

# Identifying Qualitatively Different Outcomes of Actions: Experiments with a Mobile Robot

Tim Oates, Matthew D. Schmill and Paul R. Cohen

Computer Science Department, LGRC  
University of Massachusetts, Box 34610  
Amherst, MA 01003-4610  
{oates,schmill,cohen}@cs.umass.edu  
Voice: 413-577-0669  
Fax: 413-545-1249

## Abstract

We present an unsupervised learning method that allows a situated embodied agent to identify and represent qualitatively different outcomes of actions. The initiation of a particular action triggers the collection of multivariate time series of sensor values. Those time series are clustered using Dynamic Time Warping as a measure of similarity, and prototypes are extracted from the clusters. Each prototype represents a distinct experience, i.e. one possible outcome of engaging in the activity. Prototypes can be used for off-line planning and for on-line prediction by finding the best partial match among the prototypes to current sensor readings. Experiments with a Pioneer-1 mobile robot demonstrate the utility of the approach with respect to capturing the structure and dynamics of a complex, real-world environment.

## Introduction

If situated embodied agents are to engage in any of a number of cognitive tasks, they must have the ability to construct models of their physical environment.<sup>1</sup> In all but the simplest, static domains, such models must represent the dynamics of environmental change. For example, planning requires knowledge of how actions affect the state of the world. Embodied agents cannot make an assumption commonly made by classical planners, that actions and their effects are instantaneous. They must consider actions that take substantial amounts of time to complete, and effects that have similar temporal extents.

This paper presents a method for unsupervised learning of models of environmental dynamics based on clustering of multivariate time series. An unsupervised learning approach to this problem is desirable because hand-coding accurate models of dynamic, stochastic environments is a daunting, if not impossible, task. Experiments with a Pioneer-1 mobile robot demonstrate the utility of the method.

Individual time series are obtained by recording the output of a subset of the robot's sensors. We call these time series *experiences*. An example of a sensor subset

<sup>1</sup>Obviously there are those that would disagree with this position (Brooks 1987), but it is one that we adhere to.

on the Pioneer-1 is an array of seven sonars. Each sonar returns the distance to the closest object in the direction that it points. Recording of time series is usually triggered by events, such as the initiation of a particular action. As shown in Figure 1, each time a given event occurs, the time series that was recorded is added to a bucket associated with that event. Once a sufficient number of time series are recorded, clusters can be formed. Clustering requires a measure of similarity between multivariate time series. One such measure that is particularly appropriate for this problem is Dynamic Time Warping (DTW) (Sankoff and Kruskal 1983). (We discuss DTW in detail in a later section.) Each cluster can then be represented by a prototype, either the cluster centroid or an average of its members.

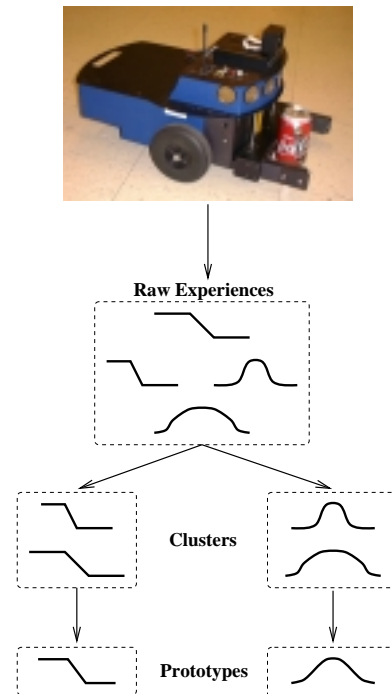


Figure 1: The formation of prototypes of qualitatively different experiences.

Cluster prototypes formed in this manner are useful for a variety of purposes. If the event driving the collection of time series was an action, cluster prototypes correspond to qualitatively different outcomes of engaging in that action. As such, they can be used for off-line planning and for on-line prediction by finding the best partial match among the prototypes to current sensor readings.

Although our work in unsupervised clustering of time series is driven primarily by an interest in situated embodied cognition, it has many potential commercial applications as well. There is clearly utility in identifying qualitatively different regimes in multivariate time series produced by financial markets, monitors in an intensive care unit, and earth orbiting satellites that measure changes in the earth’s weather.

