A model-based approach to sequence clustering

Henri Binsztok and **Thierry Artières** and **Patrick Gallinari**[1]

Abstract. We present a Hidden Markov Model (HMM) based approach to cluster sequences. This problem is addressed in term of learning Hidden Markov Model structure from data. Using a top-down approach, we iteratively simplify an initial HMM that is built to cover all training sequences. Our approach allows to learn, in an unsupervised manner, the cluster models that best represent training data. We provide experimental results on two different application fields, on-line handwriting signals and hypermedia navigation patterns.

1 Introduction

There are two categories of sequence clustering algorithms: the first category is based upon a similarity measure between sequences; the second is model-based and infers a model that describes each cluster. Our approach belongs to the second category, which seeks to build a descriptive model for the data and goes beyond the clustering result. Among this category, one common model class is the Hidden Markov Model (HMM, [9]). HMMs were introduced for speech recognition to deal with sequential information and have been used for many sequence modeling and classification tasks (handwriting, biology, etc.). There are many works for clustering sequences with HMMs. [14] provides a generic algorithm to cluster sequences into a fixed number of clusters, along with a method to find the numbers of clusters. This number is determined by cross-validation using a Monte Carlo measure, which is controlled by an hyper-parameter. This theoretical approach relies on iterative reestimation of parameters via an instance of EM, which requires careful initializations. Furthermore, the structure of the model is limited to a mixture model of fixed-length left-right HMMs, which may not correctly model sequences of varying lengths in the data.

We tackle the sequence clustering problem through the unsupervised learning of a HMM model. The learning is constrained in such a way that the trained HMM may be used, at the end, to cluster sequences. Generally, the training of HMM models is carried out in two stages: First the prior choice of a model structure (e.g. number of states, allowed transitions), then a statistical training of the parameters from data. The learning of the structure is thus performed manually through successive trials. Some work has been done in order to learn automatically the structure from the data, together with the parameters. A generic approach to learn HMM structure may be found for example in [16]. Our main contribution is to put a constraint on the learned structure so that the HMM belongs to the class of mixtures of varied lengths left-right HMMs (i.e. HMM with no cycle in transitions): comparing with [14] nor the length of left-right HMMs nor the number of components in the mixture are fixed. The main idea of such a constraint is to learn a global HMM that covers training data and whose topology consists of a few left-right HMMs, each one being seen as a cluster model.

We first describe an overview of our approach (section 2). It starts by building (topology and parameters) of an initial HMM from the data. This HMM is a mixture of as much left-right HMMs (we call these components in the following) as there are training sequences. This HMM is then iteratively simplified by removing components. We present the structure learning algorithm in section 3. In section 4, we describe experimental studies on two kinds of data: (1) on-line handwritten digit clustering and (2) determining user typologies in hypermedia navigation patterns.

2 Overview of the approach

We present our approach to clustering in the scope of model structure learning. Some studies in structure learning are specific to a given task: We focus here on generic methods. Approaches suggested in the literature use mainly a top-down approach, and start by building a complex initial model and simplifying it iteratively ([16, 7, 3]). In [3], the simplification is based on entropic prior probabilities of transition between states. One of the properties of this approach is that some transition probabilities converge towards 0, thus simplifying the structure of the model. In [16], states from the initial HMM are merged iteratively as long as the loss of likelihood is not too significant. The model structure learning problem for sequential data is not limited to HMMs. Prediction Suffix Trees (PST, [11]) are simpler models, quite close to HMMs. They are probabilistic automata in which transitions are deterministic. Structure learning for PSTs includes two main approaches: [11] uses a bottom-up approach while subsequent work in [13] starts by building a huge model, which is iteratively simplified and can lead to better performance. For more complex models (e.g. Bayesian Networks or Probabilistic Context-Free Grammars), structure learning is a even more difficult problem that does not provide good generic performance.

We chose to keep the top-down approach and propose to adapt the approach of [16] by restricting the HMM to belong to a particular class $\mathcal{MLR}$ of models that are mixture of left-right HMMs. Let M be a HMM that is a mixture of K left-right HMMs, the probability to observe a sequence O is $P(O|M) = \sum_{k=1}^{K} P(\lambda_k)P(O|\lambda_k)$ where $(\lambda_k)_{1,..,K}$ are K left-

[1] LIP6, Université Paris 6, 8 rue du Capitaine Scott, 75015 Paris, France. email: Henri.Binsztok@lip6.fr, Thierry.Artieres@lip6.fr, Patrick.Gallinari@lip6.fr

right HMMs. First, a global model M in $\mathcal{MLR}$ is built from the data, using one left-right HMM per training sequence. Then, this global HMM is iteratively simplified by merging or removing components. Doing so, the global HMM always belong to $\mathcal{MLR}$. In particular, the merging or removing is not done state by state, but directly between left-right HMMs. The main interest of this approach is that it naturally leads to a number of left-right HMM models that correspond to different clusters.

