Causal Structure Learning via Temporal Markov Networks

Aubrey Barnard BARNARD@CS.WISC.EDU
David Page PAGE@BIOSTAT.WISC.EDU
University of Wisconsin–Madison

Abstract

Learning the structure of a dynamic Bayesian network (DBN) is a common way of discovering causal relationships in time series data. However, the combinatorial nature of DBN structure learning limits the accuracy and scalability of DBN modeling. We propose to avoid these limits by learning structure with log-linear temporal Markov networks (TMNs). Using TMNs replaces the combinatorial optimization problem with a continuous, convex one, which can be solved quickly with gradient methods. Furthermore, representing the data in terms of features gives TMNs an advantage in modeling the dynamics of sequences with irregular, sparse, or noisy events. Compared to representative DBN structure learners, TMNs run faster while performing as accurately on synthetic tasks and a real-world task of causal discovery in electronic medical records.

Keywords: causal discovery; graphical model structure learning; log-linear Markov networks; dynamic Bayesian networks; temporal models; adverse drug events; electronic medical records.

1. Introduction

To understand how events unfold, scientists often analyze time series data with dynamic Bayesian networks (DBNs) (Dean and Kanazawa, 1989). Even when conditions are ideal—the data are available with the right time intervals and the right Markov order is selected for the DBN—learning the structure of the relationships between variables is a combinatorial problem. The enormous search space (Robinson, 1973) prevents a complete search, and the non-convexity of the likelihood prevents guarantees about the quality of the solution. Thus, a heuristic or greedy search is frequently employed.

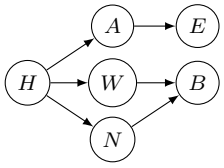
The setting above is the classic search-and-score Bayesian network (BN) structure learning setting as introduced by Cooper and Herskovits (1992). Some algorithms, such as sparse candidate (Friedman et al., 1999), choose to manage the complexity of structure learning by limiting the number of candidate parents to k for each of the n nodes. The result is a subset selection problem that has *combinatorial complexity*, $\sum_{i=1}^{k} \binom{n}{i}$, which is polynomial complexity of order k but tends to exponential complexity (2^n) as $k \to n$. Searching for subsets of size at most k is certainly better than searching over all of the 2^n subsets of the nodes, but it can still be extremely limiting in domains with networks of high degree, such as in biology. Other algorithms, such as K2 (Cooper and Herskovits, 1992) and that of Shojaie and Michailidis (2010), choose to presuppose an ordering of the variables. This requires strong assumptions or background knowledge, or it just exchanges the search over directed acyclic graphs (DAGs) for a search over permutations of variables (Teyssier and Koller, 2005).

Constraint-based methods also suffer from combinatorial complexity. Markov blanket induction (Aliferis et al., 2010), the PC algorithm (Spirtes et al., 2000), and the polynomial min-max skeleton algorithm (Brown et al., 2005) all search for possible separating sets. This is the subset selection problem rederived. In these cases, greedy search over subset members can be used to address the complexity but at the loss of accuracy. Constraint-based methods have additional problems with testing multiple statistical hypotheses and with the possible cascade of errors inherent in their greedy or sequential decision-making processes.

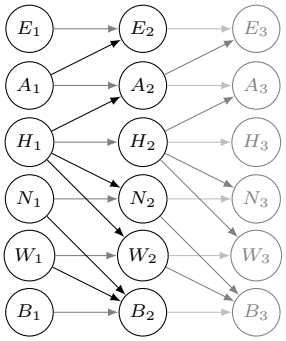
Greedy equivalence search (GES) (Chickering, 2002) may appear to avoid all of these difficulties by guaranteeing to find the optimal equivalence class after only a forward and backward pass (assuming faithfulness and infinite data), but it is still combinatorial. The number of neighboring search states encountered at each step can be exponential because adding or deleting edges involved in V-structures again faces a subset selection problem.

Drug	Condition	Label
A ACE inhibitors	E angioedema	+
T amphotericin B	R acute renal failure	+
I antibiotics	L acute liver failure	+
P antiepileptics	S aplastic anemia	+
Z benzodiazepines	F hip fracture	+
Φ bisphosphonates	U upper GI ulcer	+
D tricyc. antidep.	M acute MI	+
Y typ. antipsycho.	M acute MI	+
W warfarin	B bleeding	+
β beta blockers	X MI mortality	−
N NSAIDs	H hypertension	

(a) OMOP ADE task and list of events (variables)



(b) Causal BN



(c) Equivalent DBN

Figure 1: (a) Causal pairs of the OMOP ADE task. Additional negatives are non-ADE drug–condition pairs among the same events. (b) A real-world causal network involving the variables. (c) Its unrolled DBN.

We propose to avoid the combinatorial nature of these search algorithms by reformulating the structure learning problem as a smooth, convex, non-combinatorial optimization problem in a log-linear model: first use a temporal Markov network (TMN) to learn the undirected skeleton and then direct the edges with time. TMNs provide a way to handle sequences of irregular events (which is important for analyzing medical data), and the optimization jointly estimates all the edges, avoiding issues of multiple testing and sequential decisions. Lee et al. (2006) harness the same optimization benefits, but they only learn an undirected skeleton, one that may be biased by using a L_1-regularizer. Our method is correct and unbiased in the limit of the data (Theorem 5).

