Tornado Forecasting with Multiple Markov Boundaries

Kui Yu[*,1], Dawei Wang[*,2], Wei Ding[†,2]
Jian Pei[1], David L. Small[3], Shafiqul Islam[3], Xindong Wu[4,5]
[1]School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
[2]Department of Computer Science, University of Massachussets Boston, Boston, USA
[3]Department of Civil and Environmental Engineering, Tufts University, Boston, USA
[4]Department of Computer Science, Hefei University of Technology, Hefei, China
[5]Department of Computer Science, University of Vermont, Burlington, USA
{kuiy,jpei}@cs.sfu.ca, {dawei.wang,ding}@umb.edu
{David.Small,Shafiqul.Islam}@tufts.edu, xwu@uvm.edu

ABSTRACT

Reliable tornado forecasting with a long-lead time can greatly support emergency response and is of vital importance for the economy and society. The large number of meteorological variables in spatiotemporal domains and the complex relationships among variables remain the top difficulties for a long-lead tornado forecasting.

Standard data mining approaches to tackle high dimensionality are usually designed to discover a single set of features without alternating options for domain scientists to select more reliable and physical interpretable variables.

In this work, we provide a new solution to use the concept of multiple Markov boundaries in local causal discovery to identify multiple sets of the precursors for tornado forecasting. Specifically, our algorithm first confines the extremely large feature spaces to a small core feature space, then it mines multiple sets of the precursors from the core feature space that may equally contribute to tornado forecasting. With the multiple sets of the precursors, we are able to report to domain scientists the predictive but practical set of precursors.

An extensive empirical study is conducted on eight benchmark data sets and the historical tornado data near Oklahoma City, OK in the United States. Experimental results show that the tornado precursors we identified can help to improve the reliability of long-lead time catastrophic tornado forecasting.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications-Weather Forecasting

*K. Yu and D. Wang contributed equally to this work
†Corresponding Author

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2788612.

Keywords

Tornado Forecasting, Multiple Markov boundaries, Distributed feature data, Core feature space

1. INTRODUCTION

1.1 Tornado Forecasting

A tornado is one of the most powerful, unpredictable and destructive weather phenomena on Earth. Tornado disasters result in billions of dollars of damages each year and cause more loss of life than most other weather-related hazards. For example, in the year of 2013, 811 confirmed tornadoes cause 55 fatalities and damages up to $3.6 billion in the United States[1]. Reliable tornado forecasts with a long-lead time can greatly support emergency response and are of vital importance for the economy and society. Currently National Weather Service (NWS) issues tornado warnings based on "detection": a tornado threat is broadcasted after being observed, either on-site or through radars [3]. Simulation based warning systems can increase the prediction lead time up to several hours, but most of the predictions are false alarms [15].

Comparing to other domains like electronic advertising, the success of big data-induced process in studying the weather system is limited [7, 19]. Among all the reasons accounting for the slow progress, for data-driven methods trying to model weather dynamics the main difficulty lies in addressing the high dimensionality and the complex relationships among variables. The number of meteorological variables contributing to tornadoes, from the Cartesian product between the spatial and temporal domains, is enormous. Tornadoes are usually developed from thunderstorms and the quasi-geostrophic theory [8] demonstrates that the development of storm events requires a coupling between upper and lower levels of the atmosphere. The atmospheric variables related to this vary both horizontally and vertically and have complex relationships between each other. Even the most efficient Monte Carlo methods will suffer from computational infeasibility for such high dimensional and complex data sets.

Dimension reduction is the standard machine learning approach to deal with high dimensionality [10, 20, 21]. These methods usually report a single subset of features without

[1]http://www.spc.noaa.gov/climo/torn/STATIJ13.txt

alternative options. In our study, the tornado precursors identified are meteorological predictor variables with certain spatial and temporal information, but due to the complex interactions within the variables, the problems of only identifying a single choice precursors are as follows.

(1) Key interpretable precursors may be missed. Due to spatial and temporal adjacency, many precursors contain equivalent information for prediction, because these precursors are not physically independent of one another. For example, when there is a dropping in sea level pressure (anomalously low sea level pressure is an important characteristic of atmospheric regimes, which may lead to extreme precipitations), at the same location there must also be convergence of the winds near the surface and divergence of the winds at the top of the atmosphere. Also, the field of pressure vertical velocity may be strongly negative. In this case, the low level winds, high level winds, and vertical motions in the same location are considered equivalent precursors. A typical feature selection method will only identify one of the three equivalent precursors, the others will not be reported to the domain scientists.

(2) The precursors identified may be considered impractical. It is useful to explore alternative cost-effective but equally effective solutions in the cases where different precursors may have different costs/utilities associated with their acquisition/quality in a forecasting model at the next step. For instance, although the three precursors mentioned in the above example are considered as equivalent precursors, compared to high level and low level winds, vertical motion is a noisy and unreliable precursor. Using vertical motion as a precursor in a prediction model introduces extra uncertainty. Accordingly, if we know all of the three precursors containing equivalent information for forecasting, we can select the low level winds or high level winds for reliable prediction.

1.2 Multiple Markov Boundaries for Tornado Forecasting

It is an important yet new research topic to identify multiple Markov boundaries from data using Bayesian inference in the domain of causal discovery [17]. A Bayesian network is presented by a directed acyclic graph G and a joint probability distribution P over a set of features F [11]. If every conditional independence entailed by G is present in P, G is said to be faithful to P [11]. If a data distribution and an underlying Bayesian network which models that domain are faithful to each other, the Markov boundary of a node is unique and contains its parents, children, and the parents of the children (spouses) [11, 1]. In the past decade, people focused on identification of a single Markov boundary under the faithfulness assumption for causal discovery [1, 18]. However, tornado and other many real-world data distributions often violate the faithfulness condition due to various factors, such as (but not limited to) small sample size, noise in data, and hidden variables. Thus the Bayesian networks built from such data often contains multiple Markov boundaries [12, 17]. Developing methods of discovery of multiple Markov boundaries would improve the discovery of the underlying mechanisms to avoid overlooking key causal variables, and thus this can be useful to explore alternative cost-effective but equally effective solutions in prediction in applications where different variables may have different costs/utilities associated with their acquisition/quality.

1.3 Our Contributions

Our contributions to tornado forecasting are below.

