Data Min Knowl Disc (2011) 22:232–258
DOI 10.1007/s10618-010-0193-7

Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction

Amy McGovern · Derek H. Rosendahl · Rodger A. Brown · Kelvin K. Droegemeier

Received: 25 February 2010 / Accepted: 5 July 2010 / Published online: 29 July 2010

Abstract We introduce an efficient approach to mining multi-dimensional temporal streams of real-world data for ordered temporal motifs that can be used for prediction. Since many of the dimensions of the data are known or suspected to be irrelevant, our approach first identifies the salient dimensions of the data, then the key temporal motifs within each dimension, and finally the temporal ordering of the motifs necessary for prediction. For the prediction element, the data are assumed to be labeled. We tested the approach on two real-world data sets. To verify the generality of the approach, we validated the application on several subjects from the CMU Motion Capture database. Our main application uses several hundred numerically simulated supercell thunderstorms where the goal is to identify the most important features and feature interrelationships which herald the development of strong rotation in the lowest altitudes of a storm. We identified sets of precursors, in the form of meteorological

Responsible editor: Eamonn Keogh.

A. McGovern (✉)
School of Computer Science, University of Oklahoma, Norman, OK 73019, USA
e-mail: amcgovern@ou.edu

D. H. Rosendahl · K. K. Droegemeier
School of Meteorology, University of Oklahoma, 120 David L. Boren Blvd., Suite 5900, Norman, OK 73072-73071, USA

D. H. Rosendahl
e-mail: drose@ou.edu

K. K. Droegemeier
e-mail: kkd@ou.edu

R. A. Brown
NOAA/National Severe Storms Laboratory, 120 David L. Boren Blvd., Norman, OK 73072, USA
e-mail: Rodger.Brown@noaa.gov

quantities reaching extreme values in a particular temporal sequence, unique to storms producing strong low-altitude rotation. The eventual goal is to use this knowledge for future severe weather detection and prediction algorithms.

Keywords Temporal data mining · Multi-dimensional · Severe weather

1 Introduction

This work is motivated by the real-world problem of tornado prediction. Despite considerable progress made in recent decades in the observation, modeling, and theoretical understanding of tornadoes, warning and forecasting remains a considerable challenge. Prediction statistics have plateaued in recent years primarily because existing surveillance radars and hazardous weather detection methodologies suffer from fundamental limitations that allow key meteorological quantities and associated features to go undetected (e.g., Burgess et al. 1993; Brotzge et al. 2006; Rosendahl 2008). New advances will be required if substantial improvements in warning and forecasting accuracy are to take place. The long-term goals of our work in spatiotemporal data mining are to revolutionize our understanding of how tornadoes form and to use this to develop new techniques that can improve current predictive capabilities. The work presented in this paper is the first step in our research on developing spatiotemporal data mining algorithms with a focus on the application to severe weather (McGovern et al. 2008; Supinie et al. 2009). In this paper, we develop a temporal mining approach that identifies the key precursors to the development of strong rotation in the lower levels of the atmosphere in simulated supercell thunderstorms. This paper focuses on the data mining component of our work and Rosendahl et al. (in preparation) focus on the meteorological analysis.

The work presented here is the groundwork for our current research where we are developing spatiotemporal data mining models that will enable us to understand the interaction of weather features across both space and time. In this paper, we identify the critical dimensions of the time series data and then identify a predictive set of temporal signatures or motifs within these dimensions. These motifs are combined temporally to form a set of rules such as "if the value of storm feature x increases followed by a decrease in the value of storm feature y and storm feature z maintaining its value, then there is an 80% chance of a strong low-altitude rotation occurring within the next 30 min."

The main contributions of this work are: (1) An efficient approach to identifying predictive motifs in multi-dimensional temporal streams, (2) Verification of this approach on two real-world data sets. The efficiency is critical and introduced through both intelligent data structures and admissible pruning. Without the ability to prune, motif discovery on such high dimensional data would not be possible in a reasonable amount of time. By verifying the approach on two real-world data sets, we demonstrate that the approach is viable in large and realistic data sets.

The most closely related work to our approach is that of time series motif or shapelet discovery (e.g. Das et al. 1998; Chiu et al. 2003; Mueen et al. 2009; Ye and Keogh 2009). Das et al. (1998) provide preliminary work for a problem very similar to ours:

that of finding rules in multi-dimensional (or multi-variate) real-valued time series data. However, this work has been put into doubt by other findings that show that local patterns or motifs in unlabeled data are essentially random (Keogh et al. 2003; Denton 2005; Idé 2006; Goldin et al. 2006). Although our goals of finding local patterns remain, we are using labeled time-series data rather than searching for repeating motifs. A second difference from their work is in how we discretize the data in order to find patterns. They use clustering where we use the SAX discretization approach (Lin et al. 2003, 2007) (described below). Chiu et al. (2003) focus on identifying novel (and unknown to the user) motifs from single-dimensional time series. They include the ability to handle noise in the data. Mueen et al. (2009) build on this work with the goal of exactly and efficiently identifying recurring subsections of a time series. These recurring subsections are called motifs and are very similar to our definition of motifs. There are two main differences between their work and ours. First, we identify motifs across multiple dimensions of the data and identify the most relevant dimensions (while ignoring the irrelevant ones). Their work focuses on a single dimension of the data only. Second, we use labeled data and look for motifs that can be used for prediction, rather than description. Their motifs are focused on the descriptive case with unlabeled data.

Our approach uses the SAX discretization technique (Lin et al. 2003, 2007) to efficiently index our time streams and to create data structures for search. However, their work focuses on single-dimensional time series analysis whereas we have a hundred dimensions to analyze simultaneously. Our approach does not limit the number of dimensions at 100, this is simply imposed by the severe weather data that we studied. Multi-dimensional time series analysis techniques have been proposed but they generally focus on the problem of indexing the data rather than the task of feature selection and identifying predictive motifs. For example, the multi-dimensional indexing approach of Vlachos et al. (2006) enables efficient queries and search for spatiotemporal time series where all of the dimensions matter in the query. Our data differs in that each dimension is independent of the others and therefore a multi-dimensional index is not necessary. Likewise, the work of Kahveci et al. (2002) focuses on non-Euclidean similarity metrics for multi-dimensional data. Lee et al. (2000) assume that all dimensions of the time series are critical. Although Faloutsos et al. (1997) study temporal signatures, their signatures are a precursor to the efficient indexing approaches of Lin et al. (2003) and not the temporal motifs that we identify. There are many other time series analysis approaches that address related problems (e.g. Tanaka and Uehara 2003; Yin and Gaber 2008; Cheng and Tan 2008; Kasetty et al. 2008) but none of these address the issue of multi-dimensional motif discovery with the additional task of identifying the most relevant dimensions of the data. Recent work on indexable SAX (Shieh and Keogh 2009) has introduced a very efficient way to store and search large time series and this work could make our approach even more efficient.

Our approach for growing motifs draws from the efficient approaches of Agrawal and Srikant (1994), Oates (1999), Zaki (2001), and Zaki et al. (2005). Both Agrawal and Zaki discuss the requirement for longer sequences to be supported by shorter sequences, which enables efficient pruning of the search space. We are able to make use of this principle to efficiently prune the initial search space and continue to search only by building on the shorter sequences. Similarly, Oates (1999) and

Oates et al. (1998) mine temporal streams of data efficiently by making use of the statistics of the evaluation metric to perform admissible pruning of the search space (similar to Webb 1995; McGovern and Jensen 2008). In addition to building on the shorter sequences, we use this idea to further prune our search. Also, similar to the doubling approach described by Zaki (2001), we double the length of each motif until it fails to double and then grow linearly within the failed region.

The next section presents our approach in detail and the following section focuses on our two empirical results: motion capture and severe weather. We finish with a discussion and directions for future work.

2 Approach

Informally, the goal of our approach is to identify the most relevant dimensions of a multi-dimensional time series, grow a set of predictive rules from motifs discovered in each of those dimensions, and use these to improve our understanding of the data and for prediction. In this section, we formalize our approach and definitions.

Definition 1 A time series $T = \langle t_1, t_2, \ldots, t_{n-1}, t_n \rangle$ is an ordered sequence of real-valued observations taken at discrete times: $1, 2, \ldots, n-1, n$.

This is the standard definition of time series (e.g., Mueen et al. 2009). Our examples are all temporally ordered sequences but other orderings are possible.

Definition 2 A d-dimensional time series $T^d = \langle T_1, T_2, \ldots, T_d \rangle$ is a set of time series all associated with a single event and correlated in time.

Rather than examining only a single attribute as it varies in time, we assume that the event can be measured with a variety of attributes (d in this definition), each of which is measured on the same discrete time interval. These measurements are not required to be independent of one another. Our severe weather simulations have $d = 100$ but the independent dimensionality of the data is much less (approximately $d = 40$).

Definition 3 A labeled multi-dimensional time series is a tuple $E = \{T^d, l\}$ where T^d is a d-dimensional time series and $l \in \mathcal{L}$ where $\mathcal{L}$ is a discrete set of labels (and it is not required to be binary).

Because each of our d-dimensional time series is associated with an event, each example E_i is labeled. In the case of our severe weather data, $\mathcal{L}$ is either binary (positive/negative) or takes on one of three possible values: positive, negative, and intermediate. The approach does not restrict the possible numbers of labels but it does require that the cardinality of $\mathcal{L}$ be finite.

Our data set, $D = \langle E_1, E_2, \ldots, E_n \rangle$, consists of a set of labeled multi-dimensional time series, each of which can last for a variable amount of time but each of which is assumed to have the same dimensionality. That is, all attributes that measure an event are assumed to be present in each labeled example.