1.1 ADE Discovery

Identifying adverse drug events (ADEs) is the causal task that motivates this work. In the USA, ADEs are estimated to be the fourth leading cause of mortality, affecting more than 2 million people each year and incurring $136 billion in additional medical care (U.S. Food and Drug Administration, 2009). To combat this problem and improve patient safety, the Observational Medical Outcomes Partnership (OMOP) led research into drug safety surveillance methods by developing an ADE identification task (Figure 1a) and making electronic medical records (EMR) data sets (Figure 3b) available to researchers in a laboratory.[1]

One of the challenges of identifying ADEs is that it is an inherently causal task, and so requires appropriate methods. Causal methods fall into two broad categories: observational studies (e.g., cohort and case–control studies) and structural causal models (SCMs) (Pearl, 2009; Spirtes et al., 2000), such as causal BNs (Figure 1b). With SCMs, causal discovery becomes a structure learning problem. While most of the work on the OMOP ADE task has focused on observational studies (e.g., Ryan et al., 2012), a contribution of this work is the application of machine learning to the task: learning the structure of causal DBNs (Figure 1c).

The challenges of causal discovery are amplified in the EMR realm where the data is a messy collection of events. Patients interact with the medical system sporadically, on their own initiative, and usually only when they are ill, not when they are well. While EMR data contains thousands of variables describing the state of a patient's health, only a few are recorded at any visit. Thus, observations of a patient are irregular, subject to large time gaps, and very sparse. Furthermore, they are noisy and biased by patient health, by hospital procedure, or by convenience. Of course, EMR data is observational and so also susceptible to confounding.

1. This work now continues under the Innovation in Medical Evidence Development and Surveillance (IMEDS) and Observational Health Data Sciences and Informatics (OHDSI) programs.

Contributions Learning the structure of causal DBNs is difficult due to the combinatorics of deciding which edges to include. This difficulty is worse when learning from EMR data, which is irregular, noisy, and sparse, and thus lacks the regular, full observations needed for DBN learning. Causal structure learning via TMNs addresses these problems by (1) learning the directed structure using an undirected model, wherein the parameters indicate the edges and learning the parameters is a convex optimization problem, and (2) using features to model the irregularity, sparsity, and temporality of EMR data. As far as we are aware, combining structure learning via parameter learning and coarse temporal modeling is novel, and the results show that it is effective for causal structure learning.

2. Background

Related Work While many methods could be used to identify ADEs in longitudinal data—ranging from graphical Granger methods (Arnold et al., 2007) to computational epidemiology (Simpson et al., 2013)—BN structure learning will be the focus here because of its potential to yield a SCM. Algorithms such as PC and fast causal inference (Spirtes et al., 2000) measure conditional independence to detect provably causal structures, but noise can affect independence tests and lead to a cascade of errors. Score-based BN structure learners (e.g., Heckerman et al., 1995) avoid these problems but are not guaranteed to learn a causal structure (although they may do so under certain conditions (Meek, 1997)). Local learners determine the neighborhood or the Markov blanket of each node before stitching them together (Margaritis and Thrun, 1999; Tsamardinos et al., 2003; Niinimäki and Parviainen, 2012). Aliferis et al. (2010) show that these "grow-shrink" algorithms can be sound and complete and therefore causal. A related algorithm learns an undirected skeleton with a local search method and then directs the edges in a greedy hill-climbing search (Brown et al., 2005).

Other, non-causal BN structure learning methods directly address the combinatorial optimization by using dynamic programming (Koivisto and Sood, 2004) or any-time, branch-and-bound search (de Campos et al., 2009). Similar linear programming approaches (Jaakkola et al., 2010; Cussens, 2011) operate in a continuous optimization space, but finding an integral solution to the relaxation may require combinatorial search.

Learning undirected structures over temporal variables, as is possible in the DBN setting where the temporal order of variables is given, opens the door to non-combinatorial structure learning algorithms. The classic example of such an algorithm is selection of Gaussian graphical models, where the zeros in the inverse covariance matrix indicate the absence of edges (Lauritzen, 1996). The same ideas have been developed for discrete variables, including methods for nodewise structure learning using L_1-regularized regression (Loh and Wainwright, 2013). In contrast, our method directly uses the zero parameters to indicate conditional independence, as in Liu and Page (2013) and Lee et al. (2006), but it is unbiased and also addresses the recovery of directed models.

Probabilistic Graphical Models A probabilistic graphical model (PGM) is a model of a probability distribution over a set of random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ that uses a fixed graph G to represent the conditional independence relationships of the distribution. In a PGM, each variable corresponds to a vertex in G.

The structure of a distribution refers to its factorization and conditional independence properties, which are related as follows (Lauritzen, 1996; Koller and Friedman, 2009; Loh and Wainwright, 2013). These statements lay the foundation for Theorem 5, one of our main contributions.

Definition 1 (Factorization property) *A distribution* $P(\mathcal{X})$ factorizes *according to an undirected graph* G *if its density can be expressed as a product of non-negative potential functions on the cliques* C *of* G*.*

$$P(\mathcal{X}) \propto \prod_{c \in C} \psi_c(X_c) \tag{1}$$

Definition 2 (Global Markov property) $X_A \perp\!\!\!\perp X_B \mid X_S$ *if and only if* S *separates* A *from* B *in* G*.*