- To cope with high-dimensionality, we define a new concept of a core feature space to represent the feature space of all possible Markov boundaries of a target variable, and then confine mining multiple Markov boundaries to this core feature space instead of the original entire feature space.
- To discover this core feature space from distributed feature data sets, we process distributed feature data sets in a sequential scan without loading the whole tornado data set in memory in advance. Specifically, we process each distributed tornado data set one at a time, and features in each set are processed one-by-one in a sequential scan.
- We design and implement the MB-DEA algorithm, to discover multiple Markov Boundaries from Distributed fEature dAta for tornado forecasting. MB-DEA firstly discovers the core feature space from which multiple Markov boundaries are mined.
- We apply the MB-DEA algorithm to study the historical tornado data near Oklahoma City, OK. Our empirical study includes 810,000 features of 35 years data. We are able to identify physically meaningful and practical tornado precursors which may lead to reliable and long-lead time of catastrophic tornado forecasting.

The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 describes the tornado data set and Section 4 proposes the MB-DEA algorithm, respectively. Section 5 reports our experimental results on tornado forecasting. Finally, Section 6 concludes the paper.

2. RELATED WORK

We will give a brief review of tornado forecasting and multiple Markov boundaries in this section. Tornadoes are usually developed within thunderstorms and most tornado warning systems are based on the prediction of thunderstorms [3]. One official National Weather Service product is the tornado watch, which is based on weather forecasting model outputs and observations, with a lead time up to several hours. Simulations of supercell thunderstorms are performed in [16] with the aim of discovering precursors of tornadoes. Such process usually resulted in a large number of false alarms (false positives) because only very few storms would produce tornadoes. A three-dimensional (in space and time) object identification algorithm is applied in [4] for forecasting tornado path lengths through the predictions of hourly maximum updraft helicity as a measure of storm severity. In [13] the authors implement principal component analysis (PCA) on the outputs from a simulation model (the Weather Research and Forecasting model), and build a tornado prediction model using supported vector machine with the PCA results. They have achieved an accuracy of 0.7 with a one day lead time.

Causal discovery is of fundamental and practical interest in many areas of science and technology [11]. Global causal induction attempts to learn a complete Bayesian network over all features on training data. However, global causal induction is not scalable and cannot even handle thousands

of features [5]. Local causal discovery aims to learn a local causal structure closely related to a target feature of interest, i.e., a Markov boundary of a target feature of interest. If a joint probability distribution satisfies the faithful condition, it is guaranteed to have a unique Markov boundary for every node in a Bayesian network [1]. Accordingly, in the past decade, most of the existing algorithms of local causal discovery focused on identification of a single Markov boundary under the faithfulness assumption [1, 12, 18].

However, many real-world data distributions often violate the faithfulness condition due to small sample sizes, noise in data, and hidden variables, and thus contain multiple Markov boundaries [12, 17, 22]. Algorithms of multiple Markov boundaries attempts to discover all Markov boundaries of a target feature containing in data without missing causative variables [17]. Among the most notable advances in the field of discovery of multiple Markov boundaries are the KIAMB algorithm and the TIE* algorithm [17]. Peña et al. [12] proposed a stochastic Markov boundary algorithm, called KIAMB by employing a stochastic search heuristic that repeatedly disrupts the order in which features are selected for inclusion into a Markov boundary with the probability p at each round, thereby introducing a chance of identifying alternative Markov boundaries of a target feature. The limitation is that we do not know how many iterations the KIAMB algorithm needs to run because the exact number of Markov boundaries of a target feature is unknown and varies in different data. To solve this problem, Statnikov et al. [17] recently proposed the TIE* (Target Information Equivalence) algorithm and proved that TIE* can discover all Markov boundaries under the non-faithful condition.

Integrating the drawbacks of the existing work for long-lead tornado forecasting and multiple Markov boundaries, it provides a natural choice to find multiple precursors for long-lead tornado prediction using multiple Markov boundaries. However, both the KIAMB and TIE* algorithms focus on mining multiple Markov boundaries from data on a single feature set. With the real-world tornado data, the KIAMB and TIE* algorithms face the challenges of both very high dimensionality and multiple feature sets. This motivates us to further investigate new algorithms of multiple Markov boundaries for long-lead tornado prediction.

3. THE TORNADO DATA SET

The tornado data set contains eight explanatory variables and tornado information near Oklahoma City, OK, one of the most tornado-prone areas in the United States. All the explanatory variables are sampled at the spatial domain of $90°E$ to $357.5°E$ and $0°N$ to $90°N$ with a horizontal resolution of $2.5° \times 2.5°$ (totally $2,700$ locations) and a daily temporal resolution for the months of March, April and May (MAM, when the highest frequency of violent tornadoes occurs in the studied area) for the years 1979-2013. The tornado study area near Oklahoma City is located in the spatial region of $261.25°E - 263.75°E$ and $33.75°N - 36.25°N$. The eight variables at different levels (Table 1) are selected from the NCEP-NCAR Reanalysis data set [9] by the domain scientists (co-authors of the paper) for the study. In particular, the Relative Humidity data only goes from 1000hPa to 300hPa because the amount of water in the upper troposphere was thought to be negligible when the dataset was designed. Two variables only have one single level (for Sea Level Pressure, the values represent the surface level; for Precipitable Water, the values are column integrated from all the levels). A particular day of the tornado study area is labeled as a positive instance if one or more $EF1$(Enhanced Fujita scale [6]) or above tornadoes are reported, otherwise a negative instance. In total there are 98 tornado days (positive instances) out of the $3,045$ days study period.

We set our "look ahead" days as 5. For example, to forecast tornado situations at tomorrow ($day0$) in the study area, we will look at the explanatory variables from today (day-1) back to previous five days (day-5). The tornado data set presents two characteristics of a difficult task:

- Extremely high dimensionality. With such a setting, the total number of precursors (features) for one instance is $810,000$ in the tornado data set (e.g., the variable of Relative Humidity has 8 levels, 5 days, and 2700 locations. In total there are $8 \times 5 \times 2700 = 108,000$ features). Any existing algorithm of multiple Markov boundaries cannot cope with such large number of features.
- Distributed feature data. The whole tornado data set is large (more than 13GB in total), and thus we have the data distributed in eight data sets according to the eight explanatory variables and one class attribute set. On average each explanatory variable set still has more than $100,000$ features. Mining multiple Markov boundaries from multiple feature data sets is an untouched research topic.

