Improved Estimation of Hidden Markov Model Parameters from Multiple Observation Sequences

Abstract

The huge popularity of Hidden Markov models in pattern recognition is due to the ability to "learn" model parameters from an observation sequence through Baum-Welch and other re-estimation procedures. In the case of HMM parameter estimation from an ensemble *of observation sequences, rather than a single sequence, we require techniques for finding the parameters which maximize the likelihood of the estimated model given the entire set of observation sequences. The importance of this study is that HMMs with parameters estimated from multiple observations are shown to be many orders of magnitude more probable than HMM models learned from any single observation sequence — thus the effectiveness of HMM "learning" is greatly enhanced. In this paper, we present techniques that usually find models significantly more likely than Rabiner's well-known method on both seen and unseen sequences.*

1. Introduction

The successful application of Hidden Markov Models (HMMs) to diverse applications such as speech recognition [1, 2, 3, 4], face recognition [7], handwriting recognition [6], and gesture recognition [5] demonstrates the immense utility of the HMM as a workhorse for spatio-temporal pattern recognition. The usefulness of the HMM stems from the ability to learn HMM parameters from observation sequences through the Baum-Welch reestimation procedure, and the consequent ability to provide a form of context handling in pattern recognition tasks.

When N observation sequences are known to arise from the same model, then the "Third Problem" outlined in Rabiner's work (see [1] Chapter 6) is to find the model parameters so that the model has a high likelihood of generating all N observation sequences. For example, in the case of speech recognition, we may have N examples of a speaker saying a certain word and we need to find an HMM that has high likelihood of generating all N of these speech signals.

Rabiner describes a method where all N observation sequences are used at each step of the Baum-Welch reestimation procedure to produce a single HMM parameter estimate. Here we propose a class of new estimation methods where the Baum-Welch reestimation procedure is run separately on the N observations. In this paper, we investigate a number of alternative techniques to combine the N resulting models to produce a final model to matches the N observation sequences.

2 Same-Structure HMM Estimation

A hidden Markov model ([1] chapter 6) consisting of a set of n nodes, each of which is associated with a set of m possible observations (the structure of the model). The parameters of the model include an initial state π which describes the distribution over the initial node set, a transition matrix a_{ij} for the transition probability from node i to node j conditional on node i, and an observation matrix $b_i(O_m)$ for the probability of observing symbol m given that the system is in state i. Rabiner uses $\lambda = (a, b, \pi)$ to denote the model parameters.

For each model M_k inferred from a sequence S_k (generated from a single source model M_S), there is an associated probability $\hat{P}_k$ of that model producing the sequence S_k. Similarly, we may define $\hat{P}_k^{all}$ as the probability of that model k generating all sequences S_k, for all $k = 1, \ldots, n$. We can also define G_k and G^{all} for the probabilities of generating sequences given the original generating model as indicated below.

$$
\begin{aligned}
P_k &= P(k^{th}\text{training seq. } | \text{ model for seq.}k) \\
P_k^{all} &= P(\text{all training seq. } | \text{ model for seq.}k) \\
G_k &= P(k^{th}\text{observation seq. } | \text{ generating model}) \\
G^{all} &= P(\text{all observation seq. } | \text{generating model})
\end{aligned}
$$

Two algorithms of major importance are the Forward Algorithm and Backward Algorithm [1], which fall in the category of HMM algorithms for evaluating probabilities.

The Forward Algorithm calculates the probability α_t of a sequence of observations $O_1, \ldots, O_t$. The Backward Algorithm calculates the probability β_t of observa-

tions from a sequence of length n from times t until n, i.e. $O_t + 1, \ldots, O_n$.

The Baum-Welch algorithm is an "iterative update" algorithm which constructs a hidden Markov model of specified structure which best fits a given observation sequence.

An important issue in this paper is the thresholds used by the Baum-Welch algorithms - namely the convergence level for P_{all} (the probability of generating all training sequences) on consecutive re-estimations. Another important issue is the $maxloops$ parameter which is the maximum number of permitted re-estimation calculations. These were adjusted to the point where the results of the algorithm were no longer affected significantly by the threshold levels.

