# Online Learning of Time-varying Causal Structures

**Erich Kummerfeld**[1] **and David Danks**[1,2]
[1]Carnegie Mellon University and [2]Institute for Human & Machine Cognition
Pittsburgh, PA 15213
{ekummerf, ddanks}@andrew.cmu.edu

## Abstract

Causal structure learning algorithms have focused almost exclusively on learning in "stable" environments in which the underlying causal structure does not change. Such changes often occur, however, without warning or signal in real-world environments. In this paper, we present *DOCL*, a novel causal structure learning algorithm that processes data in a dynamic, real-time manner and tracks changes in the generating causal structure or parameters. The algorithm learns in an online fashion from sequential or ordered data rather than "batch-mode" from a full dataset, and so supports causal learning in memory-limited settings. We show by simulation that the algorithm performs comparably to batch-mode learning when the causal structure is stationary, and significantly better in non-stationary environments.

## 1 INTRODUCTION

Over the past twenty years, a wide array of causal structure learning algorithms have been developed and successfully applied in many different domains (Pearl, 2000; Spirtes et al., 2000; Chickering, 2002). All of these algorithms share a crucial feature: they all assume that the underlying causal structure does not change over the course of data collection. Standard causal learning algorithms assume all data are i.i.d. (or can be transformed into i.i.d. data); a dataset in which the causal structure changes partway through is clearly not i.i.d. In the real world, however, causal structures often change: the '+' key on a calculator breaks, or a wire comes loose in one's car, or medication alters the effect of some disease. In all of these cases, it is critical that we quickly detect the causal structure change and then learn the new causal structure.

In many contexts with possible causal structure change, we also do not have the luxury of collecting large amounts of data and then retrospectively determining when (if ever) the structure changed. Unlike all standard causal learning algorithms, we cannot operate in "batch mode," but must instead learn the causal structure in an online manner, processing the data as it arrives.[1] A range of online learning algorithms have been developed to detect and handle changes in the learning environment, but none are capable of *causal* learning, as they have principally focused on tracking the state of a system or variable over time.

In this paper, we develop and assess the *D*ynamic *O*nline *C*ausal *L*earning (DOCL) algorithm — a causal structure learning algorithm that is robust to the possibility that the causal structure varies over time. More precisely, the algorithm assumes only that our data are sets of (locally) i.i.d. datapoints, rather than assuming that all datapoints are i.i.d. In addition, DOCL is an online algorithm, so is suitable for computational devices that are incapable of storing all of the data (e.g., a sensor in a network, particularly if it is multi-channel). In the next section, we quickly survey related methods and show that they are individually insufficient for the task of online causal structure learning. We then explain the DOCL algorithm in detail, and present simulation evidence that it can successfully learn causal structure in an online manner. Importantly, when there is a stable causal structure generating the datapoints, the performance of the DOCL algorithm is indistinguishable from a standard batch-mode causal structure learning algorithm. That is, there is no cost to using DOCL in "normal" causal learning situations. We close with a discussion of future directions for the DOCL algorithm.

## 2 RELATED WORK

Essentially all current causal structure learning algorithms output causal Bayesian networks or causal Structural Equation Models (SEMs): directed acyclic graphs over ran-

[1] Many "batch mode" algorithms require only a covariance matrix and sample size as input, and so could (in theory) be converted to online algorithms. In practice, however, these covariance matrices are almost always estimated in batch from the full dataset.

dom variables with corresponding quantitative components (Chickering, 2002; Heckerman et al., 1999; Spirtes et al., 2000; Pearl, 2000). For simplicity, we will use the term "causal models" throughout to refer to both types of structures. Causal model learning algorithms divide roughly into two distinct types: Bayesian/score-based procedures and constraint-based ones.