4. MACHINE LEARNING FORMULATION

In this section, we discuss the MB-DEA algorithm for tornado forecasting. The algorithm tackles the discovery of multiple Markov Boundaries from Distributed fEature dAta. The design of the MB-DEA algorithm consists of (1) finding the core feature space from the distributed tornado data, and (2) mining multiple Markov boundaries from the core feature space.

4.1 Notations and Definitions

Given a training data set D containing N training instances and M features, we define a distributed feature data set as that D is vertically divided into W feature blocks without overlapping features between each block, that is, $D = \{B_1, B_2, \cdots, B_W\}$ which is distributed in W files. F_i $(1 \leq i \leq M)$ is an feature within B_j $(1 \leq j \leq W)$ and C is the class attribute. Let P be a joint probability distribution on a set of random variables F via a directed acyclic graph G. We call the triplet $\langle F, G, P \rangle$ a Bayesian network if $\langle F, G, P \rangle$ satisfies the Markov condition: every variable is conditionally independent of any subset of its non-descendant variables given its parents [11].

In the rest of the paper, the terms "variable" and "feature" are used interchangeably. To measure the statistical relationships between features, we adopt the measure of mutual information [14]. Given two variables X and Y, the mutual information between X and Y is defined as follows.

$$I(X;Y) = H(X) - H(X|Y) \tag{1}$$

where $H(Y)$ and $H(Y|Z)$ are computed as follows.

$$H(X) = -\sum_{x_i \in X} (P(x_i) \log_2(P(x_i)) \tag{2}$$

Table 1: Meteorological variables

Name	Level(hPa)
Temperature	200,250,300,400,500,600,700,850,925,1000
Geopotential Height	200,250,300,400,500,600,700,850,925,1000
Meridional Wind	200,250,300,400,500,600,700,850,925,1000
Zonal Wind	200,250,300,400,500,600,700,850,925,1000
Pressure Vertical Velocity	200,250,300,400,500,600,700,850,925,1000
Relative Humidity	300,400,500,600,700,850,925,1000
Sea Level Pressure	–
Precipitable Water	–

$$H(X|Y) = -\sum_{y_j \in Y} P(y_j) \sum_{x_i \in X} (P(x_i|y_j) \log_2(P(x_i|y_j)) \quad (3)$$

From Equations (1) to (3), the conditional mutual information is computed by

$$\begin{aligned} &I(X;Y|Z) \\ &= H(X|Z) - H(X|YZ) \\ &= -\sum_{z_i \in Z} P(z_i) \sum_{x_j \in X} \sum_{y_k \in Y} P(x_j y_k|z_i) \log_2 \frac{P(x_j y_k|z_i)}{P(x_j|z_i)P(y_k|z_i)} \end{aligned} \quad (4)$$

The lower cases x_i, y_i, and z_i in the above equations denote possible values that the variables X, Y and Z take.

Definition 1 (Faithfulness). *[11] Give a Bayesian network $\langle F, G, P\rangle$, G is faithful to P over F if and only if every independence present in P is entailed by G and the Markov condition. P is faithful if and only if there exists a directed acyclic graph G such that G is faithful to P.* ■

Definition 2 (Markov boundary). *[11] If a Bayesian network satisfies the faithfulness, the Markov boundary of any node T in this Bayesian network is unique with the set of parents, children and spouses (the parents of the children of T) of T.* ■

However, if a Bayesian network does not satisfy the faithfulness, the Markov boundary of any node may not be unique [17]. We use Theorem 1 to explicitly construct and verify multiple Markov boundaries when the distribution P violates the faithful condition.

Theorem 1. *[17] If MB_1 is a Markov boundary of T that contains a feature set S_1, and there exists a subset S_2 such that $MB_2 \subseteq (MB_1 - S_1) \cup S_2$, if $I(T; MB_1|MB_2) = 0$, then MB_2 is also a Markov boundary of T.* ■

In Theorem 1, both MB_1 and MB_2 are Markov boundaries of T, since $S_1 \subset MB_1$ and $S_2 \subset MB_2$ contain the equivalent information about T.

4.2 The Core Feature Space

According to the design of the MB-DEA algorithm, the key to the algorithm is how to discover the core feature space from distributed feature data. The core feature space we are looking for is defined as a feature space that contains all possible Markov boundaries of a target feature. By theorem 1, we get Corollary 1 below.

Corollary 1. *Assuming MB_1 is a Markov boundary of T and $MB_2 \subseteq \{MB_1 - \{F_i\} \cup \{F_j\}\}$, if $I(C; F_j|F_i) = 0$, MB_2 is also a Markov boundary of T.* ■

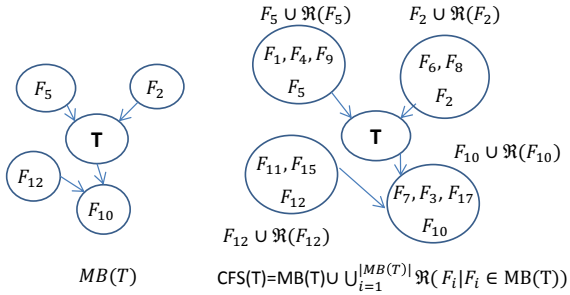


Figure 1: An example of $MB(T)$ and $CFS(T)$

Definition 3. *Assuming $MB(C)$ is a Markov boundary of C and $F_i \in MB(C)$, we define $\Re(F_i)$ as the set of features that satisfies the term $\forall F_j \in \Re(F_i)$ s.t. $I(C; F_j|F_i) = 0$.* ■

By Corollary 1 and Definition 3, we define the core feature space of C as follows.

Definition 4 (Core feature space). *Assuming $MB(C)$ is a Markov boundary of C and $F_i \in MB(C)$, the core feature space (CFS) of C is defined as*

$$CFS(C) = \{MB(C) \cup \bigcup_{F_i \in MB(C)} \Re(F_i)\}.$$

■

With Definition 4, our first problem is how to find $MB(C)$ efficiently. To find $MB(C)$, in general, we can use the existing single Markov boundary algorithms, such as HITION-MB [1] and MMMB [18]. For example, assuming a data set contains 20 features $F = \{F_1, F_2, \cdots, F_{20}\}$ and feature F_{20} is considered as a target feature T, if $MB(T) = \{F_5, F_2, F_{12}, F_{10}\}$, a Markov blanket of T, is found by HITION-MB or MMMB and $\Re(F_5) = \{F_1, F_4, F_9\}$, Figure 1 illustrates the relationships between $MB(T)$ (the left figure) and $CFS(T)$ (the right figure).

