Recognizing mobile organizations from column formations using Hierarchical Hidden Markov Models: a simulation experiment

Fredrik Stavfors[a] and Per Svensson[b,1]

[a]Dept. of Numerical Analysis and Computer Science, KTH, Stockholm, Sweden

[b]Dept. of Data and Information Fusion, FOI, Stockholm, Sweden

email: Per.Svensson@foi.se

Abstract - *In a paper at Fusion 2000, Björnfot and Svensson discussed the use of HMMs to recognize mobile military organizations moving in column formation along a road from observations of the types and relative positions of its component vehicles. Straightforward use of standard HMMs in the sequence recognition task restricts the application of this technique to the rather unlikely situation where all vehicle types normally follow each other according to a fixed sequential pattern. In this paper, this restriction is relaxed by representing a multi-level organization by an Hierarchical Hidden Markov Model, in which each possible component sequence for each organizational subunit type is modelled by a separate "sub-HMM". Experiments are carried out using simulated data which illustrate the increased flexibility and applicability of this approach in column recognition tasks.*

Keywords: Force aggregation, ground target recognition, vehicle column recognition

1. Introduction

In [1], Björnfot and Svensson discussed the concept of using HMMs to recognize military organizations moving in column formation along a road from observations of the types and relative positions of its component vehicles. In the course of that work, it was noted that the use of additional *a priori* knowledge about hierarchical organization structure should help in the recognition task. The straightforward use of standard HMMs in a vehicle sequence recognition task restricts the application of the technique to the rather unlikely situation where all vehicle types normally follow each other according to a fixed sequential pattern. Also, it precludes the application of the technique in the probably quite common situations where vehicle types would be allowed to switch positions while staying within the spatial extent of their local organizational unit.

In this paper, these restrictions are relaxed. The basic idea used is to represent as an HMM submodel each possible component sequence for each organizational unit type. These are then connected together into an hierarchical HMM (HHMM) structure. Experiments are carried out using simulated data which illustrate the increased flexibility and applicability of the HHMM approach in column recognition tasks.

Recently, efficient algorithms for learning and inference in hierarchical HMMs have been proposed [2,3,4]. In these papers, as well as in the theses [5,6], improved representation and algorithmic techniques are described. In particular, in [5] Murphy shows how to embed a large number of special cases of HMMs into the general framework of Dynamic Bayesian Networks (DBN). This shift of perspective also encouraged the application of fast algorithmic techniques developed for Bayesian networks to HHMM training and recognition [3].

Since our goal in this work is primarily to study the benefits of using the hierarchical approach in representing and solving the column recognition problem *per se* rather than exploring HHMM algorithms in general, we decided to approach our task by using essentially the same basic HMM learning and recognition algorithms as in [1], but structure the problem statement and computations differently.

[1]Corresponding author.

In: Proc. 8th Int. Conf. on Information Fusion, Philadelphia, PA, USA, July 25-29, 2005.

To deal with our first issue, representing hierarchies, we assume that we have available a set of training samples which provide:
1. “bags” of platoon types each of which define a company type (see Sec. 2),
2. bags of vehicle types each of which define a platoon type.

For comparison, we also assume that a set of samples is available where each sample is a sequence of vehicles which defines a company type. This set should cover essentially all relevant vehicle sequences. In order to carry out this experiment comfortably on a 1.5 GHz PC, we decided to use somewhat smaller test examples than are likely to be encountered in practical intelligence work, specifically, to use templates with three vehicles per platoon and three platoons per company.

Important issues that need to be addressed when proposing the use of HMM-based recognition methods to the problems described above are:
- what kinds of changes in relation to distributions expected *a priori* are likely to occur in different practical situations?
- will the method be robust with respect to small such changes ?
- will the method be robust with respect to missing, extraneous, or misclassified data?
- are there effective techniques that can be used to learn vehicle sequence distributions from data, or generate them from known rule sets such as military transportation doctrines?