Bayesian learning algorithms aim to find the causal model $M$ that maximizes $P(M|Data)$. In practice, one typically has uniform priors over the possible causal models, and assumes that the parameters are independent (in a precise technical sense), and so each causal model can be scored using a decomposable measure based on $P(Data|M)$ and the number of parameters in $M$ (Chickering, 2002; Heckerman et al., 1999). Because the number of possible causal models is super-exponential in the number of variables, however, it is usually impossible to perform an exhaustive search of all possible causal models. Instead, one uses a greedy procedure that starts with a seed graph, scores nearby neighbors, moves to the highest-scoring neighbor, and iterates until no higher-scoring causal model can be found. If the greedy search is done properly, then it is asymptotically reliable (Chickering, 2002). Outside of the causal learning context, Bayesian learning algorithms—or various approximations to them—are frequently used for online learning, since case-by-case Bayesian updating yields the same output as batch-mode processing (assuming the data are i.i.d.). In the situations of interest here, however, the underlying causal structure can change, and so we do not necessarily *want* to have the same output as a batch-mode algorithm.

A different approach to learning causal model structure is to leverage the fact that every causal model—more precisely, every causal graph—predicts a pattern of (conditional) independencies over the variables, though multiple causal models can predict the same pattern. Constraint-based algorithms (e.g., (Pearl, 2000; Spirtes et al., 2000)) determine a minimal set of (conditional) independencies in the data, and then find the set of causal models that best predict those (conditional) independencies. All existing constraint-based algorithms use traditional null hypothesis statistical tests to determine independencies from data, but there is no requirement that they do so. Properly speaking, constraint-based algorithms require only that some method be available that can provide the (conditional) independencies, which could be null hypothesis statistical tests, or Bayesian statistical tests, or some other method.

The key point for the present research is that both types of causal structure learning algorithms assume that the data come from a single generating causal structure, and so neither type is directly usable for learning when causal structure change is possible. Both types require only the sufficient statistics as input data, and so can (and do, in DOCL) provide a part of the solution. But neither has any mechanism for detecting change, responding to it, or learning the new causal structure.

If one turns instead to online learning methods that can track some feature in an environment, one finds that neither of the standard types of methods — temporal-difference learning (TDL) and change-point detection (CPD) — tracks causal structure, and both require substantial modifications to be suitable for it.

The classic TDL algorithm, TD(0) (Sutton, 1988), provides a dynamic estimate $E_t(X)$ of a univariate random variable $X$ using a simple update rule: $E_{t+1}(X) \leftarrow E_t(X) + \alpha(X_t - E_t(X))$, where $X_t$ is the value of $X$ at time $t$. That is, one updates the estimate by $\alpha$ times the error in the current estimate. The static $\alpha$ parameter encodes the learning rate, and must be chosen quite carefully (or somehow learned from the data) to optimally trade-off convergence rate and robustness to noise (in stable environments). This latter property is a particular concern for causal structure learning, since causal structures frequently have indeterministic causal connections. In general, TDL methods are good at tracking slow-moving changes in the environment, but perform suboptimally during times of either high stability or dramatic change.

Both Bayesian (Adams and MacKay, 2007) and frequentist (Desobry et al., 2005) online CPD algorithms are effective at detecting abrupt changes in the environment that indicate breaks between periods of stability. To do so, however, these algorithms must store substantial portions of the input data; for example, the output of a Bayesian changepoint detector (Adams and MacKay, 2007) is the probability of a changepoint having occurred $r$ timesteps ago, and so the algorithm must store more than $r$ datapoints. Furthermore, CPD algorithms assume a model of the environment that has only abrupt changes separated by periods of stability. Environments that evolve slowly but continuously will have their time-series discretized in seemingly arbitrary fashion, or not at all.

Perhaps the most closely related work has been on the problem on inferring time-indexed graph structures (not necessarily causal graphs) from time-series data (Talih and Hengartner, 2005; Siracusa and Fisher III, 2009), though our approach differs in important ways. (Talih and Hengartner, 2005) allow for the possibility of graph changes by taking an ordered data set and dividing it into a given number of connected blocks of data (which may be empty), each with an associated undirected graph that differs by one edge from its neighbors. In contrast with our work, they focus on a particular type of graph structure change, operate solely in "batch mode," and use undirected graphs instead of causal models. (Siracusa and Fisher III, 2009) is more closely related, as that paper uses a Bayesian approach to find the posterior uncertainty over the possible directed edges at different points in a time-series. Our approach differs by using frequentist methods instead of Bayesian ones (since we would otherwise need to maintain a probability distribution over the superexponential number of causal models), and by being able to operate in real-time on an incoming data stream.