To find the core feature space from distributed feature data, we further analyze the relationships between features F_i and F_j in Corollaries 2 and 4 as follows.

Corollary 2. *A feature X is correlated to Y with respect to a feature subset S if $I(X; Y|S) > 0$.*

Proof. By Eq.(4), we can see that the term $I(X; Y|S) = 0$ holds only when X and Y are conditionally independent given S. The corollary is proven accordingly. ■

Corollary 3. *If F_i is correlated to C and $I(C; F_j|F_i) = 0$ holds, then $I(F_i, C) > I(F_j, C)$.*

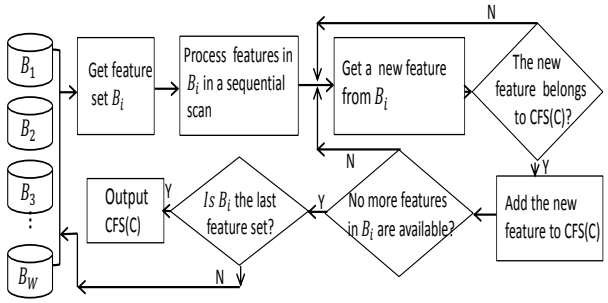


Figure 2: Discovery of the core feature space of C from distributed feature data

Proof. By $I(A;B|C)-I(A;B) = I(A;C|B)-I(A;C)$, we get $I(C;F_j|F_i) - I(C;F_j) = I(C;F_i|F_j) - I(C;F_i)$. Since the term $I(C;F_j|F_i) = 0$ hold, then $I(C;F_i) = I(C;F_i|F_j)+ I(C;F_j)$. By Corollary 2, the corollary is proven. ∎

COROLLARY 4. *If both MB_1 and $MB_2 \subseteq (MB_1-\{F_i\})\cup \{F_j\}$ are Markov boundaries of C and $I(F_j;C|F_i) = 0$ holds, then $I(F_i;F_j) \geq \max(I(F_i;C), I(F_j;C))$.*

Proof.

$$\begin{aligned} I(F_j;F_i,C) &= I(F_j;F_i) + I(F_j;C|F_i) \\ &= I(F_j;C) + I(F_j;F_i|C) \end{aligned} \quad (5)$$

Since $I(F_j;C|F_i) = 0$ holds and F_i and F_j are correlated, we can get $I(F_i;F_j) \geq I(F_j;C)$. And at the same time, the following equation also holds.

$$\begin{aligned} I(F_i;F_j,C) &= I(F_i;F_j) + I(F_i;C|F_j) \\ &= I(F_i;C) + I(F_i;F_j|C) \end{aligned} \quad (6)$$

By Corollary 1, we can also get $I(F_i;C|F_j) = 0$. By Eq.(6), we have $I(F_i;F_j) \geq I(F_i;C)$. With the equations (5) and (6), Corollary 4 is proved. ∎

By Corollaries 3 and 4, we can get the following.

OBSERVATION 1. *If $I(F_i;C) > I(F_j;C)$ and $I(F_i;F_j) \geq max(I(F_i;C), I(F_j;C))$, then F_i is in a Markov boundary of C ($MB(C)$), and F_j is included by $\Re(F_i)$.* ∎

4.3 Discovery of Multiple Markov Boundaries from Distributed Feature Data

Figure 2 gives the new framework to efficiently find the core feature space of C from distributed feature data. In Figure 2, each feature block B_i is processed independently at a time, and as a feature block B_i arrives, features in B_i are processed one-by-one in a sequential scan.

As illustrated in Figure 2, we discuss the pseudocode of the MB-DEA algorithm in Algorithm 1. To achieve a relatively low computational complexity to deal with high-dimensional yet distributed tornado data, according to Observation 1, Corollaries 3 and 4, the MB-DEA algorithm uses online pairwise comparisons as the selection criterion for adding features into the core feature space.

In Algorithm 1, $\Re(F_i)$ keeps the set of features correlated to F_i that satisfies Definition 3; $\Re$ dynamically keeps the features in $\bigcup_{i=1}^{|MB(C)|} \Re(F_i)$; $MB(C)$ is defined in Definition 4; and $CFS(C)$ denotes the core feature space of C. Two key

Algorithm 1: *The MB-DEA Algorithm*

Data: $D = \{B_1, B_2, \cdots, B_W\}$;
$MB(C) = \{\}$;$CFS(C) = \{\}$;$i = 0$;
$\delta = max(I(Y;C), I(F_i;C))$

```
1  /*Steps 2-25: Mining the core feature space CFS*/
2  for j=1 to W do
3      Get-new-feature-set(B_j);
4      repeat
5          i = i + 1;
6          Get-new-feature(F_i, B_j);
7          if I(F_i;C) = 0 then
8              Discard F_i; goto step 23;
9          end
10         for each feature Y ∈ MB(C) do
11             if I(Y;C) > I(F_i;C) and I(F_i;Y) ≥ δ then
12                 ℜ(Y) = ℜ(Y) + F_i;
13                 ℜ = ℜ + F_i;
14                 Goto step 23;
15             end
16             if I(F_i;C) > I(Y,C) and I(F_i;Y) ≥ δ then
17                 MB(C) = MB(C) − Y;
18                 ℜ(F_i) = ℜ(F_i) + Y;
19                 ℜ = ℜ − ℜ(Y);
20             end
21         end
22         MB(C) = MB(C) ∪ F_i;
23     until F_i is the last feature of B_j;
24 end
25 CFS(C) = MB(C) ∪ ℜ
26 /*Steps 27-35: Mining Markov boundaries from CFS*/
27 Use HITION_PC to learn a Markov boundary MB_1 of
   C from D over CFS(C) (the original distribution);
28 Output MB_1;
29 repeat
30     Generate a data set D_e from the embedded
       distribution by removing a subset of features within
       the discovered Markov boundaries from CFS(C);
31     Use HITION_PC to learn a Markov boundary
       MB_new of C from D_e;
32     if acc(MB_new) ≥ acc(MB_1) then
33         output MB_new;
34     end
35 until all data sets D_e generated have been considered;
```

stages are given in Algorithm 1. One is to mine $CFS(C)$ from Steps 2 to 25, and the other is to discover Markov boundaries from $CFS(C)$ from Steps 27 to 35.