3 Learning algorithm

The learning algorithm proceeds in two steps: First, we present our procedure to construct an initial HMM (section 3.1). Then, we describe the iterative simplification algorithm applied to this initial model (section 3.2). The simplification algorithm relies on a distance that is detailed in sections 3.3.

3.1 Building an initial HMM from data

This first stage consists of building a HMM $M_0 \in \mathcal{MLR}$ summarizing all training sequences. Let $D = (s_1 \dots s_n)$ be this sequence set. Each sequence s_i of length l_i is a sequence of symbols $s_i = (\sigma_{i,1} \dots \sigma_{i,l_i})$ where each symbol $\sigma_{i,j}$ belongs to an alphabet Σ and let $|\Sigma| = \text{card } \Sigma$. We start by building n left-right HMMs, each one derived from a training sequence. For sequence s_i, we build a left-right HMM with l_i states. The initial HMM M_0 is then a mixture of these n left-right HMMs.

We then have to estimate one emission p.d.f. for each state of each left-right HMM. The approach suggested by [14] that consists of learning these p.d.f. with a standard algorithm for HMMs did not appear relevant to us. Indeed, this training is delicate insofar as it requires finding a good initialization. We rather chose to share the emission p.d.f. between states corresponding to the same symbol of Σ so that there is only $|\Sigma|$ p.d.f. to estimate, one for each symbol. For instance, if the first and last symbols of a sequence are the same symbol σ, the two corresponding states in the left-right HMM built from this sequence will share the same emission p.d.f.

For a given application, we could use prior knowledge to define these emission p.d.f. The strategy we chose does not require prior knowledge and consists of directly estimating emission p.d.f. by counting. We consider as similar two symbols which appear in the same context: Let $s \in \Sigma^*$ be a string over alphabet Σ and let $P_s(\sigma)$ be the probability of observing symbol σ after prefix s. An estimate for $P_s(\sigma)$ is $P_s(\sigma) = \frac{w(s\sigma)}{w(s)}$ where the count $w(s)$ represents the number of occurrence of the subsequence s in D. One may define a profile for any symbol σ as the set $\{P_s(\sigma)/s \in \text{sub}(D)\}$ for all subsequences s in training data. Intuitively, two symbols with similar profiles should be very close. Hence, we derive the similarity κ between two symbols $(\sigma_1, \sigma_2) \in \Sigma^2$ by the correlation between the distributions P_{σ_1} and P_{σ_2}: $\kappa(\sigma_1, \sigma_2) = \text{corr}(P_{\sigma_1}, P_{\sigma_2})$. Finally, the emission p.d.f. b_σ built from symbol σ is given by $b_\sigma = \left\{ \frac{\kappa(\sigma,\sigma')}{\sum_{\sigma'' \in \Sigma} \kappa(\sigma,\sigma'')}, \sigma' \in \Sigma \right\}$. We will see in section 5 that this estimate provides good results.

3.2 Simplification algorithm

The general outline of the algorithm is to iteratively merge the two *closest* components in the model using a distance δ between left-right HMMs. The distance is presented in next subsection. The algorithm is as follows:

1. Build the initial HMM model $M = M_0$ as detailed in previous section.
2. Repeat until the stopping criterion is satisfied:
   (a) Select the couple of components (u, v) closest with respect to δ.
   (b) Merge the two components to obtain new model M

In this preliminary work, merging is restricted to selection, i.e. we preserve among the two new HMM candidates $M \backslash \{u\}$ and $M \setminus \{v\}$ the model that optimizes the criterion C.

Several stopping criteria are possible. An interesting approach is to penalize the likelihood with the size of the model, and thus to favor a compact model with Minimum Description Length (MDL, [10]). In our model, the criterion expression is: $C = \log P(D|M) - \alpha[\log N|\Sigma|]$ where $P(D|M)$ is the likelihood of the data by the HMM, N the total number of states of all components of the HMM and α is a parameter allowing tuning the importance of the two terms of the criterion: the modeling accuracy and the model complexity. The stopping criterion is satisfied when the criterion C ceases growing.