Code reliability was established by running a series of tests on the final algorithm, including parallel tracing, and executing the Baum-Welch algorithm on a single observation sequence and comparing the results with Rabiner's multiple-sequence merge, supplied with an identical observation sequence, and also with two copies of that same sequence.

3. Methodology

A parameter estimation method for a set of hidden Markov models produces a new hidden Markov model with that same structure, but with different transition a_{ij} and observation $b_i(O_m)$ probabilities. Method evaluation is done using elementary Monte Carlo techniques [8]. The methodology of this paper is to calculate the relative strengths of each estimated model using the product of the probabilities of generating a set of unseen data from the unknown hidden Markov model.

A set of 50 initial generating models was used to generate 20 observation sequences, each of length 5 (short sequences are better suited to Left-Right models). These 20 sequences were then used to train a HMM using the range of vector learning techniques being compared in this paper. The inferred HMM was then evaluated using a set of 20 unseen observation sequences generated by the same initial generating model.

A single model was randomly generated, containing 3 states and 4 possible observation values 1. The model structure was upper triangular in the transition matrix a_{ij} (required by the Left-Right property). Finally, observation probabilities were assigned to each state.

Short sequence merging and and long sequence merging were tested in the Left-Right case. In the Left-Right case, Baum-Welch re-estimation was used with a randomly selected Left-Right model. In the cyclic case, a general random model was used and longer sequences were used.

This paper only includes the results of the short-sequence Left-Right model as this is their most useful form in most applications.

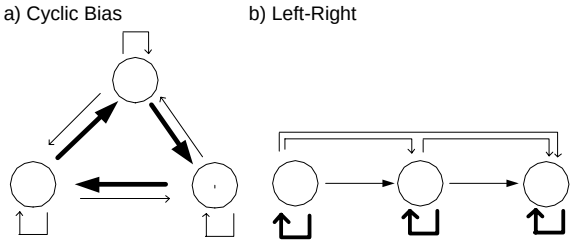

Figure 1. Cyclic and Left-Right structures. Bold arrows indicate higher probabilities on average. No arrow connecting a vertex pair indicates a forbidden (zero-probability) transition.

The final two estimation methods in the list below are *iterative update methods* in which the information in the training sequence set is *learned* using multiple-sequence Baum-Welch. The method as a whole, using a vector of observation sequences to iteratively update a single HMM is described by Rabiner [1]. The re-estimation formula for the two iterative update methods as follows (reproduced from [1]):

$$\overline{a}_{ij} = \frac{\sum_k W_k \sum_{t=1}^{T_k} \alpha_i^k a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^{(k)}(j)}{\sum_k W_k \sum_{t=1}^{T_k} a_t^k(i) \beta_t^k(i)} \quad (1)$$

$$\overline{b}_{ij} = \frac{\sum_{k=1}^{K} W_k \sum_{O_t(k)=v_j} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{K} W_k \sum_{t=1}^{T_k} \alpha_t^k(i) \beta_t^k(i)} \quad (2)$$

All other methods use single-sequence Baum-Welch, combine the resulting inferred models directly and then take a simple weighted average of the model parameters as shown below:

$$\overline{a}_{ij} = \sum_k \frac{W_k}{N_a} a_{ij}^{(k)} \quad (3)$$

$$\overline{b}_{ij} = \sum_k \frac{W_k}{N_b} b_{ik}^{(k)} \quad (4)$$

$$\pi_i = \sum_k \frac{W_k}{N_\pi} \pi_i^{(k)} \quad (5)$$

where W_k is the weighting factor for each sequence, N_a, N_b, N_π are normalisation factors and where $\lambda^{(k)} = (a^{(k)}, b^{(k)}, \pi^{(k)})$.