# 3 DYNAMIC ONLINE CAUSAL LEARNING (DOCL) ALGORITHM

Given a set of continuous variables $V$, we assume that there is, at each moment in time, a true underlying causal model over $V$. We restrict the set of causal models to recursive causal Structural Equation Models (SEMs) — a pair $\langle G, \mathbf{F} \rangle$, where $G$ denotes a directed acyclic graph over $V$, and $\mathbf{F}$ is a set of linear equations of the form $V_i = \sum_{V_j \in pa(V_i)} a_{ji} \cdot V_j + \epsilon_i$, where $pa(V_i)$ denotes the variables $V_j \in G$ such that $V_j \rightarrow V_i$, and the $\epsilon_i$ are normally distributed noise/error terms. We assume that the data are, at any point in time, generated independently from the true underlying causal model, though we do not assume that this causal model is stationary through time. Notice that a causal model can change in both structure (i.e., adding, removing, or reorienting edges) and parameters (i.e., changes in $a_{ji}$'s or the $\epsilon_i$ distributions).

At a high level, the DOCL algorithm is separated into three, functionally distinct components.

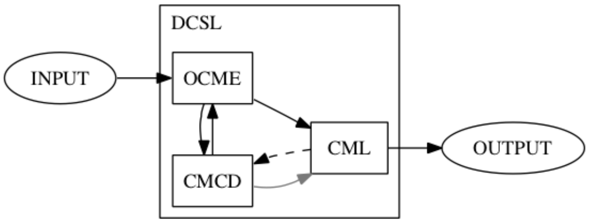


Figure 1: Basic DOCL Architecture

The *O*nline *C*ovariance *M*atrix *E*stimator (OCME) receives each datapoint sequentially as input, and estimates a (possibly non-stationary) covariance matrix to provide the "raw materials" for learning the causal structure. The *C*ausal *M*odel *C*hange *D*etector (CMCD) tracks the divergence between recent datapoints and the estimated covariance matrix to detect changes in the environment, or significant errors in estimation. It then uses that information to adjust the weights on previous datapoints. The *C*ausal *M*odel *L*earner (CML) takes the covariance matrix and learns the causal model at that point in time. The gray arrow from CMCD to CML indicates the use of information about changes to determine when to relearn the causal model. We describe this probabilistic rescheduler below. The dashed arrow from CML to CMCD indicates the possibility of using the current estimated causal model to dynamically (and intelligently) influence the change detection; for example, graphical features that have been present for an extended period of time are plausibly more stable than those that have changed frequently. That feature is currently not part of the DOCL algorithm, but will be added in future research.

## 3.1 ONLINE COVARIATION MATRIX ESTIMATION (OCME)

The OCME module performs the online updating of the sufficient statistics for causal learning from observational data. In particular, OCME maintains an estimated covariance matrix $\mathbf{C}$ over the variables $V$, as well as the "effective sample size" $S$ of the current estimate, and updates $\mathbf{C}$ in response to incoming datapoints. Let $N = |V|$ and $M$ be the total number of datapoints observed. Because OCME does not store any of the incoming datapoints, its memory requirements are only $O(N^2)$ for the estimated covariance matrix, in contrast with $O(NM + N^2)$ memory for batch-mode algorithms. OCME thus has a substantial memory advantage for the common real-world situation of $M \gg N$. OCME can also function as a stand-alone, single-pass covariance matrix estimator for very large datasets.

Let $\mathbf{X}^r$ be the $r$-th multivariate datapoint and let $X_i^r$ be the value of $V_i$ for that datapoint. Because we do not assume a stationary causal model, the datapoints must potentially be weighted differently (e.g., weighting more recent datapoints more heavily after a change occurs). Let $a_r$ be the weight on the $r$-th datapoint (where $a_i \in (0, \infty)$), and let $b_r = \sum_{k=1}^{r} a_k$ be the sum of the weights on each datapoint. The weighted average of $V_i$ after datapoint $r$ is $\mu_i^r = \sum_{k=1}^{r} \frac{a_k}{b_r} X_i^k$. These means can be computed in an online fashion using the update equation:

$$\mu_i^{r+1} = \frac{b_r}{b_{r+1}} \mu_i^r + \frac{a_{r+1}}{b_{r+1}} X_i^{r+1} \tag{1}$$