Steps 2 to 25. As a feature set B_j is achieved, features within B_j are processed one-by-one in a sequential scan. Step 7 is to determine whether a new coming feature F_i is correlated to C; if $I(F_i;C) > 0$, then we consider F_i is correlated to C; otherwise, F_i is discarded.

If F_i is correlated to C, Step 11 determines whether to keep F_i given the current set $MB(C)$. If not, F_i is added to $\Re$ and $\Re(Y)$, respectively. If F_i can be added to $MB(C)$ at Step 11, Step 16 further prunes $MB(C)$ due to F_i's inclusion. If Y in $MB(C)$ is removed, the set $\Re(Y)$ is removed from $\Re$ accordingly. Meanwhile, Y is added to $\Re(F_i)$. With the MB-DEA algorithm, we can discover $\Re$ without any additional computation costs.

Table 2: Summary of the benchmark data sets

	Data set	♯ features	♯ training set	♯ testing set
1	arcene	10,000	100	100
2	dexter	20,000	300	300
3	dorothea	100,000	800	300
4	colon	2,000	42	20
5	leukemia	7,129	48	24
6	lung-cancer	12,533	121	60
7	ovarian-cancer	2,190	144	72
8	thrombin	139,351	2,000	543

Steps 27 to 35. We integrate the TIE* algorithm into our MB-DEA algorithm to mine Markov boundaries from the discovered core feature space. The main idea of TIE* [17] is to first identify a Markov boundary of a target feature T in the original data distribution and then iteratively run a single Markov boundary induction algorithm from the embedded distributions that are obtained by removing subsets of features from the original Markov boundary in order to identify new Markov boundaries in the original distribution. From Steps 27 to 35, with the core feature space, the TIE* algorithm can search for multiple Markov boundaries in a smaller feature space.

Step 27 uses a single Markov boundary induction algorithm HITION_PC [1] to learn a Markov boundary, called MB_1, from the data set D defined on the feature set $CFS(C)$ (i.e., in the original distribution). Step 30 generates a data set D_e (the embedded distribution) that removes a subset of features from CFS (Regarding how to generate an embedded distribution, please see the IGS algorithm in [17], Page18, Figure 9). The motivation is that D_e may lead to identification of a new Markov boundary of T that was previously "invisible" to a single Markov boundary induction algorithm, because it is shielded by another subset of features within the discovered Markov boundaries. Step 32 uses prediction accuracy as a criterion to verify whether a discovered feature set from the embedded distribution is a new Markov boundary or not. If the prediction accuracy of MB_{new} in Step 32 is not less than that of MB_1, MB_{new} is also considered as a Markov boundary of C. Steps 30-34 are repeated until all data sets D_e generated have been considered.

5. EMPIRICAL STUDY

In this empirical study, since the state-of-the-art multiple Markov boundary discovery algorithm, the TIE* algorithm [17], cannot deal with the high-dimensional yet distributed tornado data set, we will first validate the effectiveness and efficiency of the MB-DEA algorithm against the TIE* algorithm using the benchmark data sets, and then use the MB-DEA algorithm for tornado prediction.

5.1 Experiments on the Benchmark Data Sets

We have chosen eight benchmark data sets as described in Table 2, which cover a wide range of real-world application domains. In Table 2, for the first four NIPS 2003 challenge data sets and the *hiva* data set from the WCCI 2006 performance prediction challenge, we use the originally provided training and validation sets; for the other five data sets we adopt the first 2/3 instances for training and the last 1/3 instances for testing. As for discrete data sets, we use mutual information to compute the correlation between features while for the continuous data sets, the Fisher's Z-test is employed [2] in which the significance level for the Fisher's Z-test is set to 0.01. All experiments on the benchmark data sets are conducted on a computer with Intel(R) i7-3770 3.4GHz CPU and 32GB memory.

Table 3: The F-ratio and MB-ratio

Data set	F-ratio	MB-ratio
arcene	0.1157	1
dexter	0.0057	-
dorothea	0.0636	1
colon	0.033	0.5
leukemia	0.0537	1
lung-cancer	0.0.0608	1
thrombin	0.0249	-
ovarian-cancer	0.1685	1

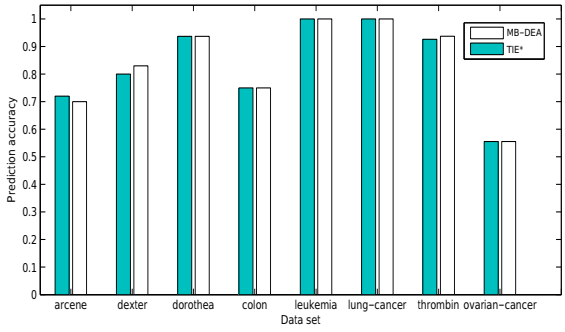


Figure 3: Prediction (classification) accuracies of MB-DEA against TIE*

In our experiments, each training data set in Table 2 is partitioned into ten feature sets evenly to simulate distributed feature data for MB-DEA while TIE* is directly implemented on each training data set. TIE* is parameterized with HITON_PC [1] as the base Markov blanket induction algorithm, and classification accuracy as a criterion is used to verify whether a new feature subset is a Markov boundary or not. The parameter alpha of HITION_PC is set to 0.01.

In the following figures and tables, we report the comparison results of MB-DEA and TIE*. The running time on the *dexter* and *thrombin* data sets exceed 3 days for TIE* and we do not show the experimental results of TIE* on those two data sets.

Table 3 reports the F-ratio and MB-ratio. An F-ratio is the ratio of the number of features of the core feature space discovered by MB-DEA divided by the number of total features on the same data set. An MB-ratio is the ratio of the number of Markov boundaries discovered by MB-DEA in the number of Markov boundaries identified by TIE*. In Table 3, we can see that on each data set, the number of features within the core feature space is a very small fraction of the total number of features. Furthermore, from the MB-ration, except for the *madelon* and *thrombin*, MB-DEA discovers the same number of Markov boundaries as TIE* on the remaining data sets. For the *dexter* and *thrombin* data sets, we do not have their MB-ratios due to long running time of TIE* and denote them as "-".

