EFFICIENT HIGH-ORDER HIDDEN MARKOV MODELLING

J.A. du Preez and D.M. Weber
Department of Electrical and Electronic Engineering, University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa,
Email: dupreez@firga.sun.ac.za

Abstract

We present two powerful tools which allow efficient training of arbitrary (including mixed and infinite) order hidden Markov models. The method rests on two parts: an algorithm which can convert high-order models to an equivalent first-order representation (ORder rEDucing), and a Fast (order) Incremental Training algorithm. We demonstrate that this method is more flexible, results in significantly faster training and improved generalisation compared to prior work. Order reducing is also shown to give insight into the language modelling capabilities of certain high-order HMM topologies.

1 Introduction

A number of researchers (e.g. [1, 2]) have noted the potential for, and the computational cost of, high-order hidden Markov models (HMM). Prior work [2, 3] derived second-order extensions to HMM training algorithms, but their approaches required new training algorithms for each increase in model order. This work is very different in that we are able to show that any high-order model (fixed- and mixed-order) may be converted to an equivalent first-order model using our ORder rEDucing (ORED) algorithm [5]. ORED provides a unifying paradigm for reasoning about HMMs of any order because it makes the relationship between HMM topology and HMM order explicit. Using this insight, HMMs can be designed using higher-order specifications and then reduced to make its topology explicit using a first-order equivalent model. ORED also allows any standard first-order HMM training algorithm to train and otherwise manipulate arbitrary order HMMs. The ORED algorithm also provides a powerful opportunity for efficient training of otherwise computationally intractable high-order HMM systems by Fast Incremental Training (FIT). Details of the ORED and FIT algorithms can be found in [5] and [6]. Application of these techniques to automatic language recognition can be found in a companion paper [4].

Section 2 introduces the notion of high-order HMMs. Sections 3 and 4 outline the ORED and FIT procedures respectively. Section 4.2 discusses how these can be applied to important language modelling issues such as context and duration, and Section 5 provides quantitative evidence of the efficiency of this approach.

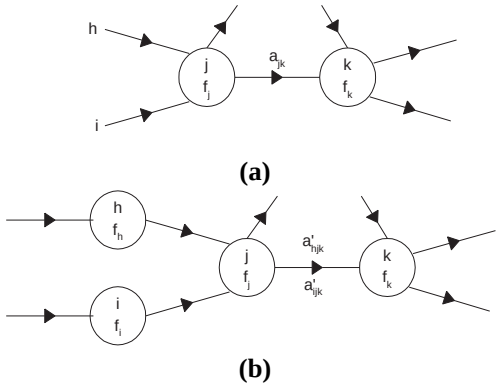

Figure 1: First (a) and second (b) order HMMs indicating the increased history of state transition dependence of the latter. The symbol f_i is the pdf associated with state i.

2 High-order HMMs

First-order HMMs [1] are characterised by a set of N emitting states. Each state S has an associated probability density function (pdf) denoted as f_i which quantifies the similarity between an input feature vector $\mathbf{x}$ and S. States are coupled by transition probabilities. For first order HMMs, the transition probabilities are indicated by the symbol a_{jk}. This indicates the probability of making a transition to state k given that the current state is j. High-order HMMs state transition probabilities depend on two or more prior states and are characterised by probabilities with three or more indices. For a second order HMM, a_{ijk} indicates the probability of jumping to state k given that the current state is j and the prior state is i. Paths between pairs of states thus have multiple transition probabilities, each depending on prior states. This is illustrated in Figure 1.

When converting high-order models to lower-order equivalents, we avoid ambiguity by indicating the high-order model transition probabilities with a prime (a'_{ijk}) to distinguish them from the corresponding lower-order transition probabilities. The pdfs and state transition probabilities of a given HMM are usually determined from training

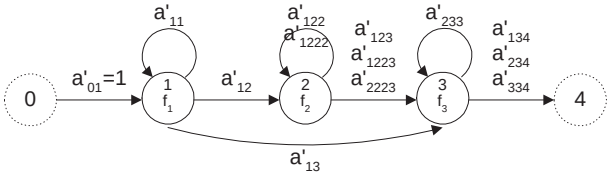

Figure 2: A mixed-order left-to-right HMM with a maximum order of 3. The stippled states (0 and 4) are initial and terminal null states [5].