In this paper we give partial answers to these questions by presenting experimental evidence which supports the following conclusions:
- HHMM recognition methods can be useful in situations where the order between “symbols” is not known in advance, given that it is possible to learn or predict whole families of permissible alternate sequences and these sequences are relatively short. In our application, symbols correspond to vehicle and platoon types. Note that in the general unordered case, computational complexity grows exponentially with the length of the sequence;
- such sets of sequences can probably be used to model military organizations as long as the number of subunits within each unit is relatively small, say less than 10;
- standard HMM learning and recognition algorithms can be adapted to cover this case;
- these methods can be made reasonably tolerant of missing, extraneous, and misclassified data.

1.1. Related work

In [7], HHMMs are used in a different tactical fusion application, feature-based recognition of aircraft behaviors in an air patrol context. Both the problem characteristics and the modeling approach used there are quite different from the one considered here.

Another remotely related problem discussed in the literature is gene splice site prediction, see [8,9]. In a sequence of genomic data, a gene occurs as a band of alternating non-coding *introns* and coding *exons*. The issue is to decide which subsequences are exons and which are introns. The start of an intron is characterized by a *donor* subsequence and the end by an *acceptor* subsequence.

In [10], a Bayesian network is used to estimate force type from vehicle observations, given template models which describe the vehicle composition of force types and Bayesian models for the probability of false vehicle classification. Information about formations or multilevel organizational hierarchies are not used.

In [11, 12] some of the concepts from [10] are extended to an hierarchical target evaluation problem.

The books [13, 14] provide background information on Hidden Markov Models and the Baum-Welch (EM) algorithm used in this paper.

2. Problem statement and representation

The class of recognition problems we study is the following:
- Entities we want to recognize are called *companies.* We observe sets or sequences of objects (*vehicles*) and want to decide which *company type* they might belong to, if any. To

enable us to do this, we assume that we know *organization templates* for a number of different company types.

- A company organization template consists of a known *bag* of *platoon types*, *i.e.*, only the number of *platoon instances* of each type but not the order between them is assumed to be relevant for classification.
- Each platoon template consists of a known bag of *vehicle types.*
- We observe a set of vehicles (either one at a time or all at once) and are able to classify each type of vehicle with known probability of correct classification.
- Using the organization template, we are then able to infer first which bag of platoon types is most likely to be at hand, then which company type this bag most likely indicates.

In principle, this technique can be applied hierarchically level by level, although both as a function of the depth of the organization tree and of the maximum length of organizational subunits, computational complexity for training and recognition both increase exponentially. Therefore, we have only applied the approach to three-level hierarchies with small maximum branching factor (“fan-out”). Fortunately, this seems to be the most interesting case in applications. Note that the assumption that organization templates are known does not mean that they have to be perfectly obeyed. In fact, as a soft modeling technique, HMMs are fairly robust to noisy input, *cf.* the results shown in Sec. 4.

3. Experiments in learning organizational structure

Organizational structure is learned by applying the standard Baum-Welch algorithm [13] to a set of linear HMMs on each organizational level, figures 1-3. The training is done for each linear HMM separately, *e.g.*, each of the three parallel sequences in figure 1. In this manner, the number and types of main states are learned from data, as are necessary transition probabilities. Then the HMMs are connected, so as to form a complete hierarchical structure like that in figure 2.

In the experiments, the organizational structure was defined as follows: A single company type was used, consisting of three platoons, two of type A and one of type B. Platoons of type A contain two vehicles of type a and one of type b, while platoons of type B contain one vehicle of type b and two of type c. This structure was learned in three different ways, as two hierarchical HMMs with different noise levels, and for comparison, also as a “bank” of flat HMMs, in each case using 120 training instances. Each training instance is a unique vehicle sequence, simulating a noisy company observation. The bank of 81 flat HMMs uses a standard HMM method in the manner described in [1], repeated for each of the 81 different vehicle sequences that satisfy the organizational structure. Training times for each of the two hierarchical models was 4 minutes, whereas training the flat HMM bank required 10 hours of CPU time on a 1.5 GHz PC.