The estimation techniques used were as follows:

- Direct parameter averaging across models, $W_k = 1$
- Direct parameter averaging across the best 50% in terms of their $P_{all,k}$ score, with $W_k = 1$. The 50 models under this criteria were ranked, and the top 50% were selected (50% Windsorised Level)

- Windsorised method, maximized over the full range of percentiles. Maximization was done in terms of the training observation sequences since this stage of selecting the maximum was part of the method itself.
- Direct parameter averaging over the top 50% in terms of $P_{all,k}$, weighted by $W_k = P_{all,k}$
- Windsorised method, maximized over the percentile thresholds and weighted by $W_k = P_{all,k}$ Once again maximization was done in terms of the training (seen) observation sequences
- Parameter averaging of all models, $W_k = 1/P_k$
- Parameter averaging of all models, $W_k = P_k^{all}$
- Parameter averaging of all models, $W_k = P_k$
- Parameter averaging of all models, $W_k = 1/P_k^{all}$
- Rabiner's Vector Learning method [1] which incorporates unit weighting and re-estimation using multiple observation sequences at every stage of a single re-estimation operation
- Rabiner's Vector Learning method [1] which incorporates $1/P_k$ unit weighting and re-estimation using multiple observation sequences at every stage of a single re-estimation sequence.

The entire process was repeated for 50 initial generating models, and the average fit probability for each of the above methods (over all 50 initial generating models and each best-fit sequence models) was taken. The correctness of the implementation was verified using a range of tests including careful debugging and variable tracing in addition to the following:

- Comparison of Baum-Welch and Multiple-Sequence Baum-Welch on a single observation sequence
- Comparing Multiple-Sequence Baum-Welch on a single observation sequence, with Multiple-Sequence Baum-Welch with two or more copies of that same observation sequence

4 Comparison of Estimation Techniques

The performance of the final estimated model for each method was evaluated on *unseen* data, thereby providing a reliable test of the approximation to the initial generating model. The log probability mean results shown in figure 2 are generated from 20 unseen sequences and are averaged over all 50 final estimated models, one for each initial generating model.

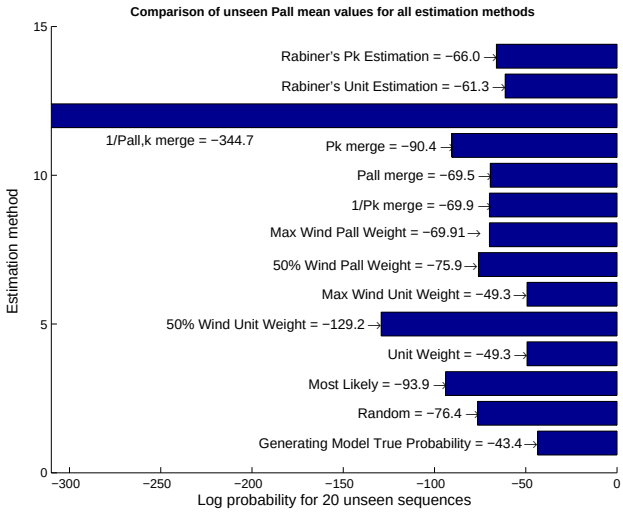

Figure 2. Unseen P_{all} mean values

Figure 3 shows a closer comparison of the best merge methods for short sequences in Left-Right models. Density estimation was performed using a Gaussian kernel [9].

The two Windsor-percentile maximization methods were evaluated in terms of the merged model's performance on unseen data, even though the selection of the percentile threshold in these cases was done in terms of seen data. Notices that as is expected, P_{true} is better than all the others. The best-performing procedure in this experiment was the maximized Windsorisation method, with unit weighting. The $1/P_k$ weighting method advocated by Rabiner [1] performs well, but not as well as the maximized Windsorisation with either unit or $P_{all,k}$ weighting, or even as well as 100% Windsorisation with unit weighting. This relative weakness may be a result of the inability to filter the impact of various sequences on the final result and will be the subject of further research.

5 Summary

In all our trials, the $P_{all,k}$-weighted Windsorised estimations showed very little variation with the Windsorisation level, which suggests that weighting the terms in this way makes high-probability terms much more significant in the Windsorisation, and hence removes any sensitivity to Windsorisation level.