3.3 Left-right HMM distance measure

Let M_1 and M_2 be two left-right HMMs, of respective lengths l_1 and l_2. We use an easily computable distance $\delta(M_1, M_2)$ between two discrete HMM that takes account of their left-right topology: It is based on an alignment between the states of M_1 and M_2 using a Dynamic Time Warping algorithm where local costs are distances between states (i.e. their emission p.d.f.) [12, 2]. We use the symmetrized Kullback-Leibler divergence d_{KL} as local cost. Hence the algorithm seeks to find an optimal alignment between the states of M_1 and the states of M_2, that minimizes the cost: $J = \sum_{k=1}^{p} d_{KL}(i_k, j_k)$ where $d_{KL}(i_k, j_k)$ is the symmetrized Kullback-Leibler distance between the p.d.f. of the state i_k of M_1 and the state j_k of M_2 and where the sequence of the indices $\{(i_k, j_k), k \in [1, p]\}$ corresponds to an authorized alignment (using classical DTW path constraints) of pair of states of the two models. As we are interested in left-right HMMs, we impose the limiting conditions $(i_1, j_1) = (1, 1)$ and $(i_p, j_p) = (l_1, l_2)$ so that δ is defined as:

$$\delta(M_1, M_2) = \hat{J} = \min_{\substack{p, \{(i_k, j_k), k \in [1,p]\} \\ (i_1, j_1) = (1,1); (i_p, j_p) = (l_1, l_2)}} \sum_{k=1}^{p} d_{KL}(i_k, j_k)$$

4 Experimental study

We show here the results of our approach with two kinds of data. We first describe our evaluation method and then our results.

4.1 Evaluation method

Evaluating unsupervised methods is an open problem. As stated in [15], we use labeled data for the sole purpose of evaluation. Evaluating the clustering as a classification task allows us to use standard evaluation methods from supervised learning. We use two complementary measures to evaluate the relevance of a clustering: the entropy measure and the F measure which is extensively used in the information retrieval community.

The result of the clustering algorithm is a set of sequences grouped in clusters. We name *clusters* the result of clustering and *classes* the labeling of the data. The first measurement, named *total entropy*, is related to the homogeneity of the clusters compared to the class information. If all the sequences in a cluster correspond to the same class, the cluster is perfectly homogeneous, and its entropy is null and minimal. More formally, for a cluster j, the entropy is defined by $E_j = -\sum_i p_{ij} \log p_{ij}$, where p_{ij} is the probability that an element of the cluster j belongs to class i. The total entropy is computed with: $E_T = \sum_j \frac{n_j E_j}{n}$, where n_j is the number of elements of the cluster j and n the total number of sequences in the data. The second measurement is F measure. If all the clusters are homogeneous and non-redundant (there does not exist two clusters corresponding to the same class), F measure is worth 1 and is maximal. F measure incorporates two measurements: precision and recall. The precision captures information similar to the entropy, while recall is high when classes are not much scattered. By definition, $P(i,j) = \frac{n_{ij}}{n_j}$ and $R(i,j) = \frac{n_{ij}}{n_i}$. Then, $F(i,j) = \frac{2P(i,j)R(i,j)}{P(i,j)+R(i,j)}$. The total F measure derives from the F measure computed for each class and cluster: $F = \sum_i \frac{n_i}{n} \max F(i,j)$.

We compared our approach with a recent approach for sequences clustering proposed by [4], based on EM algorithm and Markov chains. This algorithm strongly depends on initialization: Since a random initialization provides poor result, we initialized the Cadez clustering algorithm with our method. Cadez approach is a suboptimal approach to maximize likelihood, which is one of the only criteria available for unsupervised learning. Comparing our approach with reestimations - that enhance likelihood - allows us to obtain an insight of the clustering performance of our approach.

4.2 On-line Handwriting

4.2.1 The task

On-line handwritten signals are temporal sequence of pen coordinates captured through an electronic pen or a digital tablet. Recognizing handwriting signals is a key problem for the development of mobile terminals using pen-based interface such as personal digital assistants, electronic note taking devices, electronic books. Building on-line handwriting recognition systems is not an easy task since there exists an important variability due to different writing styles (e.g. allographs). Ideally, in a recognition system, there should be as many models for a character as there are writing styles for this character. A few studies have focused on the automatic identification of writing styles to determine from the data the number of models for a character, sometimes to learn partially the topology of HMM for a character, or to recognise a writer based on his writing style. Various techniques have been used. [6] chooses a probabilistic approach to define clusters: For each character, an approach similar to EM is used to learn the probabilities that a character belongs to a given cluster. The association of clustering and HMM was approached by [8], but it highly depends on initialization which is supervised.