Definition 4 A single dimensional time series motif $M_j = \langle t_i, t_{i+1}, \ldots, t_{i+m} \rangle$ consists of a temporally ordered subsequence of a time series where $1 \leq i \leq n$ and $0 < m \leq n$. This motif is of length m and is on dimension j where $1 \leq j \leq d$.

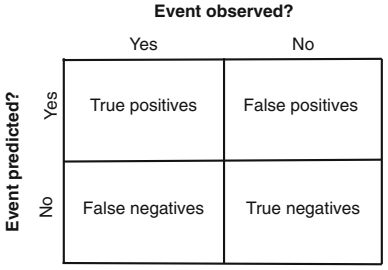
As stated above, our goal is to identify multi-dimensional times series motifs that can be used for prediction. As such, we build on the previous definition.

Definition 5 A multi-dimensional time series motif $P = \langle M_{i_1}, M_{i_2}, \ldots M_{i_m} \rangle$ is a temporally ordered set of single-dimensional time series motifs (see Definition 4). The temporal ordering specifies that the initiation of each single-dimensional time series M_{i_j} must begin after or simultaneously with the previous single-dimensional time series in the set $M_{i_{j-1}}$.

A multi-dimensional time series motif does not specify how many dimensions of the overall available dimensions must be used and it can even repeat dimensions, given that each is temporally ordered. For example, the first motif may be on dimension 1, the second on dimension 3, and the third on dimension 1 again. The temporal ordering is not strict as it requires that M_j begins after M_{j-1} but simultaneous initiations are also acceptable. M_j cannot begin before M_{j-1}. This definition differs slightly from the definition in Mueen et al. (2009), where there is a requirement for a temporal gap in between the different subsequences. We were interested in rules that could identify two features both firing at once and did not impose the temporal gap.

2.1 Performance measures

Our algorithm uses several standard statistical performance measures which we briefly review. Given binary labels on a data set, we can construct a two-by-two contingency table as shown below.



We measure three performance metrics on this table. The probability of detection (POD) is the number of times that an event was correctly predicted divided by the total number of observed events. POD ranges from 0 to 1 with 1 representing a perfect score.

$$POD = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{1}$$

The false alarm ratio (FAR) is the number of times that an event was incorrectly predicted to occur divided by the total number of events predicted to occur. FAR ranges from 0 to 1 with 0 representing a perfect score.

$$FAR = \frac{\text{false positives}}{\text{true positives} + \text{false positives}} \tag{2}$$

The critical success index (CSI, Donaldson et al. 1975; Schaefer 1990) evaluates success as a function of only the events that are predicted to be positive and the ones that were actually positive. Thus, CSI ignores the true negatives. This is particularly important for rare events such as tornadoes where it is easy to be an accurate forecaster simply by never forecasting a tornado. CSI focuses on the area of most interest in such situations. It ranges from 0 to 1 with 1 being a perfect score (perfect POD and perfect FAR). CSI incorporates both POD and FAR into one measure.

$$CSI = \frac{\text{true positives}}{\text{true positives} + \text{false positives} + \text{false negatives}} \tag{3}$$

2.2 Identifying the multi-dimensional time series motifs

A general outline of our approach for identifying multi-dimensional time series motifs is given in Table 1 and we describe each step in detail below. The general idea is to search for the critical dimensions of the data by identifying single dimensional motifs first, narrow down the set of possible single dimensional motifs using user specified minimum performance metrics and then grow the motifs across dimensions using the single dimensional motifs as building blocks for larger motifs.

Because a brute force approach to searching multi-dimensional data grows exponentially in the number of dimensions and the size of the data, it quickly becomes intractable. Additionally, searching for motifs in real-valued data is difficult, as has been noted by many researchers (for example, Das et al. 1998; Chiu et al. 2003; Mueen et al. 2009). We chose to address this latter problem using the SAX discretization technique (Lin et al. 2003) and we address the computational aspects using a combination of approximate search and intelligent data structures such as the trie described in Keogh et al. (2005).

The first step of our approach is to discretize each of the dimensions of data using SAX. SAX is a standard time series discretization technique which we describe in enough detail to understand the overall data mining algorithm but full details can be found in any of the SAX papers by Keogh et al (e.g., Lin et al. 2003, 2007;

Table 1 General approach to identifying multi-dimensional time series motifs

Grow multi-dimensional time series (D, alphabet size, word size, minimum POD, maximum FAR)
For each dimension d
Discretize E_i^d for all examples i using SAX(alphabet size, word size, averaging interval)
Build trie with pointers to start and end of each word in the sliding window
For each dimension d
Identify all single dimensional words with minimum POD and maximum FAR
Recursively grow longer rules within dimension d
For all rules that meet minimum POD and maximum FAR criteria
Grow rules across dimensions
Return list of rules sorted by CSI

Keogh et al. 2005; Shieh and Keogh 2009). SAX takes three main parameters as input and we explain each next. Our empirical results explore the effect of each parameter.

We use the following time series [1.68, 1.82, 0.28, −1.51, −1.92, −0.56] to illustrate the behavior of SAX. This example comes from six regularly spaced samples of the function $2\sin(x)$. SAX first normalizes the series to a mean of zero and standard deviation of one. For our example, this yields [1.09, 1.17, 0.20, −0.94, −1.19, −0.33]. The normalized time series is then aggregated using piece-wise aggregate approximation with a user specified parameter that we denote the *averaging interval*. This parameter specifies how many individual time series measurements should be aggregated/averaged together for the piecewise approximation. Larger intervals smooth over noise but they may miss critical structure in the data. For example, in our severe weather application, we experimented with averaging over one to five steps, which corresponds to 30 seconds to 2.5 minutes of storm time. In our simple example above, if the user specified an averaging interval of 2, it would aggregate the data by 2 steps and the time series would become [1.13, −0.37, −0.76].

Each of the piecewise aggregate numbers is then discretized using Gaussian breakpoints with the number of breakpoints specified by the user. This parameter is denoted the *alphabet size*. Keogh empirically states that this number should be relatively low in most cases, with the suggestion of three to four (Keogh et al. 2005; Xi et al. 2007). In our experiments, we varied the alphabet size from three to eight. Larger alphabet sizes provide higher resolution to the real-valued data but require more memory in the data structures and provided little benefit in our experiments. Using our example time series, an alphabet size of three would specify breakpoints at −0.43 and 0.43. This would turn our example time series into the discretized letters $[c, b, a]$.

The final parameter in the discretization is the *word size*, which specifies the length of the basic SAX words. A word is a sequence of discrete letters. Since the words are our starting point for identifying motifs, we keep the word length short so as not to miss short motifs. In our experiments, we varied the word length from two to three. The words are created using a sliding window. In our running example, a word size of two would yield the words $[c, b]$ and $[b, a]$.

In the first step of our rule discovery algorithm, we discretize each dimension of the data. To do this, we discretize all examples at once for each dimension. This ensures that the discretizations can be easily compared and that a in one time series has a similar meaning to a in another example.

Given the number of dimensions and examples, multiple passes through the data will be computationally expensive. To address this, we make use of the trie data structure as discussed in Keogh et al. (2005). Figure 1 shows an example of a trie on a short, single-dimensional time series with a three letter alphabet and two letter words. The trie forms a tree with the leaves forming all unique words. Each leaf stores a pointer into the individual time series when each word occurs. The two letter words are formed by moving a sliding window across the time series. For example, the word *aa* occurs twice in example series i and it has two pointers into the series for time step 1 and 6.

After discretizing the data using SAX, we build a trie for each dimension of the data. If the trie is fully populated, this requires $O(da^w)$ storage space for the leaves where d is the number of dimensions, a is the size of the alphabet, and w is the size of the

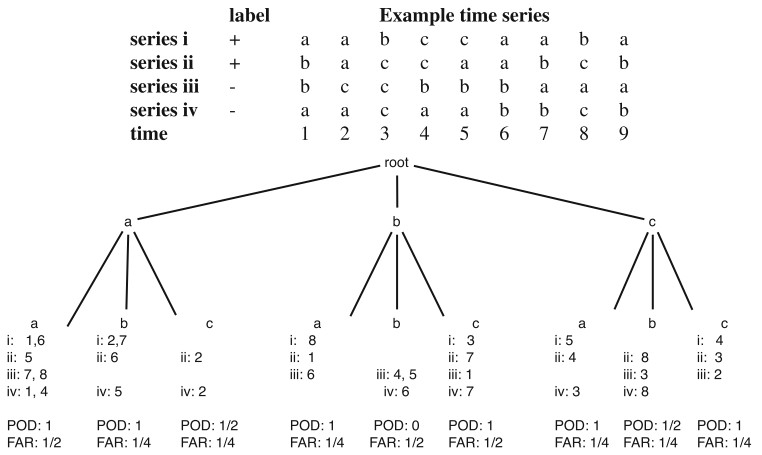


Fig. 1 Example of the trie data structure using a three letter alphabet and length two words on three example time series. Each series is labeled as positive or negative

words. Each leaf stores information on a single word. In most cases, the trie will not be fully populated as many words never occur in practice. In the worst case, an individual leaf would contain all examples from a time series, requiring t pointers where t is the length of the time series. However, if this occurred, the rest of the tree would be empty. We store a pointer for each occurrence even if they appear in order. Although this is more expensive in space, it yields O(1) access to where any word occurs.