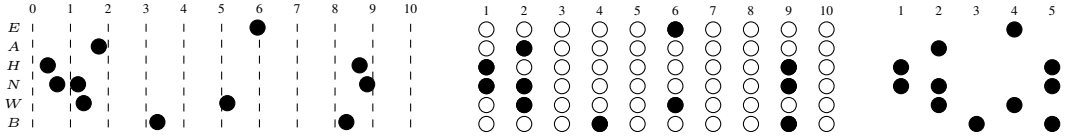
Theorem 3 (Proposition 3.8 (Lauritzen, 1996)) *For any undirected graph* G *and any probability distribution* P *on* $\mathcal{X}$*, it holds that the factorization property implies the global Markov property.*

3. Temporal Markov Networks

This paper introduces temporal Markov networks (TMNs), a type of log-linear PGM with feature functions for modeling timelines. TMNs are motivated by the need for a probabilistic causal model of EMR data, which does not have (1) synchronized timing of a consistently-observed set of events, as assumed by DBNs and other time series methods (e.g., Granger, 1969; Arnold et al., 2007), nor (2) detailed patient state and reliable timing of events, as needed by continuous time Bayesian networks (Nodelman et al., 2002) and piecewise-constant conditional intensity models (Gunawardana et al., 2011).

3.1 Timelines

A timeline (sequence) $\mathcal{S}$ is a set of random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ that occur over a set of times $\mathcal{T}$: $\mathcal{S} = \{X_{i,t} : (X_i, t) \in \mathcal{X} \times \mathcal{T}\}$ as in Figure 2a. Each $X_{i,t}$ is a point event, so a timeline is equivalently a sequence of event tuples (t, X_i, x), where t is the time of occurrence, X_i is the event type, and x is its observed value. This is the form of typical EMR data, with such a sequence for each patient. In this work, we consider only discrete times $\mathcal{T} \subseteq \mathbb{Z}_{0+}$ and binary variables (event occurrences) as in Figure 2b. A condensed timeline (Figure 2c) includes only observed events as a sequence of timesteps. It is constructed by ignoring empty timesteps, discarding durations between events, and treating the remaining timesteps in sequence.



(a) Point events in continuous time (EMR) (b) Fully-observed discrete time (DBN) (c) Condensed

Figure 2: Various forms of a sequence of events (timeline) as might be observed from a process like Figure 1b.

3.2 Log-Linear Model

Definition 4 (Temporal Markov network (TMN)) *A TMN is a tuple $(\mathcal{X}, F, \theta)$, where $\mathcal{X}$ is a set of event types (random variables), F is a set of binary feature functions $f_i(\mathcal{X}_i \subseteq \mathcal{X}) : \mathcal{X}_i \mapsto \{0, 1\}$, and $\theta \in \mathbb{R}^{|F|}$ is set of weights corresponding to the features. A TMN defines a probability distribution over timelines S through the log-linear model in Equations 2 and 3 (e.g., Koller and Friedman, 2009), where $f_i \in F$ and $\theta_i \in \theta$. The features must be (1)* hierarchical*: the variables in each feature define a clique and the cliques induce a graph G; in order to be hierarchical, F must contain a feature for each (sub-)clique in G (Lauritzen, 1996); and (2)* temporal*: F must include at least some features for temporal order or succession (see §3.3).*

$$P(S = s) = \frac{1}{Z} \exp\left(\sum_i \theta_i f_i(s)\right) \qquad (2) \qquad\qquad Z = \sum_{s \in S} \exp\left(\sum_i \theta_i f_i(s)\right) \qquad (3)$$

While being a log-linear model ensures that a TMN always represents a well-defined probability distribution, the additional semantics of a TMN depend on its features, as explained below.

3.3 Feature Functions

The following temporal indicator features model the most salient aspects of timelines as logical predicates. They are designed to capture the main effects of the events and their interactions, both temporal and atemporal. In the notation, S is a timeline, T is a timestep, X, Y, Z are events, uppercase indicates variables, and lowercase indicates instantiated values. When used in a TMN, the features are instantiated $(f_S(\cdot) \mapsto f_i(S))$ for each non-redundant combination of events and times. For example, co-occurrence ignores order, so

$f_S(w, b)$ and $f_S(b, w)$ are redundant, but $f_S(w \to b)$ and $f_S(b \to w)$ are ordered, so they are not redundant. Note that both $f_S(w \to b)$ and $f_S(b \to w)$ can be true of the same sequence as shown in Figure 2.

- *event*, $f_S(x)$: true if event x occurs in S (atemporal)
- *event@*, $f_S(x_t)$: true if event x occurs at t in S (atemporal)
- *co-occur*, $f_S(x, y)$: true if events x and y occur in S (atemporal)
- *co-occur@*, $f_S(x_{t_1}, y_{t_2})$: true if x occurs at t_1 and y occurs at t_2 in S (temporal)
- *before*, $f_S(x \to y)$: true if x and y occur in S and x occurs before y (temporal)
- *before-δ*, $f_S(x_T \to y_{T+\delta})$: true if x and y occur in S and x occurs δ timesteps before y (temporal)
- *before3*, $f_S(\{x, y\} \to z)$: true if x, y, and z occur in S and both x and y occur before z (temporal)

The "@" features are anchored to specific timesteps, but the other features float. Floating, being less specific than anchoring, ties parameters across timesteps and makes an assumption of stationarity. All the floating features except *before-δ* span any number of timesteps, allowing them to capture short- and long-range effects. The *before3* feature exists to model temporal V-structures.

Depending on the choice of features and the parameter tying they induce, TMNs can represent undirected analogs of BNs, DBNs, and event networks (Arroyo-Figueroa and Sucar, 1999; Galán and Díez, 2002), and the semantics of a TMN follow those of the analogous model. Examples of TMNs that imitate BNs and DBNs are in §4.