The remainder of the paper is organized as follows. The next section describes our method for clustering time series in detail, including a discussion of Dynamic Time Warping, the particular clustering algorithm used, and prototype formation. We then present an evaluation of the method as applied to the Pioneer-1 mobile robot. The last two sections review related work, pointing out the connection between Dynamic Time Warping and Hidden Markov Models, and outline future research, respectively.

## Clustering Experiences

This section presents our method for unsupervised learning of models of environmental dynamics based on clustering of multivariate time series. To ground the discussion, consider the Pioneer-1 mobile robot. Its sensors include, among others, a bump switch on the end of each gripper paddle that indicates when the gripper hits an object, an infrared break beam between the gripper paddles that indicates when an object enters the gripper, and wheel encoders that measure the rate at which the wheels are spinning.

Suppose the robot is moving forward at a fixed velocity. Collectively, the values returned by the sensors mentioned above can discriminate many different situations. For example, if the robot runs into a large immovable object, such as a wall, the bump sensors go high and the wheel velocities abruptly drop to zero. If it bumps into a trash can, which is large but movable, the bump sensors go high and the wheel velocities remain constant. If it comes across an object that can be grasped, the break beam goes high when the object enters the gripper and there is no change in wheel velocity. As observers of the robot’s actions, we can label and categorize its experiences. Our goal is to provide mechanisms that will allow the robot to perform that task by itself.

Let  $E$  denote an experience, a multivariate time series containing  $n$  measurements from a set of sensors such that  $E = \{e_t | 1 \leq t \leq n\}$ . The  $e_i$  are vectors of values containing one element for each sensor. Given a set of  $m$  experiences, we want to obtain, in an unsupervised manner, a partition into subsets of experiences

such that each subset corresponds to a qualitatively different type of experience. Given such a partition, reasoning with entire sets of experiences is unwieldy, so a simpler representation such as the average experience in a subset is required.

If an appropriate measure of the similarity of two time series is available, clustering followed by prototype extraction is a suitable unsupervised learning method for this problem (see Figure 1). Finding such a measure of similarity is difficult because experiences that are qualitatively the same may be quantitatively different in at least two ways. First, they may be of different lengths, making it difficult or impossible to embed the time series in a metric space and use, for example, Euclidean distance to determine similarity. Second, within a single time series, the rate at which progress is made can vary non-linearly. For example, the robot may move slowly or quickly toward a wall, leading to either a slow or rapid decrease in the distance returned by its forward-pointing sonar. In each case, though, the end result is the same, the robot bumps into the wall. Such differences in rate make similarity measures such as cross-correlation unusable.

The measure of similarity that we use is Dynamic Time Warping (DTW) (Sankoff and Kruskal 1983). It is ideally suited for the time series generated by a robot’s sensors. DTW is a generalization of classical algorithms for comparing discrete sequences (e.g. minimum string edit distance (Corman *et al.* 1990)) to sequences of continuous values. It was used extensively in speech recognition, a domain in which the time series are notoriously complex and noisy, until the advent of Hidden Markov Models which offered a unified probabilistic framework for the entire recognition process (Jelinek 1997).

Given two experiences,  $E_1$  and  $E_2$  (more generally, two continuous multivariate time series), DTW finds the warping of the time dimension in  $E_1$  that minimizes the difference between the two experiences. Consider the two univariate time series shown in Figure 2. Imagine that the time axis of  $E_1$  is an elastic string, and that you can grab that string at any point corresponding to a time at which a value was recorded for the time series. Warping of the time dimension consists of grabbing one of those points and moving it to a new location. As the point moves, the elastic string (the time dimension) compresses in the direction of motion and expands in the other direction. Consider the middle column in Figure 2. Moving the point at the third time step from its original location to the seventh time step causes all of the points to its right to compress into the remaining available space, and all of the points to its left to fill the newly created space. Of course, much more complicated warpings of the time dimension are possible, as with the third column in Figure 2 in which four points are moved.