Table 4: **Efficiency of MB-DEA vs.TIE*(seconds)**

Data set	TIE*	MB-DEA
arcene	19	6
dexter	-	17,091
dorothea	6,465	25
colon	5	1
leukemia	15	6
lung-cancer	79	16
thrombin	-	24,935
ovarian-cancer	6	3

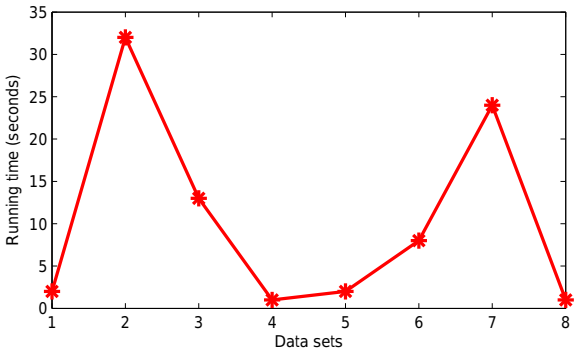


Figure 4: Running time of the discovery of the core feature space. The labels of the x-axis in both figures from 1 to 8 denote the data sets:1.arcene, 2.dexter, 3.dorothea, 4.colon, 5.leukemia, 6.lung-cancer, 7.thrombin, 8.ovarian-cancer.

Meanwhile, Figure 3 gives prediction (classification) accuracies of MB-DEA against TIE* using the K Nearest Neighbor classifier. We select the highest prediction accuracy among all of the Markov boundaries discovered by MB-DEA and TIE*, respectively. From Figure 3, we can see that with the core feature space, MB-DEA gets the same prediction accuracy as TIE*. Although MB-DEA does not find all of Markov boundaries on the *colon* data set, it still gets the Markov boundaries with the highest prediction accuracies.

Figure 4 gives the running time of the discovery of the core feature space from Steps 2 to 25 in Algorithm 1. From Figure 4, we can see that the computational cost of the discovery of the core feature space is very low due to identifying the $\Re$ feature set (see Definition 3) without additional time costs. Table 4 gives the running time of MB-DEA against TIE*. From Table 4, we can see that in term of efficiency, MB-DEA is faster than TIE* on all of ten data sets. From the *dorothea*, *dexter*, and *thrombin* data sets in Table 4, with the core feature space, MB-DEA can efficiently deal with data with very high-dimensionality while TIE* is very expensive, or even impossible due to its exhaustive search over the entire feature space.

In summary, with the above results, we can conclude that the core feature space that the MB-DEA algorithm has discovered is only a small fraction of the entire feature space, but MB-DEA gets a very promising prediction accuracy with a reasonable running time. Therefore, the MB-DEA algorithm is a scalable and accurate method to deal with distributed feature data.

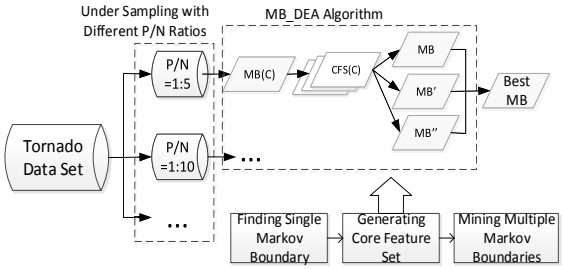


Figure 5: Flow chart of the experimental process. Using an undersampled data set as input, MB-DEA outputs the multiple Markov Boundaries and a best Markov Boundary, or the tornado precursor set, can be found according to the interesting measures.

5.2 Tornado Forecasting

5.2.1 *Experiment Setup*

Tornadoes are rare events even during the most frequent months. The positive and negative instances ratio in our study is about 1:30. We use both accuracy (Equation 7) and F1 score (Equation 8) as the performance metrics in our empirical study.

$$Accuracy = \frac{TP + FP}{TP + TN + FP + FN} \tag{7}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{8}$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives.

We divide the data set into two sets, 30 years (1979-2008, 2,610 days with 81 positive instances) as the training set and 5 years (2009-2013, 435 days with 17 positive instances) as the test set. As illustrated in Figure 5, we randomly under sample the negative instances to achieve class ratios of positive: negative (P/N) examples that are equal to 1:5, 1:10, 1:20, and 1:25 respectively. At each ratio level we perform the random process 10 times and run the MB-DEA algorithm with the correspondent training data set to identify tornado precursors from multiple Markov boundaries. We apply the tornado precursors to the K Nearest Neighbor classifier with K=1 on the test set. The average and best prediction results are reported in Table 5, in which the average values are calculated using the best precursors identified from each training data set. We plot the experimental results in Figures 6 to 7.

5.2.2 *Results and Discussion*

Our goal is to identify key interpretable tornado precursors that can latterly be used in forecasting models. As discussed earlier, we start from a single Markov boundary, then derive a core feature set from it, eventually identify the best tornado precursors. The average F1 values of the best tornado precursor sets are much higher than those of the initial single Markov boundaries at all the sampling levels (Table 5). The best average forecasting result (average accuracy=0.93, average F1=0.31) is achieved at the undersampling level of P/N ratio 1:20 and the best predicting

Table 5: The average and best prediction results

P/N Ratio	Ave. F1 of $MB(C)$	Ave. Size of CFS(C)	Ave. Acc.	Ave. F1	Best Acc.	Best F1
1:5	0.1041	269	0.9097	0.1907	0.9241	0.2667
1:10	0.1333	217	0.9218	0.2440	0.9253	0.2917
1:15	0.1579	233	0.9296	0.2688	0.9448	0.2941
1:20	0.1739	243	0.9255	**0.3129**	0.9310	0.3478
1:25	0.1778	245	0.9264	0.3018	0.9402	**0.3500**
Without Undersampling	0.1429	238	–	–	0.9218	0.2778

result (accuracy=0.94, F1=0.35) is achieved with P/N ratio of 1:25, in which we have successfully predicted 7 tornado events one day ahead out of 17 during the testing period (MAM 2009-2013, 435 days), with 16 false positives. To the best of our knowledge, our result among the most promising tornado forecasting results at daily level compared to the state-of-the-art algorithms [3].

Figure 6 uses our empirical results to explain how MB-DEA works. We plot the single Markov Boundary ($MB(C)$ in Algorithm 1), the core feature set $CFS(C)$, and the tornado precursors (the best Markov boundary according to the criteria of prediction powers) with respect to the experiment having best prediction results (F1=0.35) in Figure 6. Firstly, the single Markov boundary $MB(C)$ is learned from the tornado data set (Figure 6a), then the core feature set $CFS(C)$ is built based on the $MB(C)$ (Figure 6b-6e); finally, the best Markov boundary according to prediction powers, is reported (Figure 6f). Different variables in the core feature set $CFS(C)$ fall into clusters in the spatial domains (blue circles in Figure 6b-6e). The arrows from Figure 6a to Figure 6d show that how a feature from the field of Pressure Vertical Velocity helps to generate a cluster of related features (mostly from the field of Pressure Vertical Velocity and Relative Humidity), and later contributes to the identification of precursor 12 (Figure 6f). The variables from the same field closed to each other in the spatial domains will tend to have similar values due to the spatial autocorrelation effect [23]. These spatial clusters can be considered as real-world examples of the $CFS(C)$ illustrated in Figure 2. Our algorithm picks individual precursors out of each cluster and successfully finds the best combination (Markov Boundary) according to the forecasting task.