Since each leaf/word has information about exactly which time series that word occurs in, the trie also stores the POD and FAR measures for use in the mining. These are computed directly from the definitions above. For example, in the time series data shown in Fig. 1, the word $[a, a]$ has a POD of 1 and a FAR of 0.5. The POD is 1 because $[a, a]$ occurs in both of the positively labeled series (i and ii) and has no false negatives. The FAR is 0.5 because the sequence does occur in both negatively labeled series (iii and iv) which gives it two false positives in addition to the two true positives.

Once the tries are built, the data mining algorithm makes use of them to efficiently narrow down the search for more complicated motifs. To do this, we look through each dimension of the data and narrow down the set of basic SAX words using user specified thresholds of the POD and FAR performance measures. By specifying a minimum POD and a maximum FAR, we limit the number of elementary motifs identified. These numbers can come from a user's experience in the domain. This significantly improves the running time of the search since all possible combinations of small motifs are used to grow the larger motfis. Since the search proceeds from general motifs (e.g. short ones) to specific motifs (longer multi-dimensional motifs), the performance of the motifs will only improve POD and FAR. This is similar to the pruning search discussed in Webb (1995), Oates and Cohen (1996), and McGovern and Jensen (2008) where the type of search and evaluation measures can be combined to enable admissible pruning. Thus, we specify minimum levels of performance that would be acceptable and expect that the final numbers will be significantly better with the more

specific rules. In the example shown in Fig. 1, a minimum POD of 0.5 and a maximum FAR of 0.4 would narrow the initial words to $[a, b]$, $[a, c]$, $[b, a]$, $[c, a]$, $[c, b]$, $[c, c]$. Any words with $[a, a]$, $[b, b]$, $or[b, c]$ would be removed from the search because they do not satisfy the minimum POD and maximum FAR requirements.

Once the basic words are identified, we grow the motifs recursively using the basic words that pass the POD/FAR thresholds. The recursive growth of the motifs within a dimension works by doubling each motif, similar to the method employed by SPADE (Zaki 2001). Each motif is doubled while maintaining the minimum POD and maximum FAR requirements. For example, $[a, b]$ can be doubled to $[a, b, a, b]$ and it can also be combined with other valid words such as $[a, b, b, a]$. When a doubling fails, the motif is grown linearly by adding words until it is at its maximum length. Since motifs are grown in chunks of word size, not all possible motifs can be detected. For example, the motif $[a, b, a]$ is not discoverable in the example because the minimum word size is 2. However, in all of our experiments, we keep the word size small to minimize this issue.

The motif growing within a single dimension is repeated for all dimensions before searching across the dimensions. Although this growing sounds computationally expensive, only one pass on the original data is required. Using the trie data structure, all further motifs can be grown by examining the starting and ending times of each individual motif and ensuring that they follow one another temporally (e.g. they satisfy Definition 5 above). The POD and FAR measures continue to be directly computed from the trie by intersecting the positive and negative graphs computed for each individual piece of the motif.

The last step of the algorithm is to repeat the search by combining the different words across the dimensions of the data. Once all of the motifs have been identified for each dimension, an exhaustive search of temporal orderings across dimensions is performed. Although doing an exhaustive search sounds infeasible, it is possible because of the admissible pruning afforded by the user's minimum POD and maximum FAR measures. In addition, the O(1) access time of the trie facilitates the overall approach. Without the pruning, this approach would be untenable but the pruning significantly improves the running time by enabling the search to ignore large portions of the search space while guaranteeing that the rules that could be identified in that space will never meet the user's specified performance measures. In addition, the trie again can be used to quickly compute the POD/FAR measures across dimensions by continuing to intersect the positive and negative series observed by each piece of the motif. We do not empirically evaluate the performance of the pruning because our initial experiments indicated that the non-pruning approaches would literally take weeks to months of CPU time to complete.

Once the rules are identified, the probability of the event we are looking for can be calculated in the same manner as in a probability tree. To calculate the probability of event A occurring, you divide the number of times event A did occur in the set of all examples that match that rule by the total number of all examples that match the rule. This probability could be scaled using a LaPlace correction if necessary, although we did not employ that in this work (Provost and Domingos 2003).

In the final step, we sort the list of rules by the CSI score. CSI combines POD and FAR and enables us to sort the rules on a single scalar value.

3 Empirical evaluation

Although our work is motivated by severe weather prediction, we believe this approach has general applicability to multi-dimensional time series data. To verify this, we validated the approach on the CMU Motion Capture Database,[1] which is a well-known multi-dimensional temporal data set. We first examine the results in this domain and then focus on the severe weather domain.

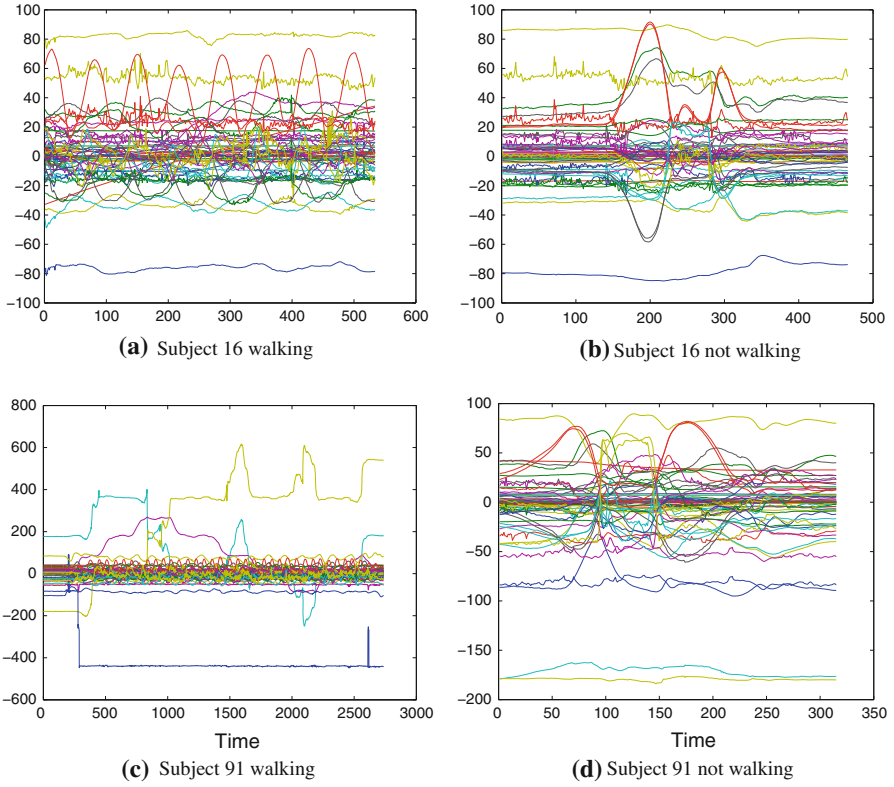
3.1 Motion Capture

The CMU Motion Capture database contains multi-dimensional time series data obtained by measuring the positions of a set of joints from humans performing a series of tasks. As an illustration of our algorithm, we examined the task of predicting whether a subject was walking or not. Examples with walking were labeled as positive and examples of other activities were labeled as negative. Subjects 16 and 91 each had sufficient examples of walking versus similar activities. Subject 16 focused primarily on running versus walking and subject 91 focused on walking in funny ways versus jumping. Although there are many other examples of subjects performing other activities in the CMU MOCAP database, our focus for this data is demonstrating the generality of the approach rather than a full analysis of the database. As such, we focus on only these two subjects. They were chosen because each provided sufficient examples for cross-validation on the task. Subject 16 has 58 examples and subject 91 has 62 examples. Each has a roughly 50/50 split between walking and other activities.

Figure 2 shows examples of the time series data for subject 16 and 91. All of the MOCAP data has 62 different joint positions recorded for a varying amount of time. Panels a and c show examples of the subjects walking and panels b and d show examples of the subjects running. Although some of the movements of the same joints are obviously different across the two sets of time series, many of them are less obvious. In addition to identifying the temporal patterns that are relevant, our approach is able to identify which of the 62 different joints is critical to distinguishing the two types of movements.

For the experiments with both subject 16 and subject 91, we varied the three main parameters to discretization across a full set of reasonable values. The alphabet size, which is the number of discretized entities, varied from 3 to 8. Keogh et al. (2005) and Xi et al. (2007) stated that the best number of letters falls in the range of 3 to 4 and we increased the resolution of the alphabet to 8 to ensure we were not losing critical detail. We varied the word size, which is the number of discretized letters in a single word, from 2 to 3. Last, we varied the number of time steps averaged into each piecewise aggregation from 1 to 5. This yielded 60 different parameter variations for both subject 16 and subject 91. For all sets of parameter settings, we trained on the full set of data and also used 10-fold cross validation. The results are reported for both the training and the test sets but we chose to train on both to assure that there was a minimum of overfitting. For all of the experiments with the MOCAP data, we chose

[1] http://mocap.cs.cmu.edu/.



(a) Subject 16 walking **(b)** Subject 16 not walking

(c) Subject 91 walking **(d)** Subject 91 not walking

Fig. 2 Example of subject 16/91 **a,c** walking and **b,d** not walking. In both cases, all 62 recorded joint positions are shown

a minimum POD of 0.8 and a maximum FAR of 0.4. These numbers were chosen empirically to maintain a quick running time.

Table 2 shows the top 10 parameter variations sorted by the test set CSI score for subjects 16 and 91. There are several striking aspects to these tables. First, for subject 16, the top 4 results are all perfect predictors on both the training and the test data. For subject 91, the rules are very accurate, missing only one example in each of the top 10 rules. Second, the parameter variations span the entire range of possible variations, indicating that the results are robust across a wide range of parameters. All parameter variations have a CSI score above 0.87 for subject 16 and 0.93 for subject 91, indicating that all parameter variations produce strong results.