3.4 Parameter Learning

The parameters are learned using standard maximum likelihood estimation. Finding the maximum of the log-likelihood is a continuous, convex optimization problem, which can be solved by gradient ascent. Because the maximum of the log-likelihood is global, it is reached when the gradient (Equation 4) is zero (e.g., Koller and Friedman, 2009).

$$\frac{\partial}{\partial \theta_i} \frac{1}{|D|} \log \mathcal{L}(\theta; D) = \mathbb{E}_D(f_i(s)) - \mathbb{E}_\theta(f_i(s)) \tag{4}$$

To compute the gradient, the expected statistics of the data ($\mathbb{E}_D$) must first be computed, but this needs to be done only once. Then, the expected statistics given the TMN ($\mathbb{E}_\theta$) must be computed, and this must be done every time the parameters change. Doing so requires inference, but inference is difficult because the graph structure defined by the features is a single clique, and hence not amenable to inference algorithms for factor graphs. This limits the inference options to sampling or, for small problems, exact inference. We chose exact inference for the sake of precision, and implemented our TMNs in Julia using L-BFGS optimization.

3.5 Causal Structure Learning via Parameter Learning

TMNs are used to learn the directed structure of a distribution of timelines by (1) detecting conditional independence between variables, (2) including only those edges that correspond to direct dependences, and (3) directing edges with time. Detecting conditional independence is done by constructing a TMN, learning the weights of its features, and comparing those weights to zero. A weight that is zero indicates the absence of the relationship modeled by that feature, and if all the weights of all the features involving a pair of variables are zero, then those variables are conditionally independent. This property allows weight learning in TMNs to recover the conditional independence structure of the generating DBN as shown in the following theorem.

Theorem 5 (TMN Structure Learning) *Given a DBN $\mathcal{M}$ that generates a true distribution $P(\mathcal{S})$ over timelines, the forward edges of the DAG G of $\mathcal{M}$ can be deduced from the weights of a TMN fit to $P(\mathcal{S})$ using maximum likelihood. Specifically, if the weights of $f_i(X \to Y)$ and all the other features containing X and Y are zero, then $X \to Y$ is not an edge in G:*

$$(\forall (i : f_i \supseteq \{X, Y\})\ \theta_i = 0) \implies X \to Y \notin G \tag{5}$$

Proof If all of the weights of features involving X and Y are zero, then those weights contribute nothing to the sum in Equation 2 and hence contribute nothing to the product in Equation 1. Since the factorization of P does not include X and Y, they must be independent by Theorem 3, and there cannot be an edge between X and Y in any graphical model consistent with P. ■

A TMN can only capture the undirected version of the generating DBN, but if it is a first-order, non-isochronal DBN (it has no edges within a timestep that represent instantaneous relationships) (Plis et al., 2015), then moralizing it adds no edges between timesteps, and the forward edges indicated by the TMN weights are exactly the edges of the DBN.

In summary, Theorem 5 shows how weight learning in a TMN can recover the DBN structure given the true distribution of timelines: include only those edges $x \to y$ that correspond to features $f_i(x \to y)$ (or $f_i(\{x, z\} \to y)$, etc.) with nonzero weights θ_i. The edges are already directed with time.

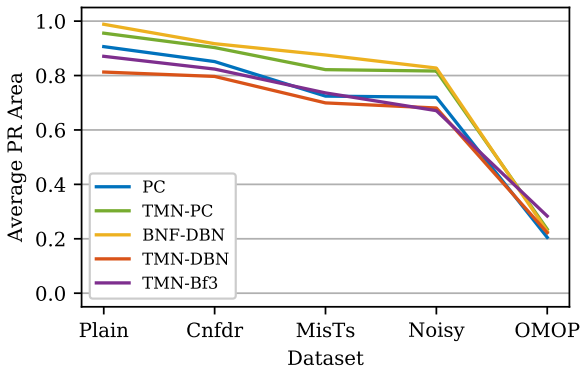
Is such a structure a causal model? If one assumes the causal Markov and causal faithfulness conditions (Spirtes et al., 2000), as is commonly done, then a DBN that has the correct independence structure is a causal DBN. Theorem 5 shows that, given the true distribution of a first-order, non-isochronal DBN, the weights of the learned TMN will indicate the correct independences and therefore describe a causal structure.

Of course, in practice the distribution is not the population one but an empirical one. The noise in such a sample alters the learned weights and obscures the independences. Thus, it becomes necessary to employ regularization or a threshold to determine the zeros. Regularization introduces bias, so we chose to threshold the weight magnitudes. This approach provides an unbiased estimator that has a straightforward interpretation: as soon as the magnitude of the noise gets larger than the magnitude of the signal, the thresholding will start to get edges wrong. This can be mitigated by choosing features that are expressive enough to accurately model the distribution, that can diffuse or absorb the noise, and that isolate relationships of interest. For example, one could use only $f(x \to y)$ and $f(y \to x)$ to model a temporal relationship, but also including $f(x, y)$ isolates their atemporal co-occurrence from their temporal precedence and splits the noise accordingly. Choosing features that are expressive enough to accurately model the distribution means choosing features that match the level of interactions (cliques) in the underlying process. At one extreme, the saturated model (e.g., Wasserman, 2004, §19.4) makes no assumptions about independence or the level of interactions, but is intractable due to the number of features involved. (The number of features in the saturated model is $2^n - 1$ for n binary variables, and there are $|\mathcal{X}||\mathcal{T}|$ binary variables.) At the other extreme, one can assume only pairwise interactions, but this will almost certainly lead to inaccurate weights and an incorrect ranking of edges. Note that using only pairwise features is similar to making an assumption that the conditional probability distributions of the underlying process are noisy-ORs.