As an algorithm of discovering multiple Markov boundaries, MB-DEA is not limited to finding the feature set with the best prediction result. MB-DEA is able to report different precursor sets with similar prediction powers according to other domain interests. For example, compared to other variables in the tornado data set, the Pressure Vertical Velocity is considered as a noisy and unreliable variable. In the feature set with the best prediction result (Figure 6f, b-set for short), we have two features (No.7 and No.12) from the field of Pressure Vertical Velocity. From all the MB-DEA outputs generated from the same input data set, we are able to find a feature set whose features are all from fields other than Pressure Vertical Velocity (Figure 7, we call it the "most reliable" set, or m-set for short.) and have similar prediction power (F1=0.3265). From Figure 6f and Figure 7, we find that more than half of the features in the two sets (No.1,2,3,4,8,10,and 11) are exactly the same, and others (except No.6) are from the same spatial clusters in the core feature set. In the m-set instead of Pressure Vertical Velocity, variables from Relative Humidity are selected (No.5 and No.7 in m-set, compared to No.12 and No.7 in b-set, respectively). MB-DEA provides a new approach of bridging data-driven and knowledge-driven processes for better data interpretation to be used in various forecasting models.

6. CONCLUSION

The development of reliable tornado forecasting techniques able to provide warnings at a long lead-time is crucial to help human beings better prepare and respond to disastrous events. In this paper, we propose the MB-DEA algorithm to identify multiple sets of the precursors for reliable tornado forecasting using multiple Markov boundaries. Our empirical study reveals that the precursors identified by our algorithm are considered more reliable and practical than the ones identified by the single Markov boundary discovery algorithm, and lead to advancements in reliable and long-lead time of catastrophic tornado forecasting.

7. ACKNOWLEDGMENTS

This work is partly supported by a PIMS Post-Doctoral Fellowship Award of the Pacific Institute for the Mathematical Sciences, Canada, the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China (under grant IRT13059), the National 973 Program of China (under grant 2013CB329604), the National Natural Science Foundation of China (under grants 61229301 and 61305064), an NSERC Discovery grant and a BCIC NRAS Team Project. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

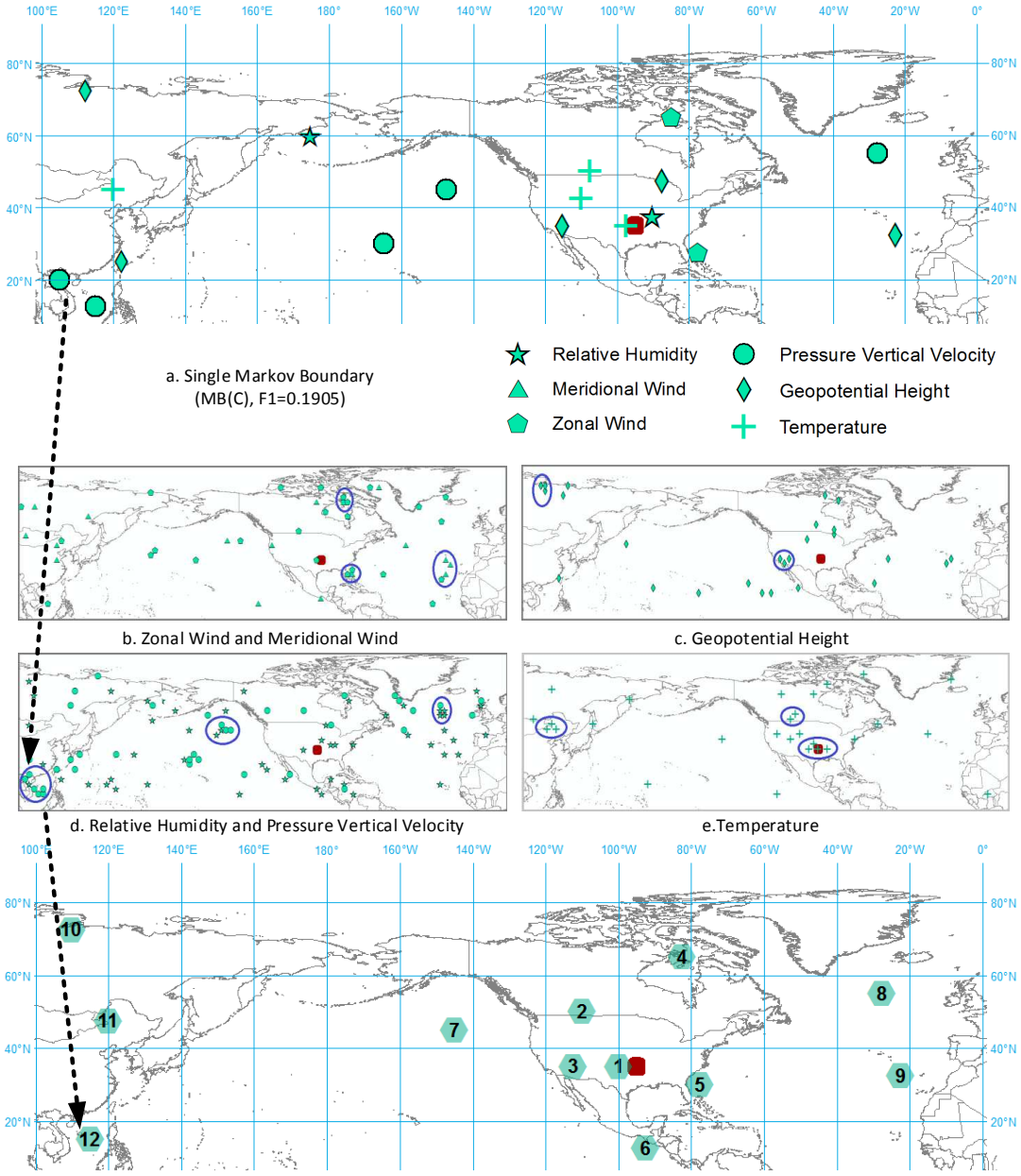
8. REFERENCES

[1] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010.