To confirm that none of the parameters had a major impact on the CSI score, we performed a multi-way ANOVA analysis. The results for both subject 16 and subject 91 are shown in Table 3. As expected, none of the results are significant for $p \leq 0.01$.

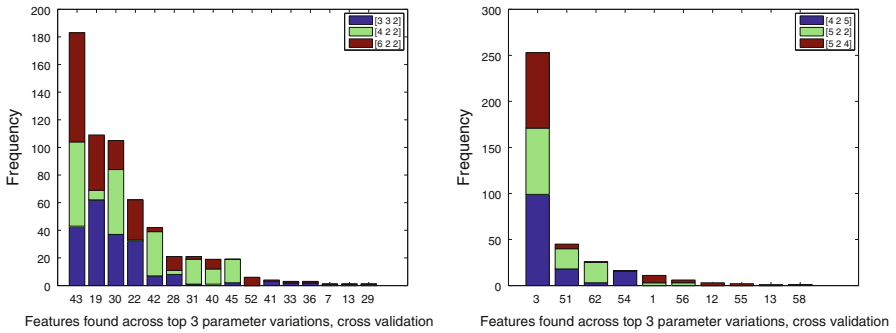
One of the key characteristics of our approach is that it can identify which parameters are most useful for prediction from a large multi-variate time stream. Figure 3a shows the features, i.e. joints, chosen for the top 10 rules over all 10 folds of 10-fold cross validation for the top 3 performing parameter variations for subject 16. Figure 3b

Table 2 Top 10 parameters for subjects (a) 16 and (b) 91. For each parameter variation, we measured the training set POD, FAR, and CSI as well as the 10-fold cross validation average of POD, FAR, and CSI. Parameter variations are ranked by 10-fold cross validation CSI score

Alphabet size	Word size	Averaging interval	Training data			10-Fold cross validation			
			POD	FAR	CSI	POD	FAR	CSI	Rank
(a) Subject 16									
3	3	2	1.000	0.000	1.000	1.000	0.000	1.000	1
4	2	2	1.000	0.000	1.000	1.000	0.000	1.000	1
6	2	2	1.000	0.000	1.000	1.000	0.000	1.000	1
7	3	2	1.000	0.000	1.000	1.000	0.000	1.000	1
6	3	1	1.000	0.000	1.000	1.000	0.025	0.975	5
7	3	5	1.000	0.000	1.000	1.000	0.025	0.975	5
6	3	3	1.000	0.000	1.000	0.967	0.000	0.967	7
3	2	1	1.000	0.000	1.000	1.000	0.058	0.942	8
5	3	4	1.000	0.000	1.000	0.967	0.033	0.933	9
6	2	1	1.000	0.000	1.000	1.000	0.067	0.933	9
6	3	5	1.000	0.000	1.000	0.933	0.000	0.933	9
(b) Subject 91									
4	2	5	1.000	0.000	1.000	1.000	0.020	0.980	1
5	2	2	1.000	0.000	1.000	0.975	0.000	0.975	2
5	2	4	0.974	0.000	0.974	0.975	0.000	0.975	2
5	3	1	1.000	0.000	1.000	0.975	0.000	0.975	2
5	3	2	1.000	0.000	1.000	0.975	0.000	0.975	2
5	3	3	1.000	0.000	1.000	0.975	0.000	0.975	2
6	2	1	0.974	0.000	0.974	0.975	0.000	0.975	2
6	2	3	1.000	0.000	1.000	0.975	0.000	0.975	2
6	3	3	1.000	0.000	1.000	0.975	0.000	0.975	2
6	3	4	1.000	0.000	1.000	0.975	0.000	0.975	2
7	3	5	1.000	0.000	1.000	0.975	0.000	0.975	2
8	2	1	1.000	0.000	1.000	0.975	0.000	0.975	2

Table 3 ANOVA P-Values for significance of parameter effects on alphabet size and averaging interval

Factor	Subject 16	Subject 91
Alphabet size	0.11	0.09
Averaging interval	0.15	0.48
Word size	0.38	0.19
Alphabet × word size	0.05	0.99
Alphabet × averaging interval	0.06	0.93
Word size × averaging interval	0.54	0.63



(a) Features chosen for subject 16 **(b)** Features chosen for subject 91

Fig. 3 Histograms of the frequency that each joint was chosen in a rule in 10-fold cross validation for the top 3 parameter variations for **a** subject 16 and **b** subject 91. Parameter variations are color coded with bracketed numbers indicating alphabet size, word size and averaging interval respectively

repeats this same histogram for subject 91. Of particular note is the consistency in the features chosen across the parameter variations and rules. For the top 3 parameters for subject 16, only 16 features out of 62 were chosen and only 5 of those were chosen very frequently, leading us to believe that these features are very predictive. Likewise, only 10 features were chosen for subject 91 and the first one clearly dominates the choices. Without the kinematic knowledge of what the different features are, we instead examine the time series for the features and the rules identified.

Figures 4 and 5 show an example of rules found in 10-fold cross validation for the top parameter settings for subject 16 and 91 respectively. Although SAX normalizes each dimension to a zero mean, standard deviation of 1, we have shown the rule in the original data by undoing the normalization. This enables the domain scientists to read the rule as it appears in the original data. In the case of Fig. 4, the alphabet size was 3, the word size was 3, and each letter represented two time steps. The rule states that feature 19 must begin a downward trend (ccb) followed by feature 43 staying low for 6 steps in a row. Features 43 and 19 are the top two chosen features for subject 16, as shown in Fig. 3a. Another interesting feature of this rule is just how short it is while being perfectly predictable. We expected the rules would be more complicated but the search approach identifies the shortest rules that are predictive, in accordance with Occam's razor. In this domain, it seems that short rules are powerful.

Figure 5 shows a rule for subject 91 for the two most frequently selected features. In this case, those features are 3 and 51 (see Fig. 3b). For this rule, the behavior is to observe feature 3 maintaining a low value for 10 time steps (recall that this rule uses an averaging interval of 5) followed by feature 51 also maintaining a low value for 10 time steps. Examination of the other rules identified within this parameter setting show variations on this same theme.

Given the simplicity of the rules yet their clearly powerful predictive nature, we examined the time series for the top features for both subjects. Figure 6 shows all of the positive and negative examples for the top feature for both subject 16 and subject 91. In both cases, the positive examples are shown in blue and the negatives in red. Looking first at panel a, which is subject 16's data, it is clear that the behavior of the

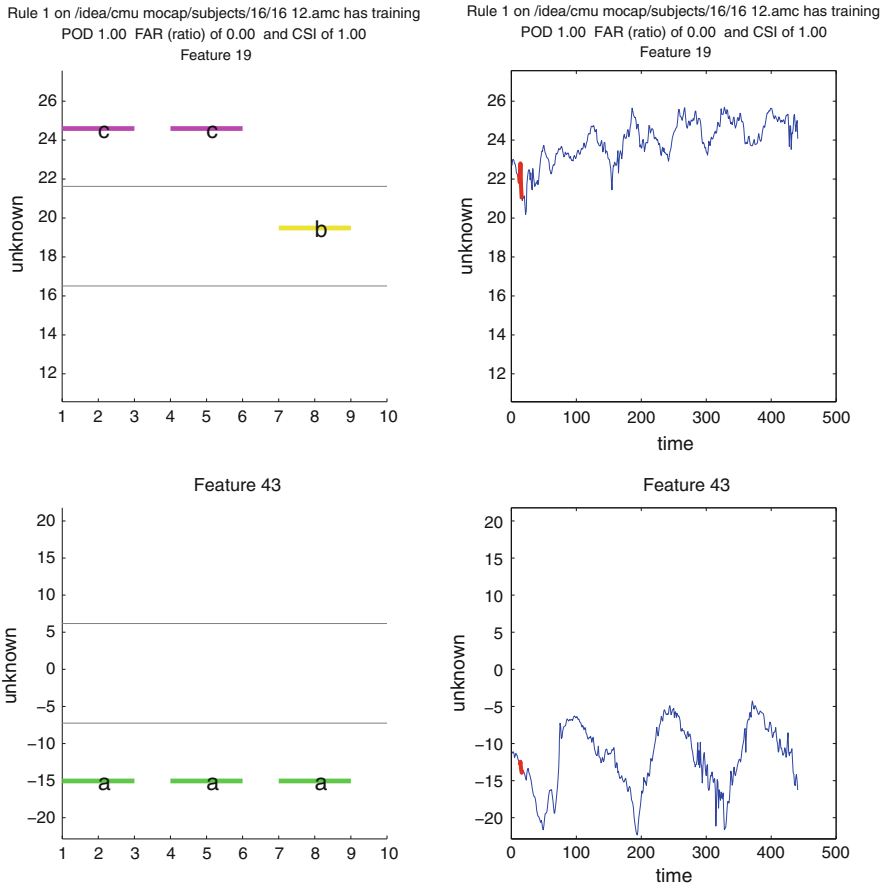


Fig. 4 Example rule from subject 16. Each row in the plot shows a word from the rule and the temporal ordering of the rule goes from the top to the bottom. The left side of each row shows the SAX word used in the rule and the right side shows an example of the rule being applied to a time series. This rule specifies that feature 19 has to drop from a high value to an average value before feature 43 stays at a low value. This rule used an alphabet size of 3, a word size of 3, and averaged over 2 time steps per letter. Since the rules are sorted based on the training set performance (within cross-validation), those measures are listed for the rule

positive and negative examples is significantly different. The majority of the negative cases have all of their values above the positive cases. Even in the cases that overlap, the highest value in the time series is generally above the mean of the positive cases. Panel b for subject 91 is even more striking. In the positive case, all of the data for feature 3 is sinusoidal. For the negative cases the amplitude is significantly lower and the time series is shorter. The rules that used feature 3 specify that the value of feature 3 should drop below a specified threshold and then stay low (e.g. Fig. 5). This is visually confirmed to be a powerful rule by this figure.