Given a warping of the time dimension in  $E_1$ , yielding a time series that we will denote  $E'_1$ , one can compare the similarity of  $E'_1$  and  $E_2$  by determining the area between the two curves. That area is shown in gray in the

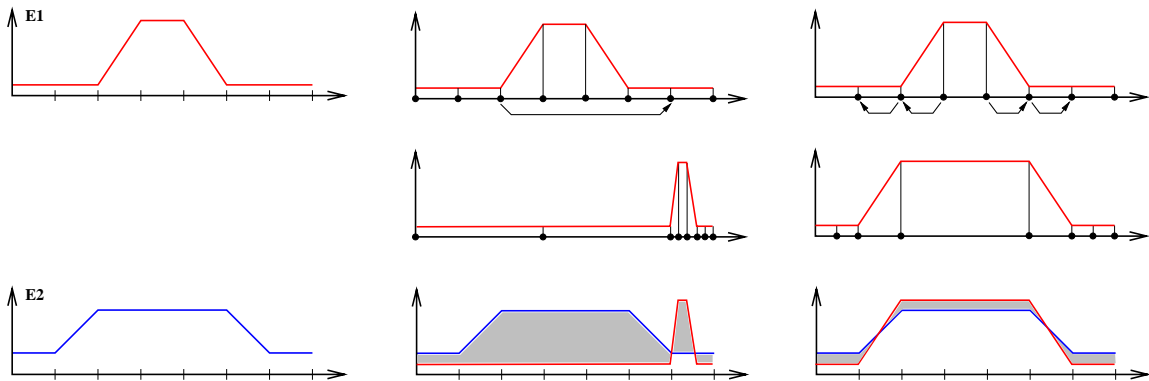


Figure 2: Two time series,  $E_1$  and  $E_2$ , (the leftmost column) and two possible warpings of  $E_1$  into  $E_2$  (the middle and rightmost columns).

bottom row of Figure 2. Note that the first warping of  $E_1$  in which a single point was moved results in a poor match, one with a large area between the curves. However, the fit given by the second, more complex warping is quite good. In general, there are exponentially many ways to warp the time dimension of  $E_1$ . DTW uses dynamic programming to find the warping that minimizes the area between the curve in time that is a low order polynomial of the lengths of  $E_1$  and  $E_2$ , i.e.  $O(|E_1||E_2|)$ .

DTW returns the optimal warping of  $E_1$ , the one that minimizes the area between  $E_1'$  and  $E_2$ , and the area associated with that warping. The area is used as a measure of similarity between the two time series. Note that this measure of similarity handles nonlinearities in the rates at which experiences progress and is not affected by differences in the lengths of experiences. In general, the area between  $E_1'$  and  $E_2$  may not be the same as the area between  $E_2'$  into  $E_1$ . We use a symmetrized version of DTW that essentially computes the average of those two areas based on a single warping (Kruskall and Liberman 1983). Although a straightforward implementation of DTW is more expensive than computing Euclidean distance or cross-correlation, there are numerous speedups that both improve the properties of DTW as a distance metric and make its computation nearly linear in the length of the time series with a small constant.

Given  $m$  experiences, we can construct a complete pairwise distance matrix by invoking DTW  $m(m-1)/2$  times (the factor of 2 is due to the use of symmetrized DTW). We then apply a standard hierarchical, agglomerative clustering algorithm that starts with one cluster for each experience and merges the pair of clusters with the minimum average intercluster distance (Everitt 1993). Without a stopping criterion, merging will continue until there is a single cluster containing all  $m$  experiences. To avoid that situation, we do not merge clusters for which the mean intercluster distance is significantly different from the mean intracluster distance as measured by a t-test.

Finally, for each cluster we select a prototype. Two

methods commonly used are to choose the cluster member that minimizes the distance to all other members of the cluster, or to simply average the members of the cluster. The advantage of the latter method is that it smooths out noise that may be present in any individual data item. Unfortunately, it is only workable when the cluster elements are embedded in a metric space (e.g. Cartesian space). Although we cannot embed experiences in a metric space, DTW allows us to use a combination of the two methods as follows. First, we select the time series that minimizes distance to all other time series in a given cluster. Then we warp all other patterns into that centroid, resulting in a set of patterns that are all on the same time scale. It is then a simple matter to take the average value at each time point over all of the series and use the result as the cluster prototype.

## Evaluation

We are interested in using the clusters produced by the combination of DTW and agglomerative clustering for the purposes of planning, so we would like clusters to map to action outcomes for planning. As such, our primary means of evaluating cluster quality is to compare them against clusters generated manually by the experimenter who designed the experiences they comprise.

