforms the standard weighted phone edit distance. The parameters can be estimated directly from the acoustic model, so there is no need for any training data.

6. References

[1] B.-H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Technical Journal*, vol. 64, no. 2, pp. 391–408, February 1985.

[2] Ling Chen and Hong Man, "Fast schemes for computing similarities between gaussian HMMs and their applications in texture image classification," *EURASIP Journal on Applied Signal Processing*, vol. 13, pp. 1984–1993, 2005.

[3] Matti Vihola, Mikko Harju, Petri Salmela, Janne Suontausta, and Janne Savela, "Two dissimilarity measures for HMMs and their application in phoneme model clustering," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Orlando, Florida, May 2002, vol. I, pp. 933–936.

[4] Claus Bahlmann and Hans Burkhardt, "Measuring HMM similarity with the Bayes probability of error and its application to online handwriting recognition," in *Sixth International Conference on Document Analysis and Recognition (ICDAR'01)*, 2001, pp. 406–411.

[5] Maruf Mohammad and W. H. Tranter, "A novel divergence measure for hidden Markov models," in *Proceedings IEEE Southeast Conf.*, April 2005, pp. 240–243.

[6] Markus Falkhausen, Herbert Reininger, and Dietrich Wolf, "Calculation of distance measures between hidden Markov models," in *Proceedings of Eurospeech 1995*, Madrid, 1995, pp. 1487–1490.

[7] Harry Printz and Peder Olsen, "Theory and practice of acoustic confusability," *Computer, Speech and Language*, vol. 16, pp. 131–164, January 2002.

[8] Jorge Silva and Shrikanth Narayanan, "Average divergence distance as a statistical discrimination measure for hidden Markov models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 890–906, May 2006.

[9] Vladimir I. Levenshtein., "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Phys. Dokl*, vol. 10, no. 8, pp. 707–710, 1966.

[10] John Hershey and Peder Olsen, "Approximating the Kullback Leibler divergence between gaussian mixture models," in *Proceedings of ICASSP 2007*, Honolulu, Hawaii, April 2007, to appear.

[11] Peder Olsen and John Hershey, "Bhattacharyya error and divergence using variational importance sampling," in *Proceedings of Interspeech 2007*, August 2007.

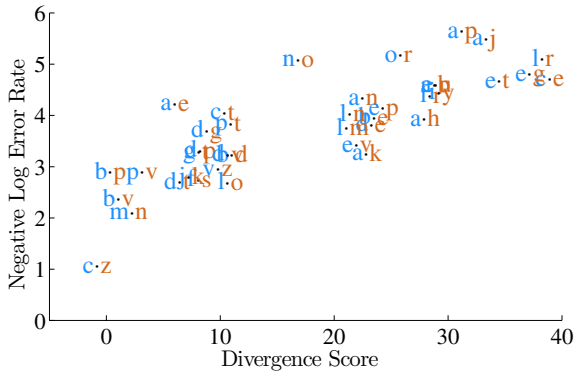

Figure 9: The negative log error rate for all spelling word pairs compared to the Bhattacharyya score with transition probabilities.