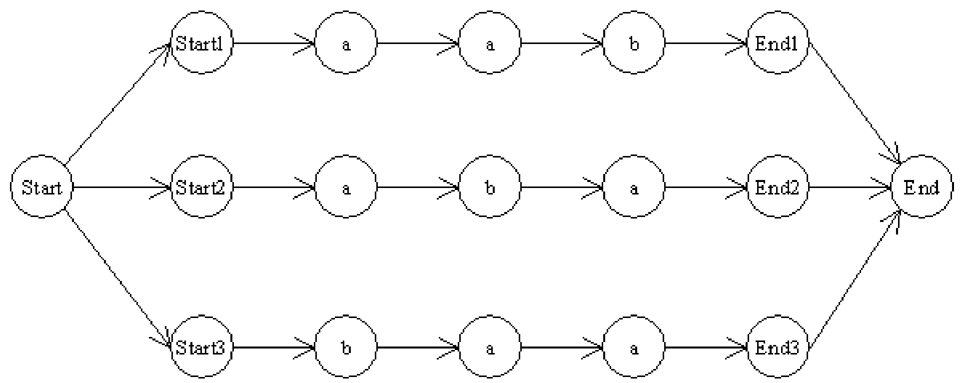

Figure 1. HMM subgraph for one platoon type, corresponding to the *a priori* case “two vehicles of type a, one vehicle of type b”. Each parallel sequence has probability 1/3. Insert and delete states are removed from the figure to improve readability.

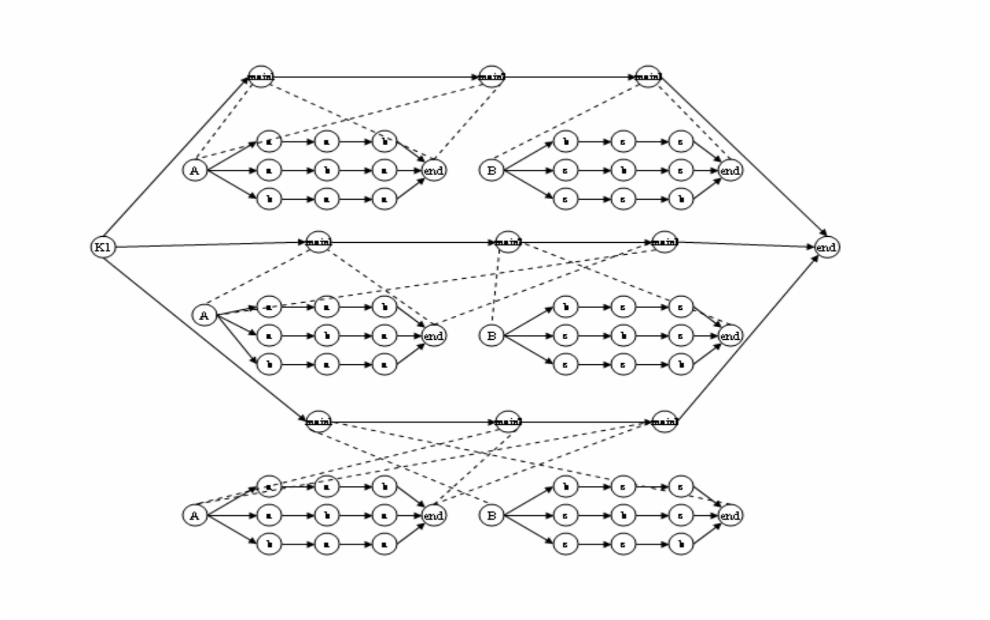

Figure 2. Hierarchical HMM structure for a complete company organization including platoons (A,B) and vehicles (a,b,c). Insert and delete states are removed from the figure to improve readability.