The (weighted) covariance between $V_i$ and $V_j$ after datapoint $r$ can then be proven to equal $\mathbf{C}_{V_i,V_j}^r = \sum_{k=1}^{r} \frac{a_k}{b_r} (X_i^r - \mu_i^r)(X_j^r - \mu_j^r)$. Because OCME is an online estimation method, we need to translate this into an update rule. The resulting update equation in terms of the current datapoint and the previous $\mathbf{C}$ is:

$$\mathbf{C}_{X_i,X_j}^{r+1} = \frac{1}{b_{r+1}} [b_r \mathbf{C}_{X_i,X_j}^r + \\ b_r \delta_i \delta_j + a_{r+1}(X_i^{r+1} - \mu_i^{r+1})(X_j^{r+1} - \mu_j^{r+1})]$$

where $\delta_i = \mu_i^{r+1} - \mu_i^r = \frac{a_{r+1}}{b_{r+1}} (X_i^{r+1} - \mu_i^r)$. If $a_k = c$ for all $k$ and some constant $c > 0$, then the estimated covariance matrix using this method is identical to the batch-mode estimated covariance matrix. If $a_r = \alpha b_r$, then OCME acts like one is using TD(0) learning for each covariance with a learning rate of $\alpha$.

Because different datapoints can receive different weights, we compute the effective sample size (which should always be less than the *actual* sample size) by adjusting the previous effective sample size based on the new datapoint's relative weight. More precisely, let $S_r$ be the effective sample size at time $r$. We assume the incoming datapoint contributes 1 to the effective sample size, and adjust the *previous* effective sample size accordingly: $S_{r+1} = \frac{a_r}{a_{r+1}} S_r + 1$. Since CMCD ensures that $a_{r+1} \geq a_r$ for all $r$ (see Eq. (2)), we have that $S_{r+1} \leq S_r + 1$. Also, if $a_{r+1} = a_r$ for all $r$, then $S_r = r$; that is, if the datapoint weights are constant, then $S_r$ is the true sample size.

### 3.2 CAUSAL MODEL CHANGE DETECTOR (CMCD)

The CMCD module tracks the "fit" between the current estimated covariance matrix and the input data to detect likely changes in the underlying causal structure, which then require changes in the OCME datapoint weights (i.e., the $a_k$'s). Since the data is assumed to have a multivariate Gaussian distribution, the "fit" between the datapoint $\mathbf{X}^r$ and the current estimated covariance matrix $\mathbf{C}^r$ is given by the Mahalanobis distance $D_r$ (Mahalanobis, 1936):

$$D_r = (\mathbf{X}^r - \vec{\mu})(\mathbf{C}^r)^{-1}(\mathbf{X}^r - \vec{\mu})^T$$

where $\vec{\mu}$ is the current estimate of the means (from Eq. (1)).

A large Mahalanobis distance for any particular datapoint could indicate simply an outlier; several large Mahalanobis distances over multiple datapoints, however, imply that the current estimated covariance matrix is a poor fit to the underlying causal model, and so new datapoints should be weighted more heavily. Our approach is to first calculate the individual $p$-values for each datapoint, and then to use a weighted pooling method to aggregate those $p$-values into a pooled $p$-value.

In general, the Mahalanobis distance of a $v$-dimensional datapoint from a covariance matrix estimated from a sample of size $n$ is distributed $T^2$ with parameters $p = v$ and $m = n - 1$, where $T^2$ is Hotelling's $T$-squared distribution. The $p$-value for the Mahalanobis distance $D_r$ of an individual datapoint at time $r$ is thus:

$$T^2(x > D_r | p = N, m = S_r - 1)$$

where $S_r$ is the effective sample size.