Data was collected for 4 sets of experiences: 102 experiences with the robot moving in a straight line while collecting data from the velocity encoders, break beams, and gripper bumper (which we will call the *tactile* sensors), 102 move experiences collecting data from the Pioneer's vision system, including the X and Y location, area, and distance to a single visible object being tracked (which we will call the *visual* sensors), 50 experiences with the robot turning in place collecting tactile data, and 50 turn experiences collecting visual data. In each experience, the robot attempted to move or turn for a duration between 2 and 8 seconds in the laboratory environment. Visible objects and objects that impeded or obstructed the robot's path were present in many of the trials.

move/tactile	turn/tactile	move/visual	turn/visual
+250 unobstructed	+100 unobstructed	no object	no object
+100 unobstructed	+100 never stops	heavy noise	can't move
-100 unobstructed	+100 bump	approach on right	pass left to right
-250 unobstructed	+100 blocked	approach disappear	pass right to left
+250 temporary bump	+100 temporary bump	discover left reverse	discover right
+100 temporary bump	+100 blocked bump	vanish on right	discover left
+250 push delayed bump	-100 unobstructed	vanish on left	left to right
+250 delayed bump	-100 temporary bump	retreat left	vanish off right
+100 delayed bump	-100 impeded turn	discover right	vanish off left
+250 crash beam1	-100 blocked	approach ahead	
+250 squash		approach, gets big	
+250 push blocked		approach on left	
+250 push		approach, stays small	
+100 push			
+100 push shallow			
+100 blocked			
-100 blocked			

Figure 3: Outcome labels given to the hand built clusters for each of the 4 experience sets.

The labels given to the hand-built clusters generated are summarized in table 3. In the visual tracking problems, the clusters correspond to visible objects’ relations to the agent during activity; the object may move across the visual field while turning or it may loom while being approached. In the tactile problems, clusters correspond to the Pioneer’s velocity and the types of contact made with objects in the environment during the activity; heavy objects halt the Pioneer’s progress, and are labeled “crash”, while light, small objects merely trigger the break beams and are labeled “push”.

We evaluate the clusters generated by DTW and agglomerative clustering with a  $2 \times 2$  contingency table called an *accordance table*. Consider the following table:

	$t_e$	$\neg t_e$
$t_t$	$n_1$	$n_2$
$\neg t_t$	$n_3$	$n_4$

We calculate the cells of this table by considering all pairs of experiences  $e_j$  and  $e_k$ , and their relationships in the target (hand-built) and evaluation (DTW) clusterings. If  $e_j$  and  $e_k$  reside in the same cluster in the target clustering (denoted by  $t_t$ ), and  $e_j$  and  $e_k$  also reside in the same cluster in the evaluation clustering (denoted by  $t_e$ ), then cell  $n_1$  is incremented. The other cells of the table are incremented when either the target or evaluation clusterings places the experiences in different clusters ( $\neg t_t$  and  $\neg t_e$ , respectively).

Cells  $n_1$  and  $n_4$  of this table represent the number of experience pairs in which the clustering algorithms are in accordance. We call  $n_1 + n_4$  the number of *agreements* and  $n_2 + n_3$  the number of *disagreements*. The *accordance ratios* that we are interested in are  $\frac{n_1}{n_1 + n_2}$ , accordance with respect to  $t_t$ , and  $\frac{n_4}{n_3 + n_4}$ , accordance with respect to  $\neg t_t$ .

Table 4 shows the breakdown of accordance for the combination of dynamic time warping and agglomerative clustering versus the ideal clustering built by hand. The column labeled “#” indicates the difference between the number of hand-built and automated clusters. In each problem, the automated algorithm clustered more aggressively, resulting in fewer clusters. The columns that follow present the accordance ratios for experiences grouped together, apart, and the total number of agreements and disagreements.

The table shows very high levels of accordance. Ratios ranged from a minimum of 82.2% for experiences clustered together ( $t_t$ ) in the move/visual set to 100% for experiences clustered together in the turn problems. For the turn problems, the aggressive clustering may account for the high  $t_t$  accuracy, causing slightly lower accuracy in the  $\neg t_t$  case.

The disparity in the number of clusters suggests that tuning the parameters of the clustering algorithm to produce more clusters might boost  $\neg t_t$  accuracy while preserving the  $t_t$  accuracy. The table for this condition is omitted for the sake of brevity, but our findings were that tuning the clustering algorithm in this way leads to a reduction in accuracy in all but the turn/tactile dataset, whose  $\neg t_t$  accuracy increased 7 points.