In contrast to the data shown in Fig. 6, the data plotted in Fig. 7 is for a feature never selected by any of the top 20 parameter variations. For both subject 16 and subject 91, that is feature 4. In both cases, the behavior of the positive versus negative time

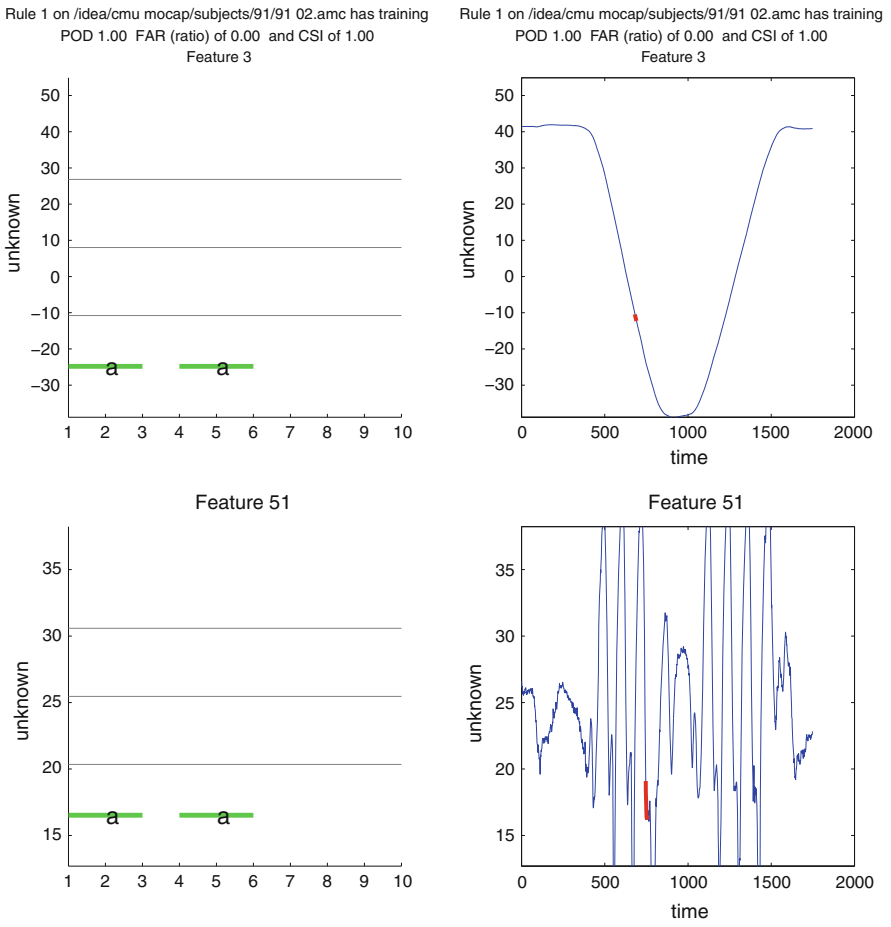


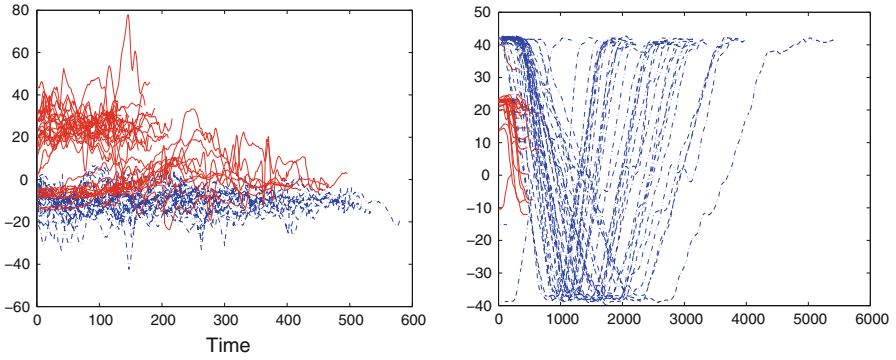
Fig. 5 Example rule from subject 91 in the same style as Fig. 4. This rule used an alphabet size of 4, a word size of 2, and averaged over 5 time steps per letter

series is difficult to differentiate with the exception of the duration of time in subject 91.

The application to the CMU Motion Capture database demonstrated that we could efficiently identify predictive rules in a large multi-dimensional data set with no domain knowledge about the data. Our algorithm identified the joints that are used primarily for walking and demonstrated that these joints are consistent across multiple parameterizations. In addition, the predictive performance of the rules was very strong.

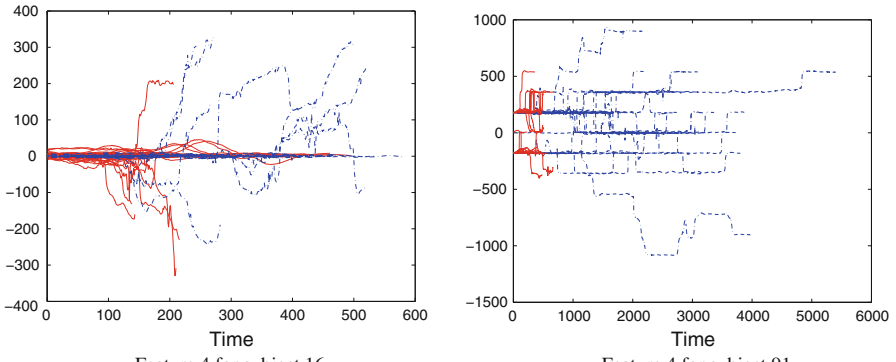
3.2 Severe weather

Rosendahl (2008) created a set of 263 simulations of supercell thunderstorms, which are the most severe type of thunderstorm and which generate the most violent torna-



Feature 43 for subject 16 Feature 3 for subject 91

Fig. 6 The positive (*blue dashed*) and negative (*red solid*) data for the top **a** feature 43 for subject 16 and **b** feature 3 for subject 91



Feature 4 for subject 16 Feature 4 for subject 91

Fig. 7 The positive (*blue dashed*) and negative (*red solid*) data for a feature never selected by the rules for subject 16 and subject 91. Feature 4 is used for both subjects

does. Each simulation was generated using the Advanced Regional Prediction System (ARPS) (Xue et al. 2000, 2001, 2003). The full details on the parameters chosen to create the storms are described in Rosendahl (2008). Tornadoes are such rare events that a data set of observed storms would likely contain only a few tornadoes, making training, testing, and validation difficult. In addition, such a data set would be missing many of the fundamental meteorological variables since they are not observed by current sensing platforms. Therefore, we use numerical simulations to generate our data. Simulations have been used successfully to study severe weather including tornadoes (for example, see Hu et al. 2004; Adlerman and Droegemeier 2005). In this paper, we focus on the results of the data mining algorithm particularly as a function of the parameters to the algorithm while Rosendahl et al. (in preparation) analyze the meteorological results identified by our rules.

3.2.1 Data extraction

Each simulation is run for 3 hours of storm time. The simulation saves the state of all relevant meteorological variables every 30 seconds of storm time for every grid point in the domain. With a resolution of 500 m horizontally and a domain size of 100 km, a stretched vertical resolution focusing on the lower altitudes (with 50 voxels vertically), the simulations must save over 100 different variables every 30 seconds for each of 2 million grid squares. In total, each simulation produces over 21 GB of data, which requires us to intelligently process and mine this data.

Although each simulation generates a full gridded field of meteorological variables, the variables near a storm cell will provide the most information. We identify and track storm cells using a modified form of the Storm Cell Identification and Tracking algorithm (Johnson et al. 1998; McGovern et al. 2007) where we track the cells based on their dominant updraft region (localized area with rising air) because it is the defining feature of a supercell storm. Figure 8 shows an example of simulated radar reflectivity 75 min into a simulation. Reflectivity measures the intensity of the precipitation within the storm which means that regions with high rain, snow, ice, or hail have a higher reflectivity value. The black outlines in Fig. 8 show the two storm regions that are being tracked during that period. Because weak short-lived storms are not of interest in this study, we only track cells that last for at least 30 min. Each simulation typically produces 3–4 such cells.

Each storm cell defines a three dimensional region of interest for mining. To create the metadata for mining, we extract maximum and minimum values of relevant meteorological quantities within each of these storm cells. We measure the maximum and minimum value for each variable from the surface to 2 km in height and then from 2 to 8 km. For some variables, we also store the maximum and minimum values at the surface. This allows us to identify whether a maximum or minimum value is associated with a surface, low, or mid to upper altitude feature. This yields a 100 dimensional time series for each storm. The full set of quantities is defined in Rosendahl (2008) and Rosendahl et al. (in preparation).

We extract the maximum and minimum values every 30 seconds for the entire 3 hours of simulation. Figure 9 shows an example of several time series extracted for two of the meteorological quantities. The left panel shows the evolution of the vertical vorticity (an instantaneous measure of spin about a vertical axis) at the surface, low altitudes, and mid- to upper-altitudes for the center storm shown in Fig. 8. Maximum and minimum values in the left panel correspond to counterclockwise and clockwise rotation respectively. The right panel shows the maximum horizontal wind speed values.