These $p$-values establish the likelihood of $\mathbf{X}^r$ given $\vec{\mu}$ and $\mathbf{C}^r$, but what we really want is the likelihood of the (weighted) data sequence $\mathbf{X}$ given $\vec{\mu}$ and $\mathbf{C}^r$. The distribution of a sum of weighted chi-square variables is analytically intractable, and so we cannot use the $D_r$ values directly. Instead, we use Liptak's method for weighted pooling of the individual $p$-values (Liptak, 1958). Let $\Phi(x, y)$ be the cdf of a Gaussian with mean 0 and variance $y$ evaluated at $x$. Then the pooled, weighted $p$-value is:

$$\rho_r = \Phi(\sum_{i=1}^{r} a_i \Phi^{-1}(p_i, 1), \sqrt{\sum a_i^2})$$

Finally, we need to determine the weight of the next point, $a_{r+1}$ given the pooled $p$-value $\rho_r$. We use a straightforward scheme: given a threshold $T$ to trigger downweighting (i.e., uniformly reducing the weight of all previous datapoints by some constant factor) and a maximum downweighting degree $1/\gamma$, we scale the degree of downweighting linearly from $\rho_r = T$ to $\rho_r = 0$:[2]

$$a_{r+1} = \begin{cases} a_r & \text{if } \rho_r \geq T \\ \frac{a_r \gamma T}{\gamma T + \rho_r - T} & \text{otherwise} \end{cases} \quad (2)$$

The CMCD process can be straightforwardly modified to function as an online changepoint detection method. Specifically, one can treat $\rho_r < \tau$ as a signal (for suitable threshold $\tau$) that a changepoint occurred at time $r$. If one wants to continue detecting changepoints, one need only clear $\mathbf{C}$ and $\vec{\mu}$ and begin the estimation and detection anew.[3]

### 3.3 CAUSAL MODEL LEARNER (CML)

The CML module actually learns the causal model from the estimated (from weighted data) sufficient statistics provided by OCME. In the current implementation, CML uses the PC algorithm (Spirtes et al., 2000), a standard constraint-based causal structure learning algorithm. A range of alternative structure learning algorithms could be used instead, depending on the assumptions one is able to make. Because learning causal models is computationally expensive (Chickering, 1995), one does not want to relearn the graphical structure after each datapoint. Instead, one should use information from the CMCD module to determine when one "ought" to perform these searches by balancing the accuracy of the current learned causal model against the computational cost of relearning the causal model. More precisely, graph relearning should be most common after an underlying change, though there should be a non-zero chance of relearning even when the causal structure appears to be relatively stable.

We have developed a probabilistic relearning scheduler which utilizes the pooled $p$-values calculated by the CMCD module to determine when to relearn the causal graph. The probability of relearning the causal graph after observing datapoint $r$ is determined by: $P_r(relearn) = P_{r-1}(relearn) + \alpha(\rho_r - \rho_r P_{r-1}(relearn))$. That is, the probability of relearning at time $r$ is essentially a noisy-OR gate with the previous probability of relearning, and $\alpha \rho_r$.

Clearly, this quantity is always in $[0, 1]$ when $\alpha \in [0, 1]$. $\alpha$ is a positive real number that modifies the frequency of graph relearning: large values result in more frequent

[2] This potentially misses a specific kind of change, when the only difference in distribution is a move from a high-variance distribution to a low-variance distribution. We are exploring modifications that can accommodate this change as well.

[3] The simulations in sections 4 and 5 use Eq. (2), not this other possibility.

relearning and small values resulting in fewer. If both $\alpha > 0$ and $\rho_r > 0$, then necessarily $P_r(relearn) \geq P_{r-1}(relearn)$. If a relearning event is triggered at datapoint $r$, a new causal graph is generated by the CML, and $P_r(relearn)$ is set to 0. In general, $\rho_r$ is higher when changepoints are detected, so $P_r(relearn)$ will increase more quickly during changepoints, and graph relearning will become more frequent. During times of stability, $\rho_r$ will be comparatively small, resulting in a slower increase of $P_r(relearn)$ and thus less frequent graph relearning.

### 3.4 PROPERTIES OF DOCL

The algorithm as presented is not convergent: that is, even if the underlying causal structure stabilizes in the limit, the algorithm output will not necessarily stabilize. This apparent failure of DOCL is a necessary one, as there is a fundamental tension between convergence and responsiveness. Say that an algorithm is *diligent* if it cannot be forced to miss arbitrarily large changes for arbitrarily long; that is, a diligent algorithm will detect and react to changes of given size within some fixed amount of time (assuming the change is reflected in the data). Both diligence and convergence are desirable methodological virtues, but they are unfortunately incompatible: provably, no learning algorithm can be both diligent and convergent (XXX, 2012).