These results suggest that overspecialisation of the learned model to the training data is important in determining the effectiveness of learned models. This may be the reason that the Most Likely model performs worse than the Random Left-Right model in generating the unseen observations.

It was also found that Rabiner's method was more sen-

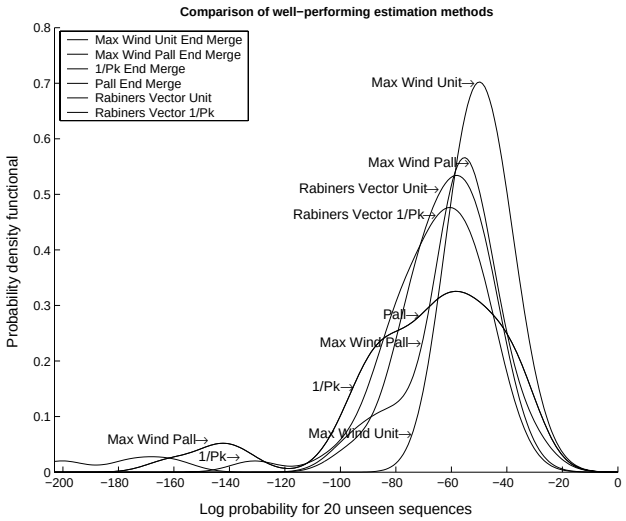

Figure 3. Best methods

sitive to the choice of initial random model in the multiple-sequence Baum-Welch re-estimation procedure.

The experiment was repeated for other forms of initial generating model, including Left-Right models with observation sequences much longer than the number of states, and for cyclic models. The same patterns were observed in all cases.

These results suggest that weighting over-emphasizes the high-$P_{all,k}$ models, in a sense reducing the number of models being used in the average. This would tend to imply that the 'effective Windsorisation Level' is no longer maximal, in an 'effective' sense. The superiority of the Maximized Unit-Weight Windsorisation Estimation Model is encouraging.

It has been demonstrated that Rabiner's method of vector learning is more easily affected by the choice of initial generating model and so it is not as robust as its unit-weighted alternative. Our results also suggest that Rabiner's re-estimation method and its (slightly superior) unit-weighted variant suffer from the problem of local minima trapping (see figure 3 in which the 'trapped' re-estimation runs appear in the low-probability part of the curve).

Our proposed Unit Weighted Maximised Windsorisation method avoids both problems by discarding bad HMMs (on a sequence-by-sequence basis) and by avoiding local minima traps with the use of multiple re-estimation runs. Future work will aim towards a complete investigation of the parallel re-estimation problem and a comparison of these algorithms with existing methods for practical applications.

References

[1] L. R. Rabiner, B. H. Juang, *Fundamentals of Speech Recognition* New Jersey Prentice Hall, 1993.

[2] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition", *The Bell System Technical Journal*, 1075-1106, Vol. 62, No. 4, April 1983

[3] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, *An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition*, The Bell System Technical Journal 1035-1074, Vol. 62, No. 4, April 1983

[4] M.K. Ravishankar, *Efficient Algorithms for Speech Recognition*, Doctor Thesis, Technial Report CMU-CS-96-143, Pittsburgh, USA, 1996

[5] Christopher Lee, Yangsheng Xu. "Online, Interactive Learning of Gestures for Human/Robot Interfaces." *1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN. vol. 4, pp 2982-2987.

[6] T. Starner and A. Pentland. "Visual recognition of american sign language using hidden markov models", *International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[7] A. V. Nefian and M. H. Hayes, "An embedded HMM approach for face detection and recognition," *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 6, pp. 3553–3556, March 1999.

[8] Statist. 41 337–348. Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications" *Biometrika* 57 97–109.

[9] Silverman, B.W., *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London and New York, 1986.

[10] . Neyman and E. S. Pearson. "On the problem of the most efficient tests of statistical hypotheses". *Phil. Trans. Roy. Soc.*, Ser. A., 231:289–337, 1933.