The 263 storm simulations generated 1168 separate storm cells that each lasted at least 30 min. Given the sheer number of storm cells, we developed an automated labeling approach based on the key characteristics of tornadic storms. Because the horizontal grid spacing of the simulations is too coarse to detect rotation on the scale of a tornado and creating higher resolution simulations requires exponentially more computational time and space, we labeled each storm as to whether it produced strong low-altitude rotation (“positive”), produced either no or very weak rotation (“negative”) or was in between these two categories (“intermediate”). Strong low-altitude

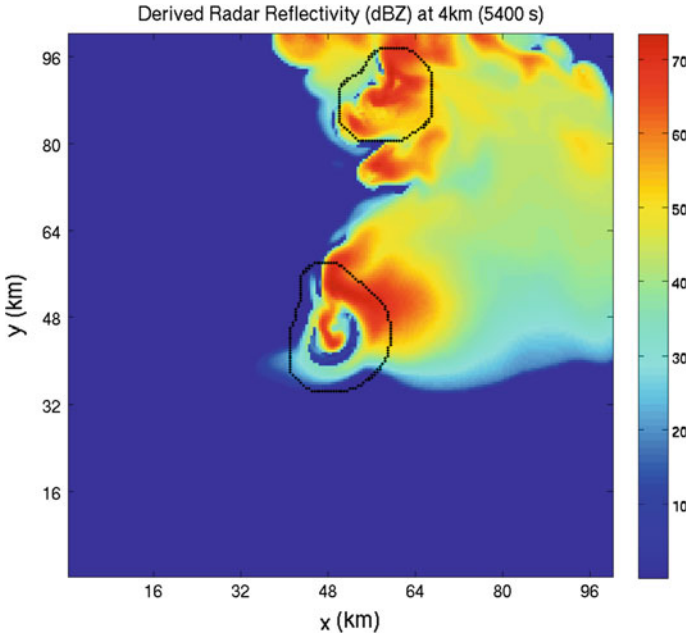
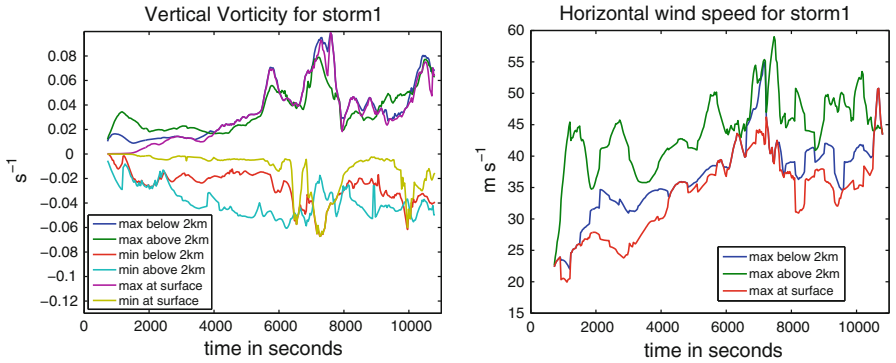


Fig. 8 Reflectivity of an example numerical storm simulation 75 min into the storm's lifetime. The scale on the right shows the intensity of the reflectivity in dBZ. Higher reflectivity regions indicate areas where the storm is producing intense precipitation. The black outlines highlight individual storm cells which are used to extract the storm metadata



(a) Example vertical vorticity data **(b)** Example horizontal wind speed data

Fig. 9 Meteorological quantities extracted from an example storm. Shown are the maximum and minimum quantities for vertical vorticity (*left*) and the horizontal wind speed (*right*). The full set of 100 extracted quantities for this storm can be found in Rosendahl (2008)

rotation was defined as a storm where there was a decrease in the surface pressure perturbation of at least −900 Pa in 1000 seconds and either an increase in the horizontal wind speed at the surface of at least 5 m s^{-1} within 750 seconds or an increase in the absolute value of the vertical vorticity of at least 0.03 s^{-1} within 500 seconds. These features had to overlap within a 600 seconds window to ensure that they were

correlated. Storms where the pressure drop fell within the range of −900 Pa to −300 Pa and met the vertical vorticity and wind speed criteria or that had a pressure drop but no corresponding increase in vertical vorticity or wind speed were labeled as intermediate storms. The remainder of the storms were labeled as negative storms. This yielded 58 positive storms, 373 intermediate storms, and 737 negatives.

Given the labeled data, we further processed it in two ways. First, if we were to feed all of the time series information to the data mining algorithm, it would identify the approach that we took to label the data. To avoid rules that simply state that pressure perturbations or vertical vorticity are critical, we remove each of the features used to label the data from consideration. Further, to ensure that identified precursors were associated with the developing strong low-altitude rotation in positive storms, we saved data for 30 min immediately prior to the beginning of the corresponding pressure drop, which was defined using a Gaussian derivative filter on the pressure perturbation time series. Because there is no definable pressure drop for the intermediate and negative storms, we randomly sampled these storms to 30 min as well to avoid obvious labeling based on the length of the time series given to the data mining algorithm.

3.2.2 Severe weather results

In this paper, we focus on a full sensitivity analysis of the three main parameters (alphabet size, word size, averaging interval) for the severe weather simulations while in Rosendahl et al. (in preparation), we focus on the meteorological implications of the results. To ensure that the results were consistent across sets of storms, we varied the training data from positives versus negatives to positives versus intermediates. For each set of training data, we also varied the parameters to the data mining algorithm. We explored all feasible variations on alphabet sizes from three to eight, word sizes from two to three, and averaging intervals from one to five. There are 60 combinations of these parameters but some of the combinations with alphabet sizes of 7 or 8 could only use a word size of 2. This was a limitation on file sizes in the file system, rather than any inherent limitation in the approach itself. Within each set of parameters, we explored the consistency of the results and possible effects of overfitting by training on both the entire data set and a 10-fold cross validation data set. For the results using the severe weather data, we used a minimum POD of 0.7 and a maximum FAR of 0.8. These numbers were chosen empirically to maintain a quick running time.

Table 4 shows the top ten parameter variations for the positives versus negatives data set. This table is ranked by the 10-fold cross-validation CSI score. The table highlights several interesting results. First, the difference between the training set CSI and the 10-fold cross-validated CSI scores is minimal, which is evidence that we are minimizing overfitting. Second, and of particular note to the sensitivity analysis of the parameters, there is no apparent pattern to the discretization parameters and the CSI score. For example, the top 10 parameter variations contain every alphabet size except 7, both word sizes and all averaging intervals. This is evidence that the results are insensitive to the parameter settings on the discretization.

We continued the sensitivity analysis on the positive versus negatives data in several different manners. There is no discernible pattern in the parameters with the rank or CSI score. To verify this statistically, we also performed a multi-way ANOVA test. The

Table 4 Top 10 parameters for discretizing the data in the positives versus negatives data. For each parameter variation, we measured the training set POD, FAR, and CSI. Parameter variations are ranked by 10-fold cross validation CSI score

Alphabet size	Word size	Averaging interval	Training data			10-Fold cross validation			
			POD	FAR	CSI	POD	FAR	CSI	Rank
5	3	1	0.827	0.123	0.740	0.823	0.128	0.704	1
8	2	4	0.817	0.056	0.779	0.756	0.117	0.693	2
4	2	3	0.821	0.146	0.720	0.776	0.147	0.688	3
5	2	1	0.865	0.153	0.748	0.823	0.190	0.684	4
6	2	1	0.827	0.099	0.758	0.796	0.184	0.684	5
6	3	1	0.829	0.103	0.757	0.753	0.126	0.683	6
5	2	3	0.818	0.118	0.737	0.726	0.154	0.667	7
5	2	2	0.827	0.119	0.743	0.773	0.181	0.665	8
6	3	3	0.821	0.132	0.729	0.783	0.214	0.654	9
3	2	5	0.825	0.164	0.710	0.790	0.220	0.652	10

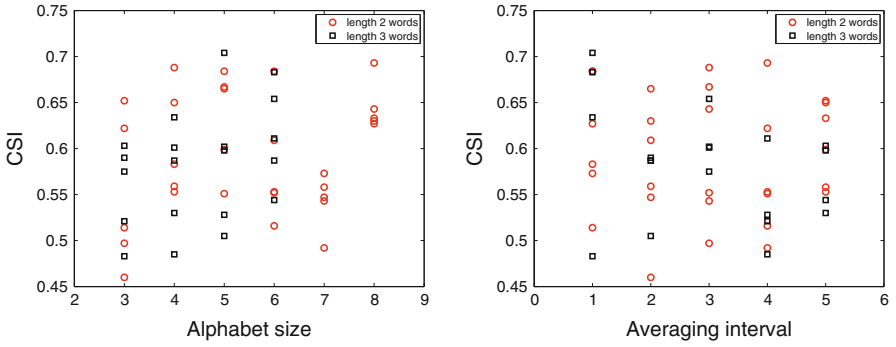
Table 5 ANOVA P-Values for significance of parameter effects on alphabet size and averaging interval

Factor	Positive versus negative	Positive versus intermediate
Alphabet size	0.21	0.03
Averaging interval	0.15	0.33
Alphabet × averaging	0.77	0.34

results of this test are shown in Table 5. For the positives versus negatives experiments, none of the parameters or cross-products of parameters is significant at the $p \leq 0.01$ level.

Due to the limitations on file sizes that we noted above, we could not complete all 60 of the parameter variations necessary for the full three-way ANOVA over all of the parameters. As Table 5 shows, we performed the analysis without word size since it was the parameter not fully represented in the variations. To examine the effect (or lack thereof) of word size, we plotted the CSI score as a function of alphabet size and averaging interval. These are shown in Fig. 10. Although we cannot test it statistically, a visual examination of the results indicates that the word size parameter does not seem to have an effect.