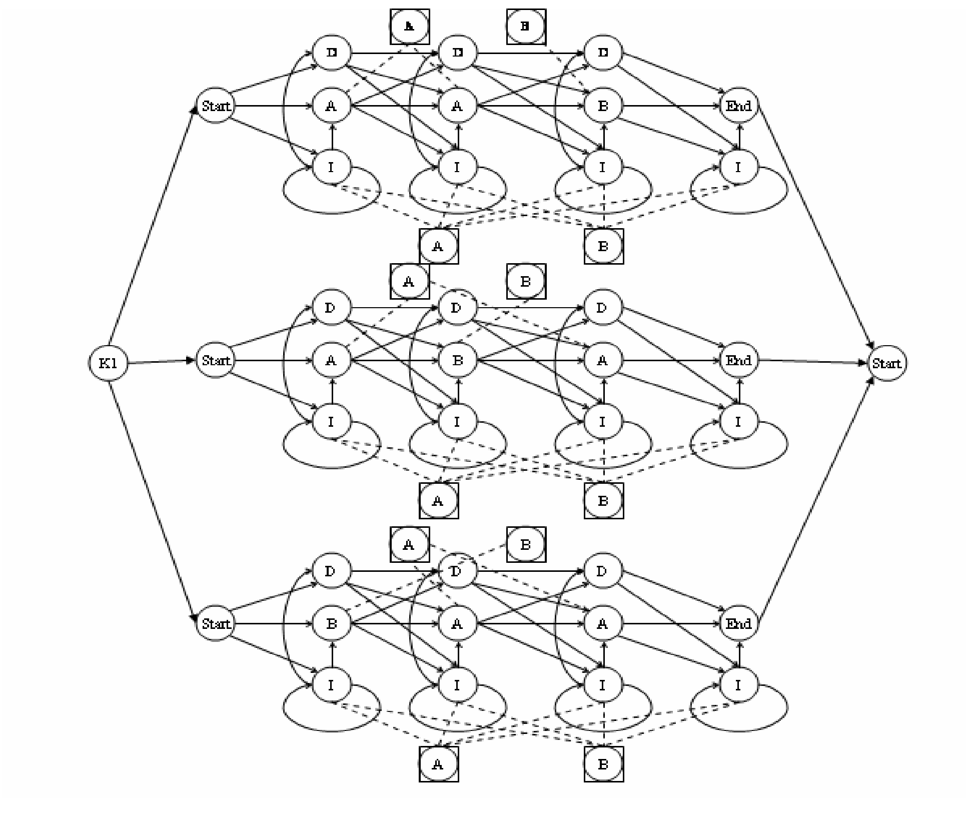

Figure 3. Hierarchical HMM recognizer for a specific company structure, composed of three platoon subsequences connected in parallel. Main states are labeled with their platoon type (A or B), delete states with D, and insert states with I. Only the upper hierarchical level is shown. The lower level submodels are symbolized by squares, which are linked by dashed lines to the upper level.

Case 1. `Hierarchical 1`

The six different HMMs on the lowest level were trained to recognize respectively the A-sequences aab, aba, baa and the B-sequences bcc, cbc, and ccb. Analogously, three HMMs on the higher level recognize the sequences AAB, ABA, and BAA respectively. For each organization type, *i.e.*, company, platoon A, and platoon B, the HMMs for the three alternative component sequences were connected in parallel, each with probability of execution 1/3.

The training sequences for platoons were generated such that the probability of an insertion or a deletion were both 0.08, and the probability of making a vehicle type exchange was 0.08. Together, this means that the probability for generating a normal vehicle at a given position was 0.7728 and the probability of generating a normal platoon was 0.4615. Training sequences for the company had probability 0.01 for both deletions and insertions and probability 0 of type exchange. Thus, the probability of generating a platoon sequence conforming exactly to the company template (without any platoon insertions or deletions) was 0.94. The probability of generating an entire vehicle (and platoon) sequence conforming exactly to the company template was $0.94\ (0.4615)^3 = 0.0925$.

Case 2. `Hierarchical 2`

Here, the probabilities for insertion, deletion, and type exchange of any platoon contained in the company template were all set to 0.08, different from the previous case. All other properties were the same. The net effect of these changes was that the probability of generating a platoon sequence conforming exactly to the company template was reduced to 0.4615 and that of generating a vehicle sequence conforming exactly to the company template was reduced to 0.045.