[2] T. W. Anderson, T. W. Anderson, T. W. Anderson, and T. W. Anderson. *An introduction to multivariate statistical analysis*, volume 2. Wiley New York, 1958.

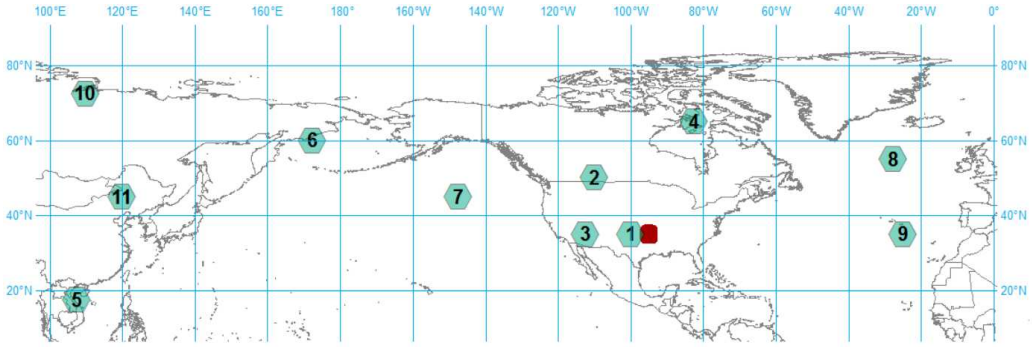
[3] J. Brotzge and W. Donner. The tornado warning process: A review of current research, challenges, and opportunities. *Bulletin of the American Meteorological Society*, 94(11):1715–1733, 2013.

[4] A. J. Clark, J. S. Kain, P. T. Marsh, J. Correia Jr, M. Xue, and F. Kong. Forecasting tornado pathlengths using a three-dimensional object identification algorithm applied to convection-allowing forecasts. *Weather and Forecasting*, 27(5):1090–1113, 2012.



No.	Name	Level	Look Ahead Days	No.	Name	Level	Look Ahead Days
1	Temperature	1000	-1	7	Pressure Vertical Velocity	1000	-4
2	Temperature	1000	-3	8	Relative Humidity	850	-5
3	Geopotential Height	925	-1	9	Meridional Wind	1000	-3
4	Zonal Wind	1000	-5	10	Geopotential Height	700	-4
5	Zonal Wind	600	-4	11	Temperature	1000	-3
6	Relative Humidity	1000	-5	12	Pressure Vertical Velocity	400	-3

Figure 6: A real world example on the processes of how MB-DEA works. MB-DEA firstly discovers a single Markov Boundary (Figure (a): $MB(C)$ from Algorithm 1). The partial CFS (Figures (b) to (e)) shows feature clusters in the spatial domain built based on features from $MB(C)$, and finally the best feature set (b-set (Figure (f)), according to the prediction power) is found from CFS. The red square in each map is the target area for tornado labeling.



Feature Set with Most Reliable Variables (F1=0.3265)

No.	Name	Level	Look Ahead Days	No.	Name	Level	Look Ahead Days
1	Temperature	1000	-1	7	Relative Humidity	1000	-4
2	Temperature	1000	-3	8	Relative Humidity	850	-5
3	Geopotential Height	925	-1	9	Meridional Wind	1000	-3
4	Zonal Wind	1000	-5	10	Geopotential Height	700	-4
5	Relative Humidity	1000	-3	11	Temperature	1000	-3
6	Relative Humidity	1000	-4				

Figure 7: The most reliable feature set (according to the types of variables in the set). The red square in the map is the target area for tornado labeling.

[5] C. P. De Campos and Q. Ji. Efficient structure learning of bayesian networks using constraints. *Journal of Machine Learning Research*, 12:663–689, 2011.

[6] C. A. Doswell, H. E. Brooks, and N. Dotzek. On the implementation of the enhanced fujita scale in the usa. *Atmospheric Research*, 93(1):554–563, 2009.

[7] J. H. Faghmous and V. Kumar. A big data guide to understanding climate change: The case for theory-guided data science. *Big data*, 2(3):155–163, 2014.

[8] I. M. Held, R. T. Pierrehumbert, S. T. Garner, and K. L. Swanson. Surface quasi-geostrophic dynamics. *Journal of Fluid Mechanics*, 282:1–20, 1995.

[9] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, et al. The ncep/ncar 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–471, 1996.

[10] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491–502, 2005.

[11] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[12] J. M. Peña, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007.

[13] C. M. Shafer, A. E. Mercer, L. M. Leslie, M. B. Richman, and C. A. Doswell III. Evaluation of wrf model simulations of tornadic and nontornadic outbreaks occurring in the spring and fall. *Monthly Weather Review*, 138(11):4098–4119, 2010.

[14] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[15] K. Simmons and D. Sutter. *Economic and societal impacts of tornadoes*. Springer Science & Business Media, 2013.

[16] N. Snook and M. Xue. Effects of microphysical drop size distribution on tornadogenesis in supercell thunderstorms. *Geophysical Research Letters*, 35(24), 2008.

[17] A. Statnikov, J. Lemeir, and C. F. Aliferis. Algorithms for discovery of multiple markov boundaries. *Journal of Machine Learning Research*, 14(1):499–566, 2013.

[18] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[19] D. Wang, W. Ding, K. Yu, X. Wu, P. Chen, D. L. Small, and S. Islam. Towards long-lead forecasting of extreme flood events: a data mining framework for precipitation cluster precursors identification. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1285–1293. ACM, 2013.

[20] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu. Online feature selection with streaming features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5):1178–1192, 2013.

[21] K. Yu, X. Wu, W. Ding, and J. Pei. Towards scalable and accurate online feature selection for big data. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 660–669. IEEE, 2014.

[22] K. Yu, X. Wu, Z. Zhang, Y. Mu, H. Wang, and W. Ding. Markov blanket feature selection with non-faithful data distributions. In *Data Mining (ICDM), 2013 IEEE International Conference on*, pages 857–866. IEEE, 2013.

[23] P. Zhang, Y. Huang, S. Shekhar, and V. Kumar. Exploiting spatial autocorrelation to efficiently process correlation-based similarity queries. In *Advances in Spatial and Temporal Databases*, pages 449–468. 2003.