4.2.2 Database

We carry out our experiments on a database of digit signals, extracted from the Unipen database [5]. The signals are labeled with the corresponding digits but there is not any label information about allographs. Then, although our approach can be used to learn the topology of a Markovian character model and/or to identify its allographs, we chose to perform our clustering experiments on databases including signals of various and close digits (e.g. 0 and 9). This enables us to easily evaluate the relevance of the discovered clusters, using the total entropy and F measure previously introduced. The rough on-line signal is a temporal sequence of pen coordinates and is first preprocessed. We chose to preprocess the signals as in [1]. A handwritten signal is represented as a sequence of *strokes* where each stroke is characterized by a direction and a curvature. The strokes belong to a finite dictionary Σ of 36 elementary strokes, including 12 straight lines in uniformly distributed directions between 0 and 360°, 12 convex curves and 12 concave curves. Such a sequence of strokes represents the shape of the signal and may be efficiently used for recognition. All handwriting signals are converted into sequences of elementary strokes. We provide in the following, results related to the estimation of emission p.d.f. in the initial HMM, as described in section 2.2 and to the clustering of handwritten digits.

4.2.3 Emission probability distribution functions

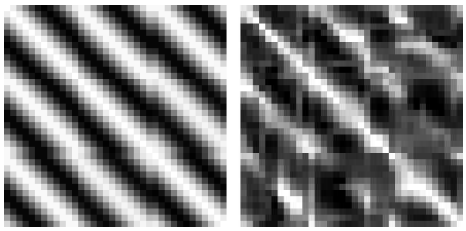


Figure 1. 36x36 matrixes representing p.d.f. for states associated to each symbol of Σ, fixed manually using prior knowledge (left) and learned from the data (right).

Figure 1 represents the estimation of emission p.d.f. for states associated to each symbol of the alphabet Σ. The dimension of both matrixes is 36x36: The square at the intersection of the i^{th} row and j^{th} column is the probability of observing symbol σ_j emitted by state associated to σ_i. Gray levels are proportional to probabilities (white = close to 1, black = close to 0). The left matrix is fixed manually according to prior knowledge [1] while the right matrix is estimated with the method presented in section 2.2. We note a strong

correlation between these matrixes, which shows that our estimation method allows capturing efficiently the information of similarity between symbols contained in the sequences D.

4.2.4 Clustering experiments

In a first experiment, (EXP1), we use 100 samples of digits '0' and '9' whose drawings are very similar. In the next two series of experiments (EXP2 and EXP3) we use 150 samples of digits '0', '1' and '2', the only difference is the value of the MDL parameter α (1.5 for EXP2, and 1 for EXP3). For each of these three series of experiments, we carried out several tests and averaged the results. The results of these experiments are provided in table 1. One notes a significant improvement of our approach compared to that of [4], regarding both homogeneity of clusters (low entropy) and limited number of discovered clusters (F measure closer to 1). By comparing the results of the two last series of experiments, differing by the value of α, one sees that the results are rather similar for F measure, but the entropy is higher for experiment EXP2, using a stronger value. That is natural since the penalization term on system complexity is stronger for EXP2, resulting in a lower number of clusters. Though we lack a generic method to adjust values of α, we observed experimentally that good results are obtained with similar values.

	Entropy: BAG	Entropy: [4]	F: BAG	F: [4]
EXP1	0.00	0.24	0.85	0.70
EXP2	0.32	0.20	0.63	0.58
EXP3	0.13	0.18	0.62	0.52

Table 1. Performance (total entropy and F measure) of our approach (BAG) as compared to Cadez [4] for on-line handwriting digits clustering.

To further study the effect of the MDL parameter α, we measured the number of clusters and the performance as a function of α. These results are presented in the figure 2 and were carried out on the same database as experiment EXP1. We note good performances, given the unsupervised nature of our task, in particular for $\alpha = 2.5$. The entropy is null and F-measure is higher than 80%. The resulting clusters from this experiment are presented in figure 3. One sees on this figure that the discovered clusters are homogeneous (including either '0' or '9' samples). The two clusters for digit '0' include indeed slightly different drawings since samples from the smaller set are drawn the other way round.

