

# An Interweaved HMM/DTW Approach to Robust Time Series Clustering

Jiaying Hu      Bonnie Ray

IBM T.J. Watson Research Center, 1101 Kitchawan Road, Route 134 Yorktown Heights, NY 10598  
{jyhu,bonnier}@us.ibm.com

Lanshan Han

Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180  
hanl3@rpi.edu

## Abstract

We introduce an approach for model-based sequence clustering that addresses several drawbacks of existing algorithms. The approach uses a combination of Hidden Markov Models (HMMs) for sequence estimation and Dynamic Time Warping (DTW) for hierarchical clustering, with interlocking steps of model selection, estimation and sequence grouping. We demonstrate experimentally that the algorithm can effectively handle sequences of widely varying lengths, unbalanced cluster sizes, as well as outliers.

## 1 Introduction

Cluster analysis is a way to derive structure from data by automatically partitioning the data samples into homogeneous groups. In model-based clustering, mathematical models are used to represent the cluster structure, with the models for each cluster selected to optimize the data fit. Compared to distance-based clustering, model-based methods can provide better interpretability and richer representation of the data.

Hidden Markov Models (HMMs) are particularly attractive for the clustering of time series, or more generally, sequence data, for two reasons. First, they represent a formal probabilistic model with solid mathematical foundations, with available efficient and well-defined algorithms for inducing HMMs from a set of sequences [4]. Second, the hidden states in HMMs provide a compact and easy-to-interpret representation of the underlying “stages” in a dynamic process.

Earlier work on HMM based sequence clustering assumed that the number of states in the models is known beforehand [5, 1, 8, 3, 6]. More recently, Li *et. al.*[2] proposed a more general clustering methodology called *Ma-*

*tryoshka*, which does not assume that the number of states in the HMMs are known beforehand or are fixed for all clusters. However, there are several drawbacks to this methodology that hinder its application in many real-world situations. First, the method assumes that all sequences are of equal length. Second, to generate new clusters, the method uses a simple approach of initializing a new cluster using the sequence that is farthest from the existing models, which leads to instability when the data contains clusters of very different sizes or outliers. Finally, the method provides no mechanism to isolate outliers or noise in the data: it attempts to account for all the data with HMMs, thus tends to get “distracted” in the presence of outliers.

In this paper we present a new algorithm for HMM-based sequence clustering designed to address these problems. A normalized Bayesian Information Criterion (BIC) measure is adopted to allow for sequences of varying lengths. A mechanism called the *outlier pool* is introduced to dynamically identify and handle outliers throughout the clustering process. Finally, we provide a more reliable methodology for creating and initializing new clusters using Dynamic Time Warping (DTW) combined with hierarchical clustering. While DTW has been used before in sequence clustering [3], our algorithm is the first to interweave it into every step of a top-down, model-based clustering scheme that searches for the optimal number of clusters and number of states for each cluster in an iterative manner, guided by a goodness of fit measure.

## 2 The Interweaved HMM/DTW Algorithm

Suppose we have a set of  $N$  sequences (samples) of varying lengths:  $X = (x_1, \dots, x_N)$ . We assume that a majority of the sequences were generated by an unknown number of HMMs, each representing a “dominant” underlying regime in the data. By “dominant” we mean it represents a significant number of sequences. However, the number of

**Table 1. Outline of the clustering algorithm**

|   |
|---|
| Assign all sequences to one cluster.  |
| Apply <b>Model Construction</b> to the cluster.                                     |
| <b>Sample Reassignment/Outlier Detection.</b>                                       |
| Compute <b>normalized</b> partition BIC measure.                                    |
| <b>while</b> BIC measure for current partition > BIC measure of previous partition: |
| <b>Partition Growing.</b>   |
| Apply <b>Model Construction</b> to each new cluster.                                |
| <b>Sample Reassignment/Outlier Detection.</b>                                       |
| Compute <b>normalized</b> BIC for current partition.                                |
| <b>end while</b>  |
| Accept the previous partition as the final partition.                               |

sequences represented by each dominant regime may vary widely (*i.e.*, the corresponding clusters may be highly unbalanced). We further assume that the data may contain outliers, or sequences that do not belong to any of the dominant regimes. The goal of the clustering algorithm is to identify the clusters that correspond to these dominant regimes, along with the underlying models that characterize the sequences in each regime.