The failure of this strategy to increase  $\neg t_t$  accuracy by tuning the clustering algorithm to terminate with more clusters indicates that it is not simply a matter of the number of clusters. Exploration of the  $t_t$  disagreements in the move/visual data, the problem with the highest error rate, indicates that 132 out of the 156 errors can be traced to two clusters in the automatically generated set that were distributed differently in the target set. The target clusters were “no object” (no visible object being tracked, some minor noise) and “heavy noise” (noise makes it unclear whether anything was being tracked). The automated set had made the split differently; experiences with any noise were grouped to-

	#	$t_t$	$t_t \wedge t_e$	%	$\neg t_t$	$\neg t_t \wedge \neg t_e$	%	Agree	Disagree	%
Move visual	-5	876	720	<b>82.2</b>	4275	4125	<b>96.4</b>	4845	306	<b>94.0</b>
Move tactile	-7	443	378	<b>85.3</b>	4708	4468	<b>95.0</b>	4846	305	<b>94.0</b>
Turn visual	-5	262	262	<b>100.0</b>	599	571	<b>95.3</b>	833	28	<b>96.7</b>
Turn tactile	-6	163	163	<b>100.0</b>	698	593	<b>85.0</b>	756	105	<b>87.8</b>

Figure 4: Accordance statistics for automated clustering against the hand built clustering.

gether from those that had none. The remaining 24 errors were covered by a handful of six or seven experiences that were also attracted into clusters by experiences the hand builder did not feel were similar.

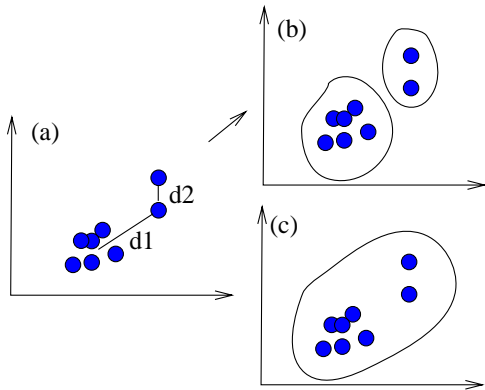


Figure 5: (a) A 2d representation of experiences. (b) The ordering effect of greedily merging based on the shorter distance  $d_2$  than the group average distance  $d_1$ . (c) The most desirable clustering.

The problem is rooted in the tendency of greedy clustering algorithms to suffer from *ordering effects* (Fisher *et al.* 1992). In clustering schemes based on sorting, the order in which instances are considered biases the clusters that result. In agglomerative clustering algorithms, the clusters that result are biased by the algorithm’s greedy choice of always considering merging the lowest distance clusters. Figure 5 illustrates how the ordering effect works on a 2d representation of the move/visual data. Because two of the noisy data are very similar (distance= $d_2$ ), they are clustered together early in the clustering process. This early decision creates two cluster centers that individually attract members based on the local greedy policy, where a global view (like our hand-builder’s) would cluster them together.

Fortunately, optimization techniques exist that can refine initial clusterings to better reflect a global view (Fisher 1996). We have implemented a simple optimization technique which iteratively reassigns experiences to neighboring clusters if a cluster is found with a smaller group average distance than to the one the experience is in. After applying this optimization technique to the clusters used to generate table 4, many of the errors in the  $t_t$  cases disappeared: accordance climbed to 91.9% or better in all cases except the  $\neg t_t$  case of

turn/tactile, which decreased to below 80%, which reflects the disparity between the number of clusters generated by our algorithm and the hand built clustering.

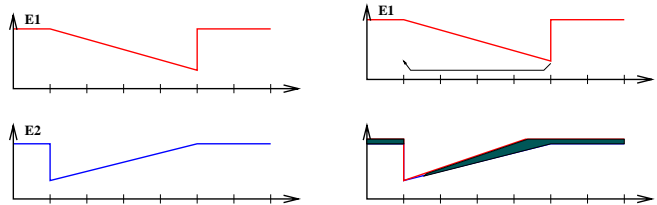


Figure 6: Two time series,  $E_1$  and  $E_2$ , and a possible warping of  $E_1$  into  $E_2$  that obscures the salient difference.