4.3 User modeling

4.3.1 The task

The task we tackle here is to automatically learn user behavior models from hypermedia navigation patterns. Discovering user typologies is an important task in the context of hypermedia. Knowing the typology of a user allows personalizing the media for him. This is particularly interesting in hypermedia (web sites) since these may be very complex and are accessed by various users. We are interested here in discovering typologies in navigation patterns. The goal is to guess,

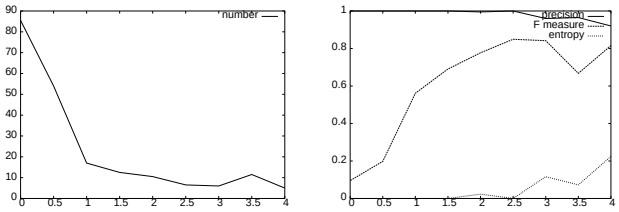


Figure 2. Number of clusters (left), total entropy, F measure and precision (right) as a function of the MDL parameter α for digit samples clustering.

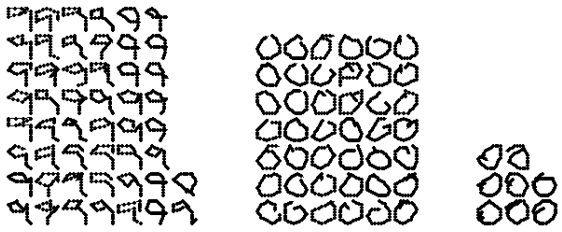


Figure 3. Discovered clusters for a database of on-line handwriting samples of digits '0' and '9'.

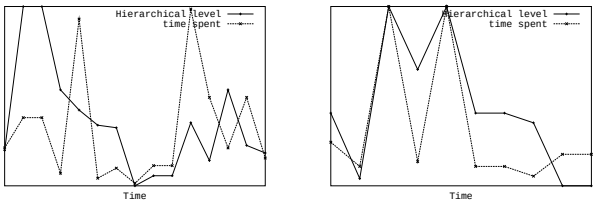
from low-level log information (i.e. temporal sequence of actions such as clicks, scrolls, etc), what the user is doing and to use this information to help and guide him.

4.3.2 Database

We performed experiments on a prototype of the future website for the National Museum of Natural History. This complex hypermedia includes thousands of pages organised in a dense lattice. Log information is collected during a user session and transformed into a temporal series of feature vectors: There is one feature vector for each page accessed during the session. Feature vectors include characteristics related to the nature of the page (i.e. leaf in the hierarchy of pages of the hypermedia, table of contents, etc.), the user activity (number of clicks, scrolls, etc.), the path followed by the user (proportion of new pages accessed vs. already seen pages in last 2 minutes). The feature vectors of a session are quantized using a codebook of 36 vectors (learned on the whole user sessions) so that a user session is represented at the end as a sequence of symbols, each one is a codebook number. In order to evaluate our clustering using criteria defined in section 4.1, as we did for on-line handwriting signals, we collected log information of user sessions in a controlled way, by asking users to answer specific questions by navigating the hypermedia. These questions were formulated so as to induce a predefined behavior such as searching, browsing etc. For example, to induce a browsing behavior, users were asked to tell what the nicest picture was, in a particular subpart of the web-site. We collected this way a database of about 60 user sessions, labeled with four predefined behaviors: exploring, browsing, searching and scanning.

4.3.3 Clustering experiments

It must be noticed that supervised results on this task are relatively poor: Data is noisy and classes are not well defined (e.g. searching and browsing). Recognition rate is below 80% for all four classes, hence unsupervised methods do not exhibit relevant performance according to criteria defined in section 4.1. However, a first experiment was conducted to cluster user sessions corresponding to two behaviors (exploring and browsing) that are not too similar, and performance reaches 83% precision with 4 clusters (64% with 2 clusters). In a second experiment, we investigated deeper the discovered cluster models learned from the whole database including sessions corresponding to the four behaviors. Although performance is worse (about 55% precision with 4 clusters), the information captured by the cluster models is interesting. Our current method which relies on sequence model selection (i.e. merging is restricted to selection) allows to exhibit a typical session that is representative of a cluster. As an example, we plot in Figure 4 typical sessions corresponding to 2 clusters. We study two characteristics of the feature vectors: The first is related to the level of the accessed page in the hierarchy of the web-site (the core information is in the leaves) and the second to the time spent on the accessed pages. One can see that the first cluster corresponds to sessions where the user navigates quickly on table of contents (with a long stay on only one of these pages) and then goes down in the hierarchy (leaves) spending more time reading pages. This behavior is typical of searching for a given data. In the typical session of the second cluster the user first spends much time on table of contents pages then navigate quickly on leaves, which characterizes a browsing behavior. This shows that our approach can be used to identify clusters and help defining more precisely the classes in the data.