4. Experiments

To evaluate TMNs, experiments were conducted to compare them to other methods on DBN structure learning tasks using synthetic and real-world data. The experiments were designed to measure how accurately the methods could recover the structure of dynamic causal networks in a variety of scenarios. With the synthetic data, the methods sought to recover known, complete causal networks having observed all the relevant variables. With the real-world EMR data, they sought to recover the causal structure among the variables in Figure 1a having observed only those same variables. This is the OMOP ADE task, which involves only a small, known subset of the causal structure in EMR data.

For comparison methods, we chose the PC algorithm and BNFinder to represent the two major BN structure learning paradigms. The causal, constraint-based paradigm was represented by the PC algorithm (Spirtes et al., 2000). It only works for static data, but it was applied to timelines by using separate variables for each timestep, unrolling the model as in Figure 1c, and by reversing edges that went backwards in time. The comparison TMN, TMN-PC, equivalently used anchored features ($f(x_t)$, $f(x_{t_1}, y_{t_2})$). The score-based paradigm was represented by BNFinder (Wilczyński and Dojer, 2009; Dojer, 2006). It finds the optimal-scoring BN structure in polynomial time given a partial order of the variables and a maximum number of parents. Being



(a) Average PR areas of the methods across data regimes

Name	People	PoI	EoI	Years
GE	11.2M	4.1M	7.1M	1995–2009
CCAE	46.5M	25.6M	47.7M	2003–2009
MDCD	10.8M	7.3M	14.0M	2002–2007
MDCR	4.6M	3.9M	12.7M	2003–2009
MSLR	1.2M	1.1M	2.1M	2003–2008

(b) GE Centricity (EMR), MarketScan Commercial Claims and Encounters (claims), MarketScan Medicaid (claims), MarketScan Medicare (claims), MarketScan Lab (claims). EoI: events of interest. PoI: people with EoI.

Figure 3: (a) Summary of results. (b) Summary statistics of the five OMOP data sets.

optimal, BNFinder subsumes GES (Meek, 1997; Chickering, 2002) and other score-based structure learners on the task of learning DBNs. The comparison TMN, TMN-DBN, used features to represent the initial and transition distributions of a first-order DBN ($f(x_{t=0})$, $f(x_{t=0}, y_{t=0})$, $f(x)$, $f(x_T, y_T)$, $f(x_T \rightarrow y_{T+1})$). A third TMN, TMN-Bf3, extended the TMN-DBN approach with long-range temporal features and three-way interactions ($f(x_{t=0})$, $f(x_{t=0}, y_{t=0})$, $f(x)$, $f(x, y)$, $f(x \rightarrow y)$, $f(\{x, z\} \rightarrow y)$). While higher-order interactions would be necessary to represent distributions in general, tuning indicated three-way features were sufficiently rich.

Both the synthetic and real-world experiments shared the same setup and analysis. The data was timelines of events (§3.1). Based on the timelines, the methods scored each possible forward edge in a first-order DBN to produce a weighted, bipartite graph. The edge score was the weight magnitude of the corresponding temporal feature for TMNs, aggregate posterior edge probability for BNF-DBN, and edge existence $\{0, 1\}$ for PC. The weighted graphs were evaluated as (soft) binary classification tasks: which of the edges belong to the true DBN graph. To do this, the edges were ranked by their score, and then classification accuracy was assessed with precision-recall (PR) analysis because class skew (edge density of the true graph) varied widely. The methods were developed and tuned using a separate set of hand-crafted and randomly-generated test cases prior to running any experiments. The specific parameters are in the supplement.

4.1 Synthetic DBN Experiments

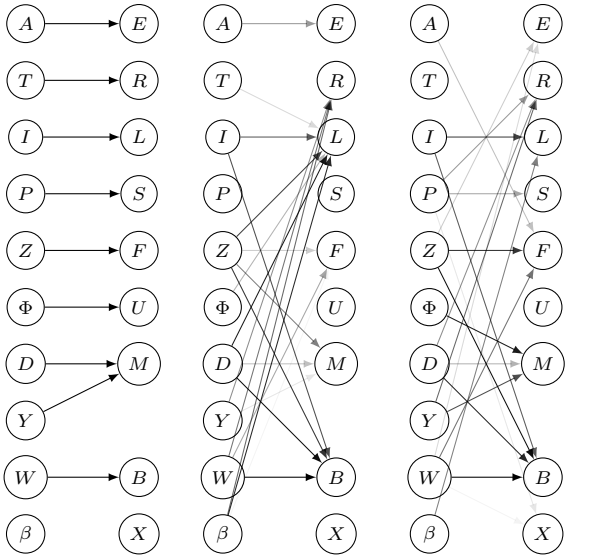
In the synthetic data experiments, the goal was to recover the structure of 1k random DBNs given data sets of 10k timelines sampled from each DBN. The first 100, 1k, and 10k timelines of each data set were used to assess statistical efficiency. Each data set received four data treatments designed to test the methods in the face of noise, missing timesteps, and confounding. For each of the four data treatments the data was represented in two ways: fully-observed and condensed, as illustrated in Figures 2b and 2c. The condensed data imitates real EMR data where negatives are typically not recorded and absolute times are not reliable, but it also simplifies the problem of modeling events that occur over widely-varying time scales. The details of the DBN generation, data generation, and data treatments are in the supplement.