Case 3. `Flat`

Each of the 81 linear HMMs which correspond to a vehicle sequence conforming to the company template was trained with 120 sequences, conforming to the sequence except for insertion, deletion and exchange noise. The probability of an insertion, deletion, and exchange in a given vehicle position were all 0.08. Thus, the probability of a given vehicle conforming to the template was 0.7728, the probability of all vehicles in a given platoon conforming was 0.4615 and the probability of an entire vehicle sequence conforming to the company template was 0.0983.

4. Recognition experiments and results

In the experiment descriptions below, the following conventions were used:

Five different experiments were performed. In all cases, the set of training cases `Flat`, `Hierarchical 1`, `Hierarchical 2` introduced in Sec. 3 are used as recognizers against which a small number (2-4) of company instances (*test instances*) were matched. Except in the `Perfect` case, the test instances were all similar but not equal to the standard company template. `Insert 1` and `2` correspond to one and two added vehicles respectively, `Delete 1` to one missing vehicle, and `Platoon inserted` to one inserted platoon.

The results for each recognizer, averaged over the number of test instances in each training case are shown in figure 4. To allow for easy comparison across training cases belonging to the same experiment, the sum of the three recognizer fitness values in each experiment was normalized to 1. The absolute fitness corresponding to the normalized unit value is given in figure 4 below each experiment identifier. While the `Hierarchical 2` recognizer scores lowest in all the four experiments with low to moderate noise, it is the most permissive one when confronted with the insertion of an entire platoon. The `Flat` recognizer, on the other hand, rejects this case entirely.

5. Discussion

From the experiments, we conclude that the hierarchical approach needs much less memory space, is vastly faster to learn and more robust to changes on the upper level of the two-level hierarchy than is the use of a complete bank of

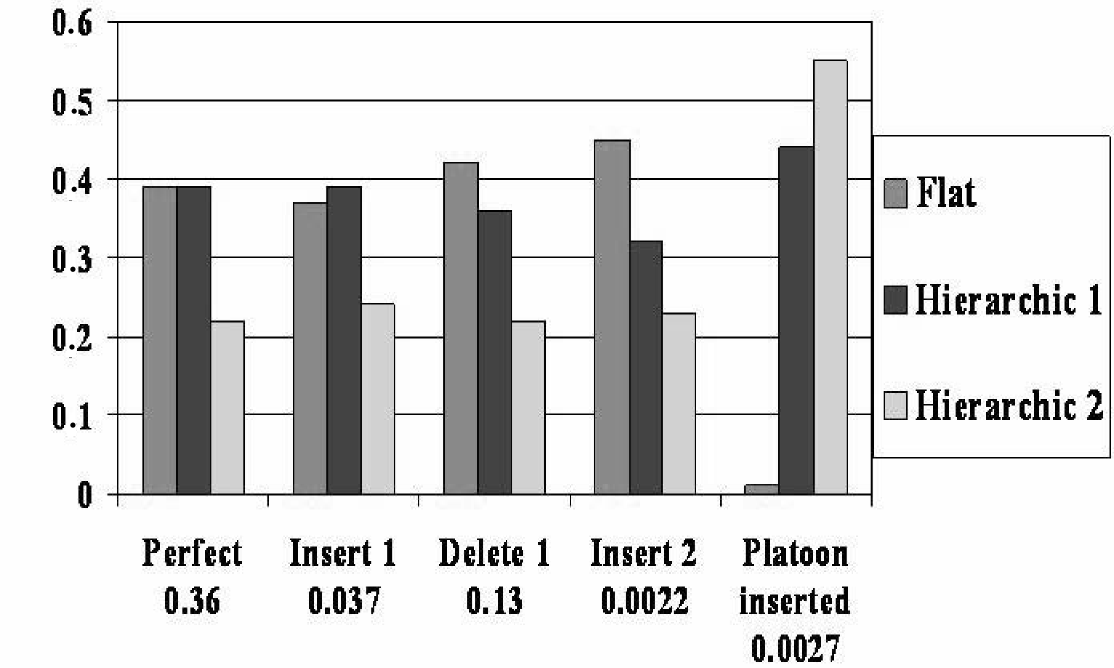

Figure 4. Results from the recognition experiments. Each group of bars corresponds to the relative results of one experiment category, see text. The number below each group identifier is the sum of the fitness values across categories for each experiment.

linear HMMs. Recognition using HHMMs is also much faster.