We repeated this experiment with the positives versus intermediates data. We had two goals in mind for the second set of experiments. First, the skill level in the positives versus the negatives data was much higher than expected. We determined that part of the reason was that the storms had greater differences in their dynamical structure. By definition, the labeling algorithm searched for extreme pressure drops to label as positive while negative storms had little to no discernible pressure drop. By training on the positives versus the intermediates, we hoped to train on a more realistic set



(a) CSI by alphabet size **(b)** CSI by averaging interval

Fig. 10 CSI as a function of *a* alphabet size and *b* averaging interval and colored/shaped by word size for the positives versus negatives data

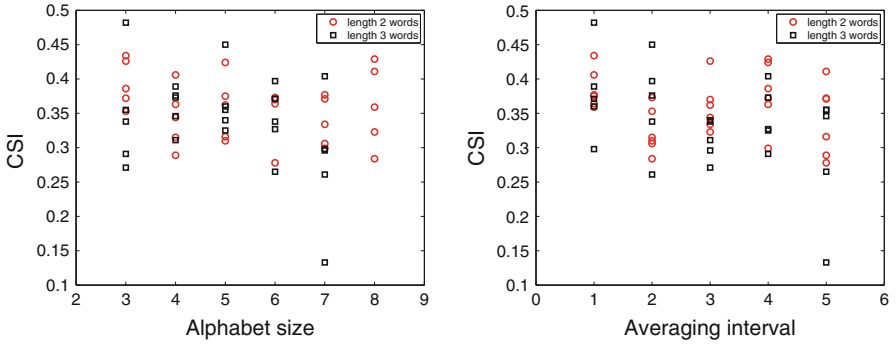
Table 6 Top 10 parameter variations for the positives versus intermediates data. Parameter variations are ranked by 10-fold cross validation CSI score

Alphabet size	Word size	Averaging interval	Training data			10-Fold cross validation			
			POD	FAR	CSI	POD	FAR	CSI	Rank
3	3	1	0.749	0.398	0.500	0.740	0.424	0.482	1
5	3	2	0.726	0.324	0.538	0.603	0.401	0.450	2
3	2	1	0.808	0.448	0.487	0.730	0.489	0.434	3
8	2	4	0.793	0.557	0.396	0.800	0.524	0.429	4
3	2	3	0.768	0.415	0.496	0.686	0.470	0.426	5
5	2	4	0.729	0.418	0.477	0.680	0.464	0.424	6
8	2	5	0.741	0.537	0.398	0.750	0.515	0.411	7
4	2	1	0.783	0.413	0.503	0.683	0.499	0.406	8
7	3	4	0.724	0.533	0.396	0.723	0.511	0.404	9
6	3	2	0.752	0.454	0.461	0.660	0.512	0.397	10

of storms where the dividing line between severe and non-severe was less clear (i.e., storms had more structural similarities). Second, we hoped that the results would be consistent across the two types of training data.

Table 6 shows the top 10 parameter variations for the positives versus the intermediates data set. The most obvious difference between these results and those shown in Table 4 is that the best 10-fold cross validation CSI is only 0.48 here while it was 0.70 when trained on the positives versus negatives data. As we hypothesized, the difference between the positives versus intermediate was less obvious and the prediction skill decreased correspondingly although 0.48 still indicates strong predictive capabilities.

We repeated the ANOVA with the positives versus intermediates data and, again, none of the parameters shows a statistically significant effect at $p \leq 0.01$. Figure 11 shows the CSI as a function of alphabet size and averaging interval. As with the



(a) CSI by alphabet size **(b)** CSI by averaging interval

Fig. 11 CSI as a function of *a* alphabet size and *b* averaging interval and colored/shaped by word size for the positives versus intermediates data

positives versus negatives data, there is no clear visual effect of the word size other than the missing runs for alphabet size of 8.

Our second goal with using the two different data sets was to verify that the results of the algorithm were consistent across the two data sets. We hypothesized that the features that showed up as important in the positives versus negatives data would be very similar to the positives versus intermediates, at least for the top performing parameter variations. As the performance decreased, we expected the correlation between the two data sets to decrease.

We examined this hypothesis in two ways. Figure 12 shows a histogram of the probability of each feature being chosen in the top 10 rules for the top three parameter variations of the two data sets. Once the features used for labeling were removed, there were 88 total features for feature selection to choose from. One of the first things to note from this graph is that only 17 of the 88 features were ever selected within the top 10 rules of these top three parameter variations. This is a strong argument in favor of the agreement between the two data sets. The second striking feature of this graph is that the relative probabilities of each feature being selected are very similar across the two data sets. Both of these patterns are true across the top 10 and top 20 parameter variations (not shown).

The second way in which we examined this hypothesis was to look at the choice of a feature within the top 10 rules as binary (it was selected or it was not selected). We then measured the percent agreement between the two data sets by measuring the number of times that feature x was selected by both or not selected by both. Figure 13 shows this measurement as a function of the top k parameter variations where k varied from 1 to 20. The graph shows the percent agreement for the 10-fold cross validation data as well as the training data. The training data agrees over 90% of the time across both data sets. The cross-validation agreement starts high and decreases as we increase the number of parameter settings being compared. This was to be expected as the performance difference of the two approaches grows as the lower performing parameter settings are chosen.

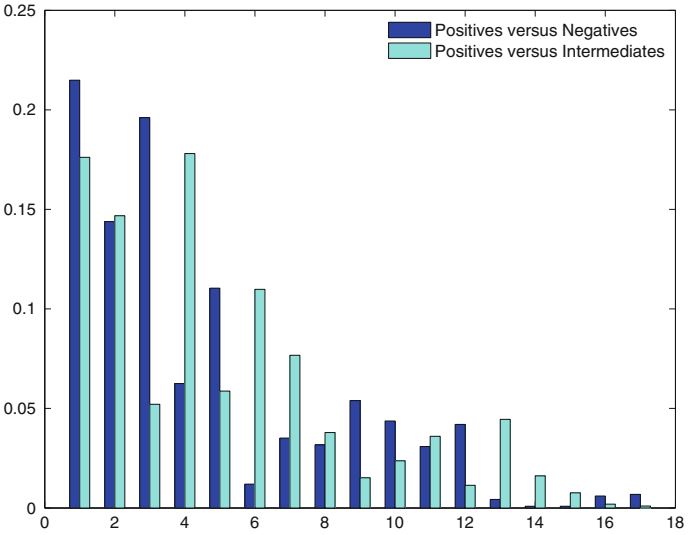


Fig. 12 Probability of a feature being chosen in the top 10 rules of the top 3 parameter variations for the positives versus negatives data and for the positives versus intermediates data

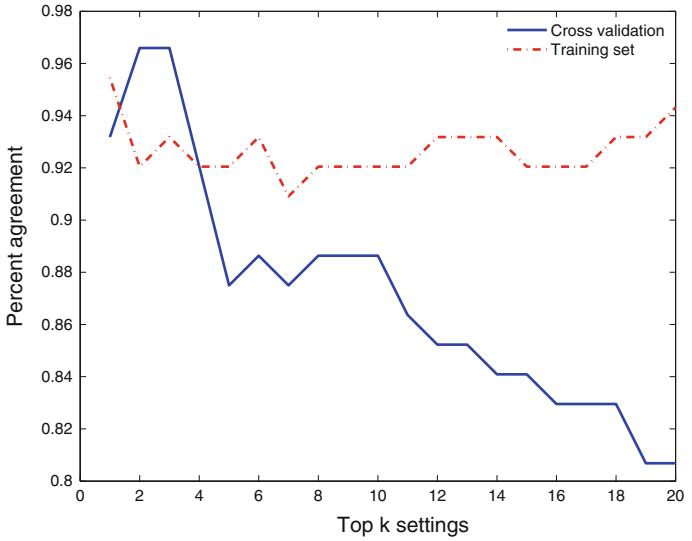


Fig. 13 Agreement of the feature selection choices across the positives versus negatives data and the positives versus intermediates data for both 10-fold cross validation and the training data only

To create the results shown in Fig. 13, we created a 2×2 contingency table where the main diagonal indicated agreement of the two data sets (choosing or not choosing a feature) and the off-diagonal indicated disagreement. We then examined the χ^2 values of each table for each data point shown in the graph. Each was statistically significant with the highest p value being 9×10^{-9}.

4 Discussion and future work

We have introduced a novel approach that efficiently identifies multi-dimensional temporal motifs in large data sets. This approach makes use of efficient data structures including the trie (Keogh et al. 2005) to minimize the number of passes through the original data. In addition, the ability to perform admissible pruning significantly reduces the search space and maintains the user's specified performance measures in the final rules.

We validated our temporal motif discovery approach on two real-world data sets. In the CMU Motion Capture database, we identified rules to distinguish walking from other similar behavior such as running using multi-dimensional data obtained through analysis of videos. These rules also identified the most important joints for distinguishing walking from the other behaviors. These rules had a very strong predictive power.

In the real-world meteorological data, we focused on the sensitivity analysis where we demonstrated that the results were robust across a wide range of parameter settings. In Rosendahl et al (in preparation), we perform a full meteorological analysis of our results on the severe weather simulations. In this latter paper, we validate that this approach identified sets of precursors, in the form of meteorological quantities reaching extreme values in a particular temporal sequence, unique to storms producing strong low-altitude rotation. These quantities are consistent with current meteorological theories on the formation of such rotation.

Part of the future work on our method includes verifying that our SAX based approach does not suffer from the wandering baseline effect sometimes observed in real data. Kasetty et al. (2008) demonstrate that their SAX-based approach to mining time series data does not have this issue. Other future work includes applying it to domains with an even wider variety of sampling rates in order to fully observe the effects of the parameters.