To assess how well the methods recovered DBNs from the synthetic data, the PR areas of their structure recovery were compared. Figure 3a shows the average PR area achieved by each method on each data regime. Overall, BNF-DBN scored the best on average, followed by TMN-PC, PC, TMN-Bf3, and TMN-DBN. We believe that BNF-DBN did so well because its assumptions exactly match the data generating model.

Behind the averages in Figure 3a, the performance of the methods varied substantially by data regime and other characteristics of the DBN structure learning problems. To assess the influence of these characteristics on the achieved PR areas, a linear regression was performed using PR area as the dependent variable and method, data regime, data size, etc. as the independent variables. Selected results are in Figure 4a and the

	X	$\hat{\beta}$	P-Value
1	BNF-DBN? * cnfdr	-0.713	0
2	density, e/n^2	0.663	0
6	TMN-Bf3? * cnfdr	-0.454	3.16e-232
7	BNF-DBN? * noise	-0.414	0
8	BNF-DBN?	0.400	0
9	TMN-PC?	0.309	0
10	BNF-DBN? * mists	-0.305	0
15	PC?	0.245	0
16	TMN-DBN? * noise	-0.225	0
17	TMN-DBN?	0.217	0
19	TMN-Bf3?	0.209	0
21	TMN-Bf3? * mists	-0.157	1.26e-186
22	# cnfdr $/n$	0.130	3.99e-39
23	log # data	0.0747	0
24	missingness	-0.0227	2.16e-09
25	noise level	-0.0213	3.79e-08
32	condensed?	0.000733	0.296

(a) Selected results from a linear regression of PR areas



(b) Ground truth (c) PC, MSLR (d) TMN-Bf3, MSLR

Figure 4: (a) Selected results from a linear regression of PR areas on attributes of synthetic DBN experiments, ranked by $\hat{\beta}$ magnitude. $R^2 = 0.699$. The method indicators contrast with random guessing. (b–d) Ground truth and selected learned networks for the OMOP task showing the top 20 edges from an ensemble of that method's MSLR runs. Figure 1a lists the variables.

full list of coefficients is in Table **??** in the supplement. Each coefficient is interpreted as the change in PR area attributable to a unit change in X, everything else held constant. They show that BNF-DBN suffered the most in the face of confounding, noise, and missing timesteps, and that TMN-DBN was the most robust to noise while TMN-Bf3 was the most robust to confounding and missing timesteps. Data size was important but condensing the data had almost no effect. Of the data treatments, missingness was the least detrimental of the three in terms of its interactions with the methods. These results suggest that treating missing data as false and condensing it is reasonable to do with EMR data (where the majority of data is not observed).

Perhaps counterintuitively, increasing the network density or increasing the number of confounders helps performance. In the case of confounders, hiding variables removes them from the problem, leaving a smaller, easier problem. In the case of density, having more of the possible edges be true reduces the chance that misranking a single edge will affect the PR area.

4.2 OMOP Experiments

In the OMOP experiments, the goal was to discover ADEs in real-world EMR and claims databases. This was formulated as a DBN structure learning task rather than a causal effect size estimation task as is the case with many other methods for causal discovery. The DBN structure learning task was based on the OMOP ADE task, which defines 9 true ADEs and 44 non-ADEs among the same events (Figure 1a). OMOP selected these positives and negatives based on drug labeling and evidence in the literature. For our purposes, the positives defined the edges of the ground truth graph (Figure 4b). The methods learned DBNs over all of the drugs and conditions, but only edges corresponding to pairs in the OMOP task were used in the evaluation.

The five methods learned DBNs from data sets of timelines extracted from the five OMOP databases (Figure 3b). The OMOP databases contained dated event tuples (§3.1), which can be viewed as timelines discretized by day (Figure 2a). To create a data set from each database, a timeline for each patient was extracted and then condensed as in Figure 2c. Variables not observed were assumed to be false. Twenty samples of 100k timelines were drawn without replacement from each data set. These replicates were drawn

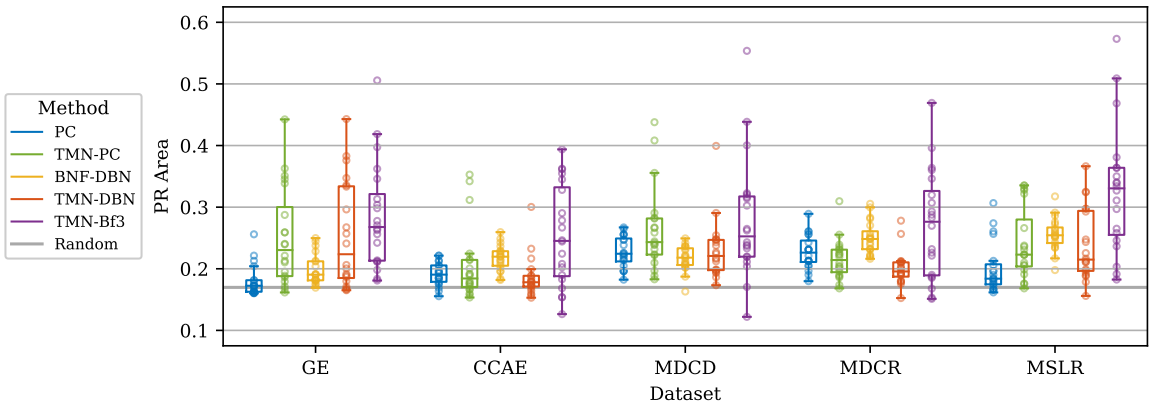


Figure 5: Distributions of PR areas from the 20 replicates drawn from each OMOP data set.

because PC and BNF-DBN could not scale to the full data size. (TMNs, needing only sufficient statistics, have no direct data size limitations.)