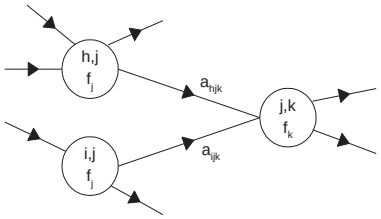

Figure 3: First order equivalent of Figure 1(b). Note that each state transition now has only one transition probability associated with it.

data using various algorithms such as the Baum-Welch re-estimation equations [1]. These algorithms can either be generalised to high-order cases, or, as we propose here, the high-order models can be reduced to equivalent first order models for training. Unfortunately, the number of transition probabilities in high-order models grows with the power of the order of the model. Conventional training procedures rapidly become computationally intractable due to processor speed and memory constraints. In spite of these problems, the additional state "memory" associated with high-order HMMs offer compelling and powerful modelling capabilities.

2.1 Fixed, mixed or infinite?

The length of state sequence memory determines the order of the HMM, and an HMM of given order may contain state transition probabilities that depend on different numbers of prior states. In the case of fixed-order models, all state transition probabilities (for a fixed third-order HMM) are all of the form a_{ijkl}. Mixed-order models have state transition probabilities with different numbers of indices, indicating the different Markov orders within the process. Figure 2 illustrates a mixed order HMM. In Section 4.2, we will show how mixed order HMMs can be used for phoneme duration modelling.

In order to model some characteristics of language (such as phoneme context), we found it necessary to model the sequence of states visited, irrespective of the number of consecutive repetitions of each state. This results in an HMM of infinite-order (i.e. a special case of a more general class of infinite-order HMMs) as we now discuss. Some infinite order HMMs contain special state transition probabilities which depend on the sequence of states visited, independent of how many times a particular state was visited. For example, the transition probability a_{ij^+k} is the probability that the next state will be k, given that the model was in state j for one or more previous time slots and prior to entering state j, the state was i. The notation j^+ thus indicates the occurrence of one or more j's in the subscript. This is a powerful tool for phoneme context modelling as outlined in Section 4.2.

3 ORED algorithm overview

First-order transition probabilities involve only the two states that are joined by them (see Figure 1(a)). In contrast, the second-order dependence of a'_{ijk} on state i, cannot be inferred from its adjoining states but is only encoded in the subscripts of the transition probability itself. We now create a new model with states corresponding to pairs of linked states from the original model, as is illustrated in Figure 1(b). Each state shares the same pdf as the second one of the original pair of states does (these are called tied pdf states). Transition probabilities are inserted between the states that respectively match the first and the last two subscripts of the transition probability e.g. a_{ijk} is inserted between states (i, j) and (j, k). This is a twofold Cartesian product of the states involved [3]. Now the indexes of the states adjoining the second-order transition probability fully describe the subscripts of this transition probability. This effectively means that we can now interpret a_{ijk} as a first-order transition probability joining states (i, j) and (j, k). By enlarging the number of states in the way we did, we were able to reduce effectively the order of the model by one, without losing any representational capability. This simple observation forms the basis of the ORED algorithm. This concept may be extended to reducing N-order HMMs to $N - 1$ order HMMs. Recursive application of this procedure can convert models of arbitrary order to first order equivalents. The proper handling of higher orders, initial conditions and the mixed-order models introduces quite a few intricacies to the algorithm that are not evident from the intuitive notion on which it is based. These complexities are detailed in [6]. On a practical level, ORED allows the application of any standard HMM algorithm to any higher-order HMM, thus greatly enhancing the usefulness of this technology. Although concerns have been expressed about the effect this increase in the number of states might have [2], we can show [6] that it does not contribute in any way to additional computational requirements.

Figure 4 shows a first order equivalent of the mixed-order model illustrated in Figure 2. A little thought will reveal the equivalence of these models.

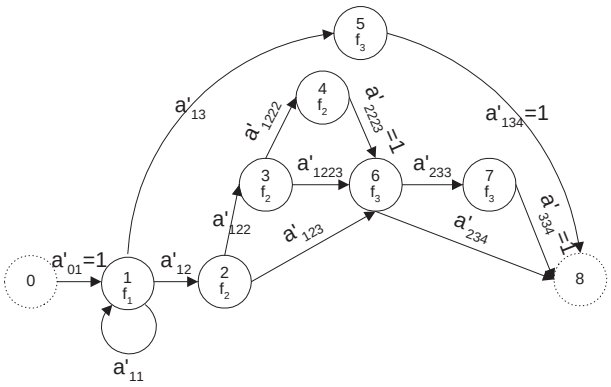

Figure 4: The first order equivalent of the HMM illustrated in Figure 2.