This clustering problem can be viewed as a model-fitting problem, where, given a set of data assumed to come from a mixture of models, we attempt to find the best estimate of the model parameters such that they maximize the likelihood of the data. The challenge is how to solve the nested problems of identifying the “right” number of clusters, and given a cluster, the “right” model size for the cluster. We adopt a top-down approach, where we start with the minimal size for both model and partition, and increment them in an estimation-maximization (EM)-like procedure until a certain “goodness” measure is reached.

Table 1 gives a high-level outline of our clustering algorithm. In the following sections we explain in detail the three key components of this algorithm: model and partition size selection, partition growing, and outlier handling.

## 2.1 Normalized BIC for size selection

The Bayesian Information Criterion (BIC) was first proposed as a criterion for model selection when fitting a mixture model in a Bayesian framework. It was derived from an asymptotic approximation formula proposed by Schwarz in 1978 [7]. The basic definition of the BIC measure given a mixture model  $M$  and data set  $X$  is:  $BIC(M, X) = \log\{P(X|M, \hat{\theta})\} - \frac{d}{2} \log(N)$ , where  $\hat{\theta}$  denotes the maximum likelihood estimate of the model parameters,  $d$  is the number of free parameters in the model, and  $N$  is the number of data samples in  $X$ . The first term in the formula is the likelihood term, which tends to favor

larger and more detailed models, while the second term is the model complexity penalty term, which favors simpler models. Thus BIC has the effect of selecting a good, yet parsimonious model for the data by trading off the contributions of these two terms.

A direct adaption of the BIC measure to sequence clustering [2] is problematic when the sequences have widely varying lengths. Due to its cumulative nature, the likelihood of a sequence tends to be lower for longer sequences. Thus BIC measures using likelihoods not normalized to account for sequence length are biased towards longer sequences.

To correct for this bias, we normalize the BIC measure by dividing the first term by the length of the sequence, and adding a regularization factor  $\alpha$  (roughly the reverse of the average sequence length) to the penalty term, resulting in what we call a *normalized* BIC measure.

For model  $\lambda_k$  with parameters  $\hat{\theta}_k$  estimated from cluster  $X_k$ , the normalized model BIC measure is defined as:  $\sum_{j=1}^{N_k} \frac{\log P(x_{kj}|\lambda_k, \hat{\theta}_k)}{|x_{kj}|} - \alpha \times \frac{d_k}{2} \log N_k$ , where  $x_{kj}$  is the  $j$ th sequence in cluster  $X_k$ ,  $|x_{kj}|$  is its length, and  $P(x_{kj}|\lambda_k, \hat{\theta}_k)$  is the likelihood of the sequence.

Similarly, for a given partition  $M$  containing  $K$  clusters, the partition BIC measure is defined as:  $\sum_{i=1}^N \sum_{k=1}^K P_{ik} \frac{\log P(x_i|\lambda_k, \hat{\theta}_k)}{|x_i|} - \alpha \times \sum_{k=1}^K \frac{d_k}{2} \log N$ , where  $P_{ik}$  is 1 if sample  $x_i$  is in cluster  $k$  and 0 otherwise.

The process of state size selection is embedded in the **Model Construction** module referred to in Table 1. The algorithm starts with a single state and increases the number of states by one at each iteration until the BIC measure begins to decrease.

Similarly, to choose the number of clusters, the algorithm starts from one cluster and keeps increasing the number of clusters until the partition BIC measure begins to decrease (outlined in Table 1).

## 2.2 Outlier handling

Outliers in the context of model-based clustering refer to objects that do not belong to any of the dominant underlying clusters. Most likely these objects have been generated due to system anomaly or noise and therefore are not of primary interest.

Outliers are very common in real world data and can cause serious difficulty in model-based clustering. First, mixing outliers in a “legitimate” cluster leads to the “contamination” of the model. Second, even if the algorithm is capable of isolating the outliers, they lead to a diversion of the model parameter resource. Thus when there are outliers, a model-based clustering algorithm that attempts to account for all the data with models will often only identify the outliers, at the expense of failing to isolate some of the dominant regimes.