In current work, we are developing spatiotemporal relational models (e.g. McGovern et al. 2008; Supinie et al. 2009; McGovern et al. 2010) and applying these models to severe weather data. These models address both the spatial and the spatiotemporal changes in the data using a relational approach. For example, we can examine the change in the relationships between various storm objects over time. The work presented here provides the foundation for our current spatiotemporal relational work. In addition, we are developing a set of high resolution simulations capable of resolving tornadoes themselves. These simulations will be the foundation of our future studies on tornado development. We are also validating our models on several assimilated cases of observed tornadic storms.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant No. REU/0453545, IIS/REU/0755462, IIS/CAREER/0746816 and corresponding REU Supplements IIS/0840956 and IIS/0938138, the NSF ERC Center for Collaborative Adaptive Sensing of the Atmosphere (CASA, NSF ERC 0313747), and the University of Oklahoma's College of Engineering. We would also like to thank Nathan Hiers, Adrianna Kruger, and Meredith Beaton for their preliminary work on this data. The motion capture data used in this project was obtained from mocap.cs.cmu.edu and their database was created with funding from NSF EIA-0196217.

References

Adlerman E, Droegemeier KK (2005) The dependence of numerically simulated cyclic esocyclogenesis upon environmental vertical wind shear. Mon Weather Rev 133:3595–3623

Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Bocca JB, Jarke M, Zaniolo C (eds) Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Morgan Kaufmann, pp 487–499

Brotzge J, Droegemeier KK, McLaughlin DJ (2006) Collaborative adaptive sensing of the atmosphere (CASA): new radar system for improving analysis and forecasting of surface weather conditions. J Transp Res Board (1948), pp 145–151

Burgess DW, Donaldson RJ Jr, Desrochers PR (1993) The tornado: its structure, dynamics, prediction, and hazards, vol 79, American Geophysical Union, chap Tornado detection and warning by radar, pp 203–221

Cheng H, Tan PN (2008) Semi-supervised learning with data calibration for long-term time series forecasting. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 133–141

Chiu B, Keogh E, Lonardi S (2003) Probabilistic discovery of time series motifs. In: In the 9th ACM SIGKDD international conference on knowledge discovery and data mining, Washington, DC, pp 493–498

Das G, Lin K, Mannila H, Renganathan G, Smyth P (1998) Rule discovery from time series. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, New York, NY, pp 16–22

Denton A (2005) Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In: Proceedings of the fifth IEEE international conference on data mining, pp 122–129

Donaldson RJ Jr, Dyer RM, Kraus MJ (1975) An objective evaluator of techniques for predicting severe weather events. In: Preprints: ninth conference on severe local storms, American Meteorological Society, pp 321–326

Faloutsos C, Jagadish HV, Mendelzon AO, Milo T (1997) A signature technique for similarity-based queries. In: Proceedings of compression and complexity of sequences, pp 2–20

Goldin D, Mardales R, Nagy G (2006) In search of meaning for time series subsequence clustering: matching algorithms based on a new distance measure. In: Proceedings of the 15th ACM international conference on information and knowledge management, pp 347–356

Hu M, Xue M, Brewster K, Gao J (2004) Prediction of Fort Worth tornadic thunderstorms using 3DVAR and cloud analysis with WSR-88D Level-II data. In: 11th Conference on aviation, range, aerospace and 22nd conference on severe local storms, American Meteorological Society, Electronically published, Paper J1.2

Idé T (2006) Why does subsequence time-series clustering produce sine waves? Lecture Notes in Computer Science. Springer, Berlin/Heidelberg

Johnson JT, MacKeen PL, Witt A, Mitchell ED, Stumpf GJ, Eilts MD, Thomas KW (1998) The storm cell identification and tracking algorithm: an enhanced WSR-88D algorithm. Weather Forecast 13(2): 263–276

Kahveci T, Singh A, Gürel A (2002) Similarity searching for multi-attribute sequences. In: Proceedings of the international conference on scientific and statistical database management, pp 175–184

Kasetty S, Stafford C, Walker GP, Wang X, Keogh E (2008) Real-time classification of streaming sensor data. In: Proceedings of the 20th IEEE international conference on tools with artificial intelligence

Keogh E, Lin J, Truppel W (2003) Clustering of time series subsequences is meaningless: implications for past and future research. In: Proceedings of the 3rd IEEE international conference on data mining, pp 115–122

Keogh E, Lin J, Fu A (2005) HOT SAX: efficiently finding the most unusual time series subsequence. In: Proceedings of the 5th IEEE international conference on data mining (ICDM 2005), Houston, Texas, pp 226–233

Lee SL, Chun SJ, Kim DH, Lee JH, Chung CW (2000) Similarity search for multidimensional data sequences. In: Proceedings of the IEEE international conference on data engineering, pp 599–608

Lin J, Keogh E, Lonardi S, Chiu B (2003) A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery, pp 2–11

Lin J, Keogh E, Li W, Lonardi S (2007) Experiencing SAX: a novel symbolic representation of time series. Data Min Knowl Discov 15(2):107–144

McGovern A, Jensen D (2008) Optimistic pruning for multiple instance learning. Pattern Recognit Lett 29(9):1252–1260

McGovern A, Supinie T, Gagne II DJ, Troutman N, Collier M, Brown RA, Basara J, Williams J (2010) Understanding severe weather processes through spatiotemporal relational random forests. In: 2010 NASA conference on intelligent data understanding (to appear)

McGovern A, Rosendahl DH, Kruger A, Beaton MG, Brown RA, Droegemeier KK (2007) Anticipating the formation of tornadoes through data mining. In: Preprints of the Fifth conference on artificial intelligence and its applications to environmental sciences at the american meteorological society annual meeting, American Meteorological Society, San Antonio, TX, Paper 4.3A

McGovern A, Hiers N, Collier M, Gagne II DJ, Brown RA (2008) Spatiotemporal relational probability trees. In: Proceedings of the 2008 IEEE international conference on data mining, Pisa, Italy, pp 935–940

Mueen A, Keogh E, Zhu Q, Cash S, Westover B (2009) Exact discovery of time series motifs. In: Proceedings of the SIAM international conference on data mining, pp 473–484

Oates T (1999) Identifying distinctive subsequences in multivariate time series by clustering. In: Proceedings of the Fifth international conference on knowledge discovery and data mining, pp 322–326

Oates T, Cohen PR (1996) Searching for structure in multiple streams of data. In: Proceedings of the thirteenth international conference on machine learning, Morgan Kaufmann, pp 346–354

Oates T, Jensen D, Cohen PR (1998) Discovering rules for clustering and predicting asynchronous events. In: Predicting the future: AI approaches to time series workshop, AAAI-98, pp 73–79

Provost FJ, Domingos P (2003) Tree induction for probability-based ranking. Mach Learn 52:199–215

Rosendahl DH (2008) Identifying precursors to strong low-level rotation within numerically simulated supercell thunderstorms: a data mining approach. Master's thesis, School of Meteorology, University of Oklahoma

Schaefer JT (1990) The critical success index as an indicator of warning skill. Weather Forecast 5(4):570–575

Shieh J, Keogh E (2009) iSAX: Indexing and mining terabyte sized time series. In: Proceedings of the IEEE international conference on data mining

Supinie T, McGovern A, Williams J, Abernethy J (2009) Spatiotemporal relational random forests. In: Proceedings of the IEEE international conference on data mining (ICDM) workshop on spatiotemporal data mining, p electronically published

Tanaka Y, Uehara K (2003) Discover motifs in multi-dimensional time-series using the principal component analysis and the mdl principle. In: Proceedings of the third international conference on machine learning and data mining in pattern recognition (MLDM 2003), pp 252–265

Vlachos M, Hadjielefheriou M, Gunopulos D, Keogh E (2006) Indexing multidimensional time-series. Int J Very Large Data Bases 15(1):1–20

Webb GI (1995) OPUS: an efficient admissible algorithm for unordered search. J Artif Intell Res 3:431–465

Xi X, Keogh E, Wei L, Mafra-Neto A (2007) Finding motifs in database of shapes. In: Proceedings of the SIAM international conference on data mining

Xue M, Droegemeier KK, Wong V (2000) The advanced regional prediction system (ARPS)—a multiscale nonhydrostatic atmospheric simulation and prediction model. Part I: model dynamics and verification. Meteorol Atmos Phys 75:161–193

Xue M, Droegemeier KK, Wong V, Shapiro A, Brewster K, Carr F, Weber D, Liu Y, Wang D (2001) The advanced regional prediction system (ARPS)—a multiscale nonhydrostatic atmospheric simulation and prediction tool. Part II: model physics and applications. Meteorol Atmos Phys 76:134–165

Xue M, Wang D, Gao J, Brewster K, Droegemeier KK (2003) The advanced regional prediction system (ARPS), storm-scale numerical weather prediction and data assimilation. Meteorol Atmos Phys 82:139–170

Ye L, Keogh E (2009) Time series shapelets: a new primitive for data mining. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 947–956

Yin J, Gaber MM (2008) Clustering distibutied time series in sensor networks. In: Proceedings of the IEEE international conference on data mining, pp 678–687

Zaki MJ (2001) Spade: An efficient algorithm for mining frequent sequences. Mach Learn 42(1/2):31–60, special issue on unsupervised learning
Zaki MJ, Parimi N, De N, Gao F, Phoophakdee B, Urban J, Chaoji V, Hasan MA, Salem S (2005) Towards generic pattern mining. In: International conference on formal concept anaysis