4 Fast Incremental Training

As already mentioned, high-order HMMs can be vastly more expensive than their first-order counterparts. The processing and memory requirements are serious issues that can easily place such a model outside the available computing capacity. Fortunately, the transition structure of a high-order HMM is usually quite sparse. Because, prior to training, it may be difficult to determine which transitions will be redundant, training normally commences with all the transitions that are potentially useful. For many problems, considerable training effort is therefore expended on estimating parameters that will eventually become zero. Referring back to Figure 1, it can be seen that a single transition probability in the lower-order model is simply being replaced by a set of refined probabilities in the higher-order model. Considerable computational effort can be avoided if the training of redundant sets of higher-order probabilities can be eliminated by noting which corresponding lower-order probabilities are zero. This observation forms the basis of the FIT algorithm.

4.1 FIT for fixed-order HMMs

The FIT training algorithm for fixed order models is now outlined:

1. Set up a first-order HMM for the application at hand.
2. Run a standard training algorithm (e.g. Baum-Welch) on the first-order model. Non-viable (i.e. zero probability) transitions will disappear.
3. Convert the optimised first-order model to a second-order model by expanding the subscripts of the remaining non-zero transition probabilities with one extra prior state. These expanded transition probabilities are initialised with the value of the lower-order transition probability they were extended from.
4. Use the ORED algorithm to create a first-order equivalent of this model.

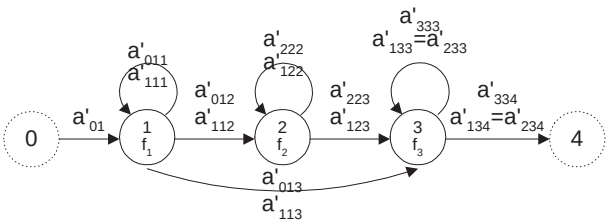

Figure 5: Duration modelling left to right second-order HMM with one state skip.

5. Repeat step 2 to train this model. This will refine the transition probabilities to their required higher-order values. Repeating this process trains successively higher-order models.

This algorithm as described above does not allow the training of mixed and infinite order HMMS, however, for certain specific useful mixed- and infinite-order models, we have formulated extensions to the ORED/FIT algorithm [6].

4.2 Mixed-order variants for language modelling

We address two applications of high-order HMMs to language modelling: phoneme duration and context. While fixed order HMMs can be used to model both duration and context, mixed and infinite order models provide more powerful paradigms as we now discuss.

Duration modelling: With the following example we want to illustrate a topology arising from emphasising the duration modelling aspects of the model while neglecting the contextual modelling that is not directly involved with the modelling of the duration of a state. This type of modelling only involves states with self-loops. In such a state we need to identify the sets of departing transition probabilities that share the same destination and involve the same number of repetitions of this state. If all the transition probabilities in such a set are constrained to be identical regardless of prior states, the resulting model will model the duration of this state. Figure 5 illustrates a second-order model constructed using these rules. Figure 6 shows the first-order equivalent (as determined by the ORED algorithm) of Figure 5. Note the similarity between this and the model proposed by Ferguson [7].

Context modelling: Finite (N-th) order HMMs build context memory over the prior N states which can severely limit their ability to model things like phoneme context. The use of infinite order HMMs can address this by allowing state transition probabilities to be independent of how many times a particular state was visited. Figure 7 shows a simple context modelling HMM. Note that the probability of a state transition from state 2 to state 3 is independent of how many times states 1 and 2 were visited. Figure 8

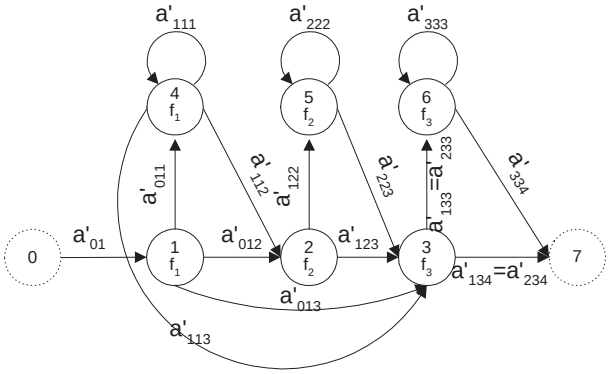

Figure 6: First order equivalent of the duration modelling HMM in Figure 5. Note the similarity to the Ferguson duration model [7].

shows the first order equivalent of Figure 7.