**Table 2. Sample Reassignment**

|   |
|---|
| <p><b>repeat</b></p> <p>  Compute the acceptance threshold for each model using Monte Carlo simulation</p> <p>  For each sequence <math>x_i</math>, identify model <math>j</math> having maximum likelihood;</p> <p>  If likelihood <math>&gt;</math> acceptance threshold then assign <math>x_i</math> to cluster <math>j</math></p> <p>  Otherwise assign <math>x_i</math> to outlier pool.</p> <p>  Apply <b>Model Construction</b> to each cluster with changed membership</p> <p><b>until</b> no more change of cluster membership</p> |
|---|

To resolve this problem, we introduce a mechanism called *the outlier pool*, detailed in the **Sample Reassignment/Outlier Detection** module in Table 2. Instead of attempting to account for all sequences with HMM models, we allow each model to reject a sequence whose likelihood is too low. A sequence that is rejected by all current models is placed in the outlier pool. The outlier pool is a special “garbage” cluster which is not modeled. Note that this outlier pool is dynamic: objects can enter into or exit from the outlier pool as the clustering algorithm proceeds.

The threshold used to determine whether a sequence should be accepted or rejected by a model is selected based on the expected likelihood of each model, estimated using Monte Carlo simulation. For each model, 500 sequences are generated according to the model parameters. Then the normalized likelihood of each sequence against the given model is computed and the average is taken as the expected likelihood of the model. The threshold is set such that sequences whose likelihood values are significantly below the expected likelihood for their cluster are flagged as outliers.

### 2.3 Partition growing using DTW

As shown in Table 1, the algorithm starts with one cluster and incrementally grows the number of clusters until the partition BIC measure reaches a maximum point. For each given number of clusters, the initial set of clusters and models are adjusted using an EM procedure as outlined in Table 2. Since the EM algorithm will only converge to a local optimal point, its outcome greatly depends on the initial partition. Thus a crucial step in a top-down model based clustering algorithm is the initialization of a new cluster from an existing set of clusters.

One possible strategy is to seed the new cluster with the data sample that is “least fit”, i.e., farthest away from all current models [2]. While this works reasonably well for clean data, it is sensitive to outliers. When there are outliers in the data, the data sample that is farthest away from all models is very likely an outlier. Using this strategy the

cluster growing process tends to be dominated by outliers.

We have adopted a more robust alternative. Instead of evaluating each individual sequence for fitness, each cluster is evaluated as a whole. The cluster with the lowest average likelihood is identified as the candidate for splitting. The identified cluster is then split into two new clusters using hierarchical clustering based on distance measures computed using DTW.

## 3. Experiments

Synthesized data were generated to systematically evaluate our algorithm and compare its performance with that of other methods. We used discrete-value, left-to-right HMMs and segmental  $k$ -means algorithm for model estimation [4]. It should be noted, however, that our approach is not predicated on these choices: the algorithm and the analysis apply to more general HMMs and different choices of HMM training techniques as well.

### 3.1 Data description

To generate a synthesized data set, we specify the parameters of the HMM represented in each cluster, along with the number of clusters and their sizes and then generate the desired number of sequences. Singleton clusters or clusters with a small number of samples are used to simulate outliers.

The HMMs used to generate our synthesized data contain two to three states. To generate different emission probability distributions, we first generated 6 normal distributions with deviation of 0.5 and means of  $i * 2.0, 1 \leq i \leq 6$ . We then calculated the probabilities for each 1.0 interval between 0.5 and 12.5 for each distribution, arriving at 6 distinct discrete probability distributions for 13 symbols. The emission probability distributions for all generating HMMs are selected from these 6 distributions. The distance between any two HMMs is controlled by the number of states that have shared emission probabilities and the self-transition probabilities of these states.

Instead of using a fixed length for all sequences as in previous methods [8, 3, 2], we allow the length of the sequences to vary, to more closely simulate the situation in most applications. Since the HMMs are left-to-right models with a forced initial and final state, the expected sequence length for each model is essentially determined by the self-transition probabilities for the states. We adjusted these transition probabilities such that all models have an expected sequence length of 50, and allowed individual sequence lengths to vary between 30 and 100.

Two synthetic data sets were used in our evaluations, generated using 10 HMMs. Models 1 to 5 (referred to as *major models*) were used to generate dominant regimes and

**Table 3. Performance comparison**

|       | PMC measure |         | DCM measure |         |
|-------|-------------|---------|-------------|---------|
|       | Matryoshka  | HMM/DTW | Matryoshka  | HMM/DTW |
| Set 1 | 126         | 16      | 0.478       | 0.083   |
| Set 2 | 122         | 10      | 0.413       | 0.026   |

models 6 to 10 (referred to as *noise models*) were used to simulate outliers. For both data sets, the sizes for the major clusters are 100,60,30,30,10 respectively. For outliers, the first data set has 5 singletons while the second one has 5 minor clusters with sizes 4,3,2,1,1.