Cluster 1 Cluster 2

Figure 4. Typical navigation sessions of two clusters. For each cluster, we display the evolution of two characteristics: The hierarchical level of the page accessed along the session and the time spent on the page.

5 Conclusion and future work

We presented a model-based approach to cluster sequences that we tackled through unsupervised HMM structure learning. We propose to learn, from the data, the structure of a global HMM within a subclass of HMMs that seems more appropriate for sequence clustering. The subclass of HMMs considered is the set of mixture of left-right HMMs (named components). The learning is a top-down approach where we build an initial model from data and then simplify it iteratively by merging components, until a MDL-like criterion is reached. The merging is here restricted to selection which allows for typical sequences identification. We provided experimental results for two different data: on-line handwritten digits and hypermedia navigation patterns. These preliminary results are promising and we look forward to validate them on larger datasets.

REFERENCES

[1] Thierry Artières and Patrick Gallinari, 'Stroke level HMMs for on-line handwriting recognition', in *8th International Workshop on Frontiers in Handwriting Recognition (IWFHR-8)*, Niagara, (August 2002).

[2] Claus Bahlmann and Hans Burkhardt, 'Measuring HMM similarity with the bayes probability of error and its application to online handwriting recognition', in *ICDAR*, pp. 406–411, (2001).

[3] M. Brand, 'Structure learning in conditional probability models via an entropic prior and parameter extinction', *Neural Computation*, **11**, 1155–1182, (1999).

[4] Igor V. Cadez, Scott Gaffney, and Padhraic Smyth, 'A general probabilistic framework for clustering individuals and objects.', in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, pp. 140–149, N. Y., (August 20–23 2000).

[5] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, 'Unipen project of on-line data exchange and benchmarks', in *International Conference on Pattern Recognition, ICPR'94*, pp. 29–33, Jerusalem, Israel, (1994). IEEE.

[6] Ali Nosary, Laurent Heutte, and Thierry Paquet, 'Unsupervised writer adaption applied to handwritten text recognition', *Pattern Recognition*, **37**, 385–388, (2003).

[7] Stephen M. Omohundro, 'Best-first model merging for dynamic learning and recognition', in *Advances in Neural Information Processing Systems*, eds., John E. Moody, Steve J. Hanson, and Richard P. Lippmann, volume 4, pp. 958–965. Morgan Kaufmann Publishers, Inc., (1992).

[8] M. Perrone and S. Connell, 'K-means clustering for hidden markov models', in *In Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pp. 229–238, Amsterdam, Netherlands, (September 2000).

[9] L. R. Rabiner, 'A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition', *Proceedings of the IEEE*, **77**(2), 257–285, (February 1989).

[10] J. Rissanen, 'A universal prior for integers and estimation by Minimum Description Length', *Annals of Statistics*, **11**, 416–431, (1982).

[11] Dana Ron, Yoram Singer, and Naftali Tishby, 'The power of amnesia', in *Advances in Neural Information Processing Systems*, eds., Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, volume 6, pp. 176–183. Morgan Kaufmann Publishers, Inc., (1994).

[12] D. Sankoff and J. B. Krushkal, *Time warps, string edits, and macromolecules: The theory and practice of sequence comparison*, Addison-Wesley, 1983.

[13] Yevgeny Seldin, Gill Bejerano, and Naftali Tishby, 'Unsupervised sequence segentation by a mixture of switching variable memory Markov sources', in *Proc. 18th International Conf. on Machine Learning*, pp. 513–520. Morgan Kaufmann, San Francisco, CA, (2001).

[14] Padhraic Smyth, 'Clustering sequences with hidden markov models', in *Advances in Neural Information Processing Systems*, eds., Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, volume 9, p. 648. The MIT Press, (1997).

[15] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques, 2000.

[16] Andreas Stolcke and Stephen Omohundro, 'Hidden Markov Model induction by bayesian model merging', in *Advances in Neural Information Processing Systems*, volume 5, pp. 11–18. Morgan Kaufmann, San Mateo, CA, (1993).