Figure 5 shows the results of the experiments on the OMOP data sets in terms of PR area distributions of replicates. The TMNs do especially well on the GE EMR data, but the performance on the claims data is mixed. Looking at the medians, TMN-PC beats PC on 3 data sets, TMN-DBN beats BNF-DBN on 2 data sets, and TMN-Bf3 is the best on all 5 data sets. The significance results in Table **??** in the supplement lead to a similar ranking of the methods by wins in a pairwise tournament: TMN-Bf3, TMN-PC, BNF-DBN, TMN-DBN, PC. We hypothesize that the success of TMN-Bf3 on the OMOP task is due to its ability to effectively model higher-order interactions and detect independence in the presence of noise.

The (min, avg, max) run times, in hours, on the OMOP task were BNF-DBN (0.6, 0.8, 0.9), TMNs (0.5, 1.3, 2.8), and PC (0.1, 2.9, 9.9). While this makes BNF-DBN look fast, the experiments had to be limited to 100k timelines to make BNF-DBN and PC tractable (whereas the TMNs were able to run on the millions of timelines in the full-size OMOP data sets (Figure 3b)). Furthermore, TMN weight learning could run faster by stopping as soon as the ranking of weights is settled (because PR area only depends on the ranking of edges), but this was not implemented. Perhaps this explains why TMNs were successful even though in many cases they did not converge within their allotted 1000 iterations. On the other hand, the lack of convergence is likely a large factor in the variation of the TMN results.

The success of BNF-DBN on both the synthetic and OMOP tasks demonstrates that DBNs may be applicable to modeling EMR data despite its sparseness and irregularity, and suggests that condensed data may also work for other discrete-time models that assume fully-observed, regularly-sampled data.

In a qualitative view of performance, additive ensembles of the networks learned by PC and TMN-Bf3 on MSLR are shown in Figure 4 along with ground truth. Both methods have six correct edges among the top 20, but TMN-Bf3's six are higher in its ranking. (The other methods have four or fewer correct edges in the top 20.) PC and TMN-Bf3 agree on four correct edges. PC concentrates many relationships on renal and liver failure, while TMN-Bf3 spreads out its edges more evenly. These results demonstrate that causal structure learning methods are applicable and relevant to problems in epidemiology despite not estimating effect sizes.

5. Discussion

There are many advantages to treating structure learning as a smooth, convex optimization problem rather than a combinatorial one. Convexity guarantees that there is a global optimum and that there are no impediments to getting there, like plateaus or local optima. This guarantees progress with every iteration, and the optimization can be stopped at any time to yield an approximate solution with the gradient giving a sense of how close the current model is to the optimum. Furthermore, the optimization focuses first on the most

important features, which are those with the largest gradients. Framing the problem as an optimization means that all the edges are estimated jointly, avoiding sequential decisions and multiple testing. This framing also removes the need for greedy or heuristic search as the optimization space is tractable and amenable to well-understood approximation (e.g. stochastic gradient descent). All these advantages combine to make our approach faster, more robust, and better able to handle noise than approaches based on combinatorial search.

The formulation as a log-linear model also comes with advantages and disadvantages. In terms of advantages, it allows arbitrary features, which can be used to handle irregular events and model short- and long-range dependencies. The data can be completely summarized by the sufficient statistics of the features, which allows scaling to very large data sets by separating the data processing from the optimization. By comparison, updating a BN structure score requires a pass over the data even if it only involves a few of the variables. The sufficient statistics, being aggregates, are also robust to noise. In terms of disadvantages, there is now a modeling problem as one must choose the right features. Part of this relates to choosing the level of interactions that the features can express. Depending on how many features are chosen, their complexity, and how many combinations of events they are instantiated for, there can be a very large number of features and a correspondingly large optimization space, which may be challenging for optimization algorithms. The optimization challenges are amplified by the inference difficulties of an extremely large, unfactorable PGM.

Unfortunately, due to the inability of undirected PGMs to express the independence in a V-structure, exact recovery of DBN edges by TMNs is limited to first-order, non-isochronal DBNs. However, assuming a first-order DBN is relatively innocuous because any higher-order DBN can be converted to an equivalent first-order DBN. Assuming a non-isochronal DBN is reasonable in cases where the timescale of the DBN is smaller than that of the system (Plis et al., 2015). This is the case for EMR data where data is available on the same scale as disease progression in both inpatient and outpatient settings.

This work represents an alternative approach to the OMOP task, one that uses structure learning instead of causal effect estimation (as would be done in epidemiology). Structure learning and effect estimation are not directly comparable because they handle direct and indirect effects differently. Effect estimation doesn't care about the path, only its overall effect, whereas structure learning cares only about the direct effects that make up the path, not its overall effect. Unfortunately, this mismatch means that the OMOP task is not necessarily a suitable evaluation for structure learning methods; it depends on how many of the OMOP pairs are direct effects in terms of the observable variables in the data. Investigating this and determining how to better apply structure learning to epidemiological tasks is ongoing work.