The remaining few percent of misses appear to be related to dynamic time warping’s ability to manipulate the time dimension. Figure 6 illustrates two time series that correspond to the horizontal location of an object on the Pioneer’s visual plane. In experience  $e_1$ , the object comes into view from the right, passes across, and disappears off the left side of the visual plane. Experience  $e_2$  represents an object moving in the opposite direction across the visual plane. Clearly, this is a salient distinction for many purposes, including planning, but it is one that DTW is able to obscure by sliding a single point of  $e_1$  backward in time.

## Related Work

The use of DTW as a measure of similarity between multivariate time series dates back a number of years to early work in speech recognition (Sakoe and Chiba 1978), although it was ultimately displaced by HMM’s (Jelinek 1997). HMM’s are actually a powerful generalization of DTW, and recent years have seen renewed interest in DTW for applications where the full power of HMM’s may not be required (Berndt and Clifford 1994). That fact notwithstanding, it is unclear how one would apply standard HMM algorithms (such as Baum-Welch and Viterbi) directly to clustering time series. One recent attempt (Smyth 1997) at that problem is much more complex, both computationally and descriptively, than our application of DTW and requires a priori knowledge of the number of clusters (although a method for attempting to determine that number is presented).

Other approaches to measuring similarity between continuous time series have been proposed in the literature (Agrawal *et al.* 1995; Keogh and Pazzani 1998).

However, these approaches are limited to univariate time series and are therefore not applicable to our problem, in which one sensor alone is insufficient to discriminate between experiences.

## Conclusion

We have presented an approach to clustering the experiences of a mobile robot in a complex, stochastic environment. Using Dynamic Time Warping as a measure of the similarity between time series sensor data, we are able to use agglomerative clustering to produce clusters based on the dynamics of experiences, rather than static features.

We evaluated the effectiveness of the unsupervised clustering algorithm by measuring the *accordance* of the clusters it generated with cluster sets generated by hand as an answer key. Accordance statistics are based on pairwise comparisons of experiences; they measure the number of times the two clusterings agree on whether pairs of experiences  $e_j$  and  $e_k$  should be clustered together or apart. Using only Dynamic Time Warping and agglomerative clustering on 150 trials of real Pioneer data in a variety of experiences, we measured 82-100% accordance between the automated and hand-built clusterings. By applying a simple iterative optimization algorithm to the initial clusterings, accordance measures increased to 91.9% and better.

Still, pathological cases exist where Dynamic Time Warping was able to find a temporal mapping that glossed over significant differences in time series exist. Though these cases cover only a small percentage of the robot's practical experiences, it is possible to constrain DTW so that these differences will be felt through the distance metric.

Future work will extend the approach described above in two ways. First, rather than using each experience in its entirety, we will develop methods for identifying subsequences within the experiences that are relevant to the clustering process. Second, we intend to leverage the relationship between DTW and HMM's to develop a method of clustering time series in which the output is a set of HMM's, one for each cluster.

## References

R. Agrawal, K. Lin, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling and translation in time series databases. In *Proceedings of the 21st International Conference on Very Large Databases*, 1995.

Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Working Notes of the Knowledge Discovery in Databases Workshop*, pages 359–370, 1994.

R. A. Brooks. Intelligence without representation. In *Proceedings of the Workshop on the Foundations of Artificial Intelligence*, Cambridge, MA, 1987. MIT Press.

T. H. Corman, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

Brian Everitt. *Cluster Analysis*. John Wiley & Sons, Inc., 1993.

D. H. Fisher, L. Xu, and N. Zard. Ordering effects in clustering. In *Proceedings of the 9th Annual Conference on Machine Learning*, pages 163–168, 1992.

Doug Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research*, 4:147–179, 1996.

Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

Eamonn Keogh and Michael J. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Working Notes of the AAAI-98 workshop on Predicting the Future: AI Approaches to Time-Series Analysis*, pages 44–51, 1998.

Joseph B. Kruskal and Mark Liberman. The symmetric time warping problem: From continuous to discrete. In *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 26:143–165, 1978.

David Sankoff and Joseph B. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: Theory and Practice of Sequence Comparisons*. Addison-Wesley Publishing Company, Reading, MA, 1983.

Padhraic Smyth. Clustering sequences with hidden markov models. In *Advances in Neural Information Processing 9*, 1997.