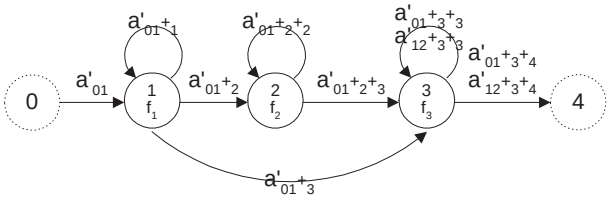

Figure 7: Third contextual order left to right HMM with one state skip. The notation k^+ is used to indicate one or more occurrences of index k.

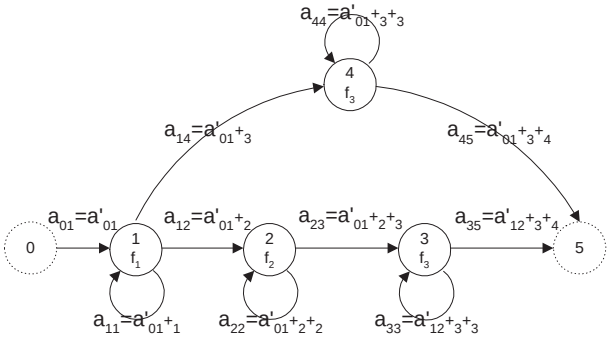

Figure 8: First-order equivalent of Figure 7.

Combined context and duration modelling: This approach also allows both context and duration modelling simultaneously, as demonstrated in [4]. Details are available in [6].

5 Performance and conclusion

We used well-controlled synthetic simulation experiments to investigate the performance of FIT models relative to that of the extended/ORED approach [5]. Each data set consisted of training and testing data drawn from randomly generated high-order HMM sources. The number of non-zero state transition probabilities in the underlying models, as well as the expected optimal classification performance[1] were thus known. Compared to the known underlying model, experiments on the training of a fourth-order HMM with eight underlying states, the FIT/(conventional) algorithm revealed: 15%/(1820%) more non-zero transition probabilities and a 48%/(170%) increase in classification error on independent test data. The dramatic increase in non-zero transition probabilities experienced by the conventional approach indicates the lack of generalisation compounding the enormous computational cost of this method. Detailed results for fixed order HMMs are available in [5].

The ORED/FIT approach was thus found to result in models with fewer non-zero state transition probabilities, better generalisation and better classification performance compared to prior (conventional) training algorithms. The ORED approach also provides invaluable insight into the topological properties of a broad class of high-order HMMs.

References

[1] Deller, J.R., Proakis, J.G. and Hansen J.H.L. *Discrete time processing of speech signals*. Macmillan, 1993.

[2] Mari, J.-F., Haton, J.-P. and Kriouile A., "Automatic word recognition based on second-order hidden Markov models". *IEEE Transactions on Speech and Audio processing*, vol. 5 no. 1, pp. 22 - 25, 1997.

[3] He, Y. "Extended Viterbi algorithm for second-order hidden Markov process". *Proceedings of the IEEE 9th International Conference on Pattern Recognition*, pp. 718 - 720. Rome, Italy, 1998.

[4] Du Preez, J.A. and Weber, DM. "Automatic language recognition using high-order HMMs", *Proceedings of the IEEE International Conference on Speech and Language Processing*, 1998.

[5] Du Preez, J.A. "Efficient training of high-order hidden Markov models, using first-order representations". *Computer Speech and Language*, vol. 12, pp. 23-39, 1998.

[6] Du Preez, J.A., *Efficient high-order hidden Markov modelling. PhD Dissertation*, University of Stellenbosch, South Africa, 1998.
URL: http://dsp.ee.sun.ac.za/reports

[7] Ferguson, J.D. "Variable duration models for speech". *Proceedings of the Symposium on the Application of Hidden Markov Models to Text and Speech* (editor: J.D. Ferguson), pp-143-179, Princeton, New Jersey.

[1] Access to the underlying synthetic models allows the best possible classification results to be obtained.