### 3.2 Performance measures

Two performance measures were used to quantitatively assess the performance of our clustering algorithm. The first is the Partition Misclassification Count (PMC) [2], which is a weighted sum of all different types of object misclassifications that occur in the derived partition. The smaller the count, the closer the derived partition is to the true partition and thus more accurate the clustering algorithm.

Another performance, the Difference of Concordance Matrix (DCM), measures the mismatch between the true and derived partitions. Given a set of  $N$  objects, the concordance matrix  $C$  is a 0-1  $N \times N$  matrix where  $c_{ij} = 1$  if the  $i$ th and  $j$ th objects are in the same cluster and  $c_{ij} = 0$  otherwise. The DCM measure is then defined as:  $DCM = \frac{e^T(|C_t - C_d|)e}{e^T C_t e}$ , where  $e$  is vector of ones,  $C_t$  and  $C_d$  are concordance matrices for the true and derived partitions, respectively, and  $|\bullet|$  denotes the component-wise absolute value of a matrix. Values of DCM range from 0 to 1 with 0 indicating a perfect match and 1 indicating a complete mismatch.

### 3.3 Experimental results

We evaluated our algorithm using the two synthetic data sets described in Section 3.1, and compared the results to those obtained using the Matryoshka algorithm developed by Li *et.al.* [2]. Table 3 shows the performance measures of both algorithms. As can be seen from the table, our algorithm significantly outperforms the Matryoshka algorithm in both measures for both data sets.

Looking in more detail for data set 2, we found that 100% of sequences from Models 1 and 5 were clustered correctly, while 5 sequences (of 10 total) from Models 6, 9, and 10 (the outlier clusters) were mixed in with major clusters from Models 2,4, and 5, respectively. The remaining outlier sequences were identified correctly.

In contrast, the Matryoshka algorithm produced a total of 8 clusters, yet failed to isolate all of the major groups:

the algorithm was unable to distinguish sequences generated by Models 2 and 3, grouping 55 (of 60 total) sequences from Model 2 together with the 60 sequences from Model 3. Another identified cluster was spurious, consisting of sequences from Models 2 plus four outlier sequences.

## 4 Conclusions and Future Work

In this paper, we have introduced refinements to existing HMM-based clustering schemes to address important shortcomings. In particular, our algorithm interweaves clustering based on a DTW-based distance measure with an HMM model-based approach, and allows for identification of outlier sequences in such a way that they do not detract from the identification of the major clusters. Experimental results with synthetic data demonstrate that these adaptations provide significant performance improvements.

Several open questions remain. First, what is the impact of the particular HMM model fitting algorithm on the cluster results? Second, does the improved performance carry over to the case of continuous HMM and/or unconstrained HMM models? Finally, we would like to understand how explicit modeling of the state durations impacts the final cluster results, compared to characterizing durations implicitly through the HMM transition probabilities. Better understanding of these issues will aid in identification of applications where the proposed technique will be most useful.

## References

- [1] E. Dermatas and G. Kokkinakis. Algorithm for clustering continuous density HMM by recognition error. *IEEE Trans. Speech Audio Processing*, 4(3):231–234, May 1996.
- [2] C. Li, G. Biswas, M. Dale, and P. Dale. Matryoshka: A HMM based temporal data clustering methodology for modeling system dynamics. *Intelligent Data Analysis*, pages 281–308, June 2002.
- [3] T. Oates, L. Firoiu, and P. Cohen. Using dynamic time warping to bootstrap HMM-based clustering of time series. In R. Sun and C. Giles, editors, *Sequence Learning, LNAI 1828*, pages 35–52. Springer-Verlag, 2000.
- [4] L. Rabiner and B. Juang. *Foundations of speech recognition*. Prentice Hall, 1993.
- [5] L. Rabiner, C. Lee, B. Juang, and J. Wilpon. HMMs clustering for connected work recognition. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1989.
- [6] A. Schliep, A. Schonhuth, and C. Steinhoff. Using hidden Markov models to analyze gene expression time course data. *Bioinformatics*, 19, 2003.
- [7] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- [8] P. Smyth. Clustering sequences with hidden Markov models. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing*, page 648. The MIT Press, 1997.