The HHMM approach described above may be used to create and train recognizers for any combination of ordered and unordered organizational components, *e.g.*, vehicles unordered within platoons while platoons are ordered within companies, or *vice versa*. Partial ordering is also possible, such as prescribing that a certain platoon type should always go first, while no order is precribed for the remaining platoons. Also, a doctrinal rule can be modelled which states that platoons are allowed to move on parallel roads while vehicles within a platoon must always follow each other.

An objection that could obviously be raised against the proposed method is its exponential computational complexity with respect to the cardinality of the organizational components (number of vehicles in a platoon, number of platoons in a company etc.), as well as with respect to the number of organizational levels. However, using present-day PCs, the method should be applicable at least up to a cardinality of 10 in each of two organizational sublevels. We believe this performance to be sufficient for the method to be useful in practice.

References

[1] Björnfot, J. and P. Svensson (2000) Modeling the Column Recognition Problem in Tactical Information Fusion, in *Proc. FUSION 2000*, pp. MoD1/24-30. International Society of Information Fusion, Sunnyvale, CA.

[2] Fine, S., Y. Singer, and N. Tishby (1998) The hierarchical Hidden Markov Model: Analysis and applications. *Machine Learning*, 32:41.

[3] Murphy, K. and M. Paskin (2001) Linear time inference in hierarchical HMMs, in *Proc. NIPS 2001*.

[4] Bui, H.H., D.Q. Phung, and S. Venkatesh (2004) Hierarchical Hidden Markov Models with general state hierarchy, in *Proc. AAAI 2004*.

[5] Murphy, K. (2002) *Dynamic Bayesian networks: Representation, inference and learning*. Ph.D. thesis, Univ. of California, Berkeley.

[6] Wierstra, D. (2004) A new implementation of Hierarchical Hidden Markov Models, Ch.

5 in *Statistical models for non-Markovian control tasks*, M.Sc. thesis INF/SCR-04-06, Institute of Information and Computing Sciences, Utrecht University.

[7] Chiao, S.-Y. and C.S. Xydeas (2004) Using Hierarchical HMMs in dynamic behavior modeling, in *Proc. FUSION 2004*, pp. 576-582. International Society of Information Fusion, Sunnyvale, CA.

[8] Kulp, D., D. Haussler, M.G. Reese, and F.H. Eeckman (1996) A generalized Hidden Markov Model for the recognition of human genes in DNA, in *Proc. ISMB 4*, pp. 134-142.

[9] Hu, M., C. Ingram, M. Sirski, C. Pal, S. Swamy, and C. Patten (2000) A Hierarchical HMM implementation for vertebrate gene splice site prediction, http://genome.dfci.harvard.edu/~mhu/reports/hmmSplice.pdf

[10] Johnson, J., and R. Chaney (1999) Recursive Composition Inference for Force Aggregation, in *Proc. FUSION 1999*, pp. 1187-1195. International Society of Information Fusion, Sunnyvale, CA.

[11] J. P. Hill and K. C. Chang (2003) Improved Representations of Sensor Exploitation for Automatic Sensor Management, in *Proc. FUSION 2003*, pp. 688-694. International Society of Information Fusion, Sunnyvale, CA.

[12] J. P. Hill and K. C. Chang (2004) Sensor Resource Management with Hierarchical Target Valuation Methods, in *Proc. FUSION 2004*, pp. 1187-1194. International Society of Information Fusion, Sunnyvale, CA.

[13] P. Baldi and S. Brunak (1999) *Bioinformatics: The machine learning approach.* The MIT Press, Cambridge, MA.

[14] G. J. McLachlan and T. Krishnan (1997) *The EM Algorithm and Extensions*. J. Wiley & Sons, New York, NY.