6. Conclusion

In learning the relationships among events, TMNs avoid the combinatorial nature of classical BN structure learning algorithms by reformulating structure learning as a smooth, convex optimization problem in a log-linear model. As shown in Theorem 5, TMNs learn the correct structure given enough data and sufficiently expressive features, and the learned structure corresponds to a causal DBN. This enables TMNs to do causal discovery, and their flexible, expressive features enable them to handle the irregularity, sparsity, and noise of EMR data. Therefore, TMNs have the characteristics necessary to address the challenges of the OMOP ADE task. In practice, they demonstrate their effectiveness by performing as well or better than representative methods for DBN structure learning. Thus, with characteristics and performance that complement existing methods, TMNs establish an alternative to DBNs for causal discovery from observational time series data.

Acknowledgments

The authors would like to gratefully acknowledge the Center for Predictive Phenotyping (NIH BD2K Initiative grant U54 AI117924 and NIGMS grant 2RO1 GM097618) for supporting this work.

References

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(Jan), 2010.

A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical Granger methods. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 13*, 2007.

G. Arroyo-Figueroa and L. E. Sucar. A temporal Bayesian network for diagnosis and prediction. In *Uncertainty in Artificial Intelligence 15*, 1999.

L. E. Brown, I. Tsamardinos, and C. F. Aliferis. A comparison of novel and state-of-the-art polynomial Bayesian network learning algorithms. In *AAAI Conference on Artificial Intelligence 20*, 2005.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov), 2002.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4), 1992. doi: 10.1023/A:1022649401552.

J. Cussens. Bayesian network learning with cutting planes. In *Uncertainty in Artificial Intelligence 27*, 2011.

C. P. de Campos, Z. Zeng, and Q. Ji. Structure learning of Bayesian networks using constraints. In *International Conference on Machine Learning 26*, 2009.

T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3), 1989.

N. Dojer. Learning Bayesian networks does not have to be NP-hard. In *Mathematical Foundations of Computer Science 2006*, 2006. doi: 10.1007/11821069_27.

N. Friedman, I. Nachman, and D. Pe'er. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In *Uncertainty in Artificial Intelligence 15*, 1999.

S. F. Galán and F. J. Díez. Networks of probabilistic events in discrete time. *International Journal of Approximate Reasoning*, 30(3), 2002. doi: 10.1016/S0888-613X(02)00071-3.

C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3), 1969.

A. Gunawardana, C. Meek, and P. Xu. A model for temporal dependencies in event streams. In *Advances in Neural Information Processing Systems 24*, 2011.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3), 1995. doi: 10.1007/BF00994016.

T. Jaakkola, D. Sontag, A. Globerson, and M. Meila. Learning Bayesian network structure using LP relaxations. In *Artificial Intelligence and Statistics 13*, 2010.

M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5(May), 2004.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, Massachusetts, 2009.

S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

S. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using l_1-regularization. In *Advances in Neural Information Processing Systems 19*, 2006.

J. Liu and D. Page. Structure learning of undirected graphical models with contrastive divergence. In *ICML Workshop on Structured Learning: Inferring Graphs from Structured and Unstructured Inputs*, 2013.

P.-L. Loh and M. J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics*, 41(6), 2013. doi: 10.1214/13-AOS1162.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12*, 1999.

C. Meek. *Graphical Models: Selecting Causal and Statistical Models*. PhD thesis, Carnegie Mellon University, 1997.

T. Niinimäki and P. Parviainen. Local structure discovery in Bayesian networks. In *Uncertainty in Artificial Intelligence 28*, 2012.

U. Nodelman, C. R. Shelton, and D. Koller. Continuous time Bayesian networks. In *Uncertainty in Artificial Intelligence 18*, 2002.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2nd edition, 2009.

S. Plis, D. Danks, C. Freeman, and V. Calhoun. Rate-agnostic (causal) structure learning. In *Advances in Neural Information Processing Systems 28*, 2015.

R. W. Robinson. Counting labeled acyclic digraphs. In F. Harary, editor, *New Directions in the Theory of Graphs*. Academic Press, New York, 1973.

P. B. Ryan, D. Madigan, P. E. Stang, J. M. Overhage, J. A. Racoosin, and A. G. Hartzema. Empirical assessment of methods for risk identification in healthcare data: Results from the experiments of the Observational Medical Outcomes Partnership. *Statistics in Medicine*, 31(30), 2012. doi: 10.1002/sim.5620.

A. Shojaie and G. Michailidis. Penalized likelihood methods for estimation of sparse high dimensional directed acyclic graphs. *Biometrika*, 97(3), 2010. doi: 10.1093/biomet/asq038.

S. E. Simpson, D. Madigan, I. Zorych, M. J. Schuemie, P. B. Ryan, and M. A. Suchard. Multiple self-controlled case series for large-scale longitudinal observational databases. *Biometrics*, 69(4), 2013. doi: 10.1111/biom.12078.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, Massachusetts, 2nd edition, 2000.

M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Uncertainty in Artificial Intelligence 21*, 2005.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for large scale Markov blanket discovery. In *International Florida Artificial Intelligence Research Society 16*, 2003.

U.S. Food and Drug Administration. Preventable adverse drug reactions: A focus on drug interactions. http:// www.fda.gov/ drugs/ developmentapprovalprocess/ developmentresources/ druginteractionslabeling/ ucm110632.htm, 2009. Accessed September 12, 2016.

L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, New York, 2004.

B. Wilczyński and N. Dojer. BNFinder: Exact and efficient method for learning Bayesian networks. *Bioinformatics*, 25(2), 2009. doi: 10.1093/bioinformatics/btn505.