In this paper, we have presented a diligent version of DOCL, as we have been principally interested in situations in which responsiveness is critical. A convergent version arises if the downweighting threshold parameter $T$ (in Eq. (2)) varies appropriately as a function of effective sample size. Specifically, let DOCL* be the variant in which $T$ (in Eq. (2)) is instead $T_r = f(S_r)$ for some $f$ with $(0, 1]$ range, and where $\sum_{i=1}^{\infty} (1 - f(i))$ is convergent.

**Theorem:** DOCL* converges to the true graph (if it exists) with probability 1 in the infinite limit.

**Proof sketch:** $\sum_{i=r}^{\infty} (1 - q_i)$ can be shown to be an upper bound on the probability that $\rho_i > q_i$ will occur for some $i$ in $[r, \infty)$, where $q_i$ is the $i$-th element of the sequence $Q$ of lower threshold values. Any sequence $Q$ s.t. $\sum_{i=1}^{\infty} (1 - q_i) < 1$ will then guarantee that an infinite amount of unbiased data will be accumulated in the infinite limit. This provides probability 1 convergence for DOCL*, since the CML module has probability 1 convergence in the limit. If $Q$ is prepended with strictly positive threshold values, the first element of $Q$ will still be reached infinitely many times with probability 1 in the infinite limit, and so DOCL* will also converge with probability 1 by using these expanded sequences.

## 4 SIMULATION RESULTS

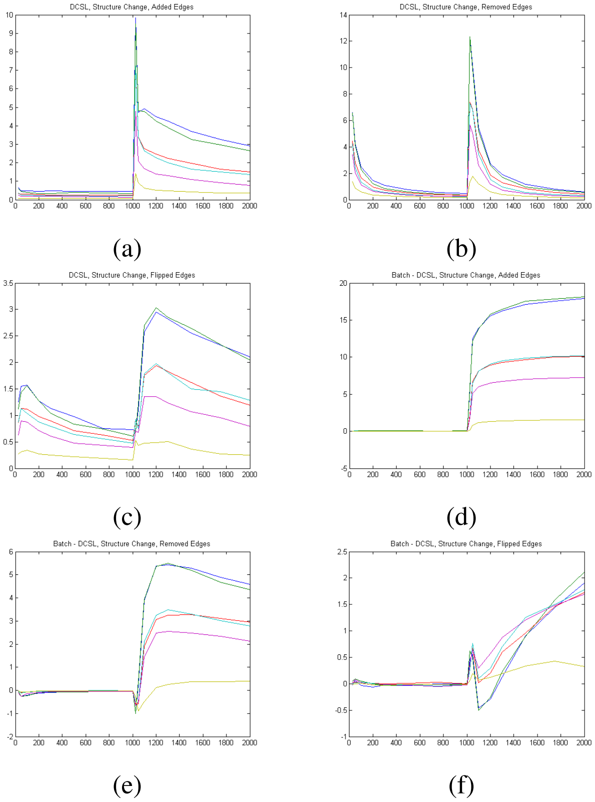
We used synthetic data to evaluate the performance of DOCL given known ground truth. All simulations used scenarios in which either the ground truth parameters or ground truth graph (and parameters) changed during the course of data collection. Before the first changepoint, there should be no significant difference between DOCL and a standard batch-mode learner, since those datapoints are completely i.i.d. Performance on these datapoints thus provides information about the performance cost (if any) of online learning using DOCL, relative to traditional algorithms. After a changepoint, one is interested both in the absolute performance of DOCL (i.e., can it track the changes?) and in its performance relative to a standard batch-mode algorithm (i.e., what performance *gain* does it provide?). We used the PC algorithm (Spirtes et al., 2000) as our baseline batch-mode learning algorithm; for the graphs and sample sizes in our simulations, any standard causal model learning algorithms would perform similarly.

In order to directly compare the performance of DOCL and PC, we imposed a fixed "graph relearning" schedule on DOCL[4]. In the first set of simulations, we used datasets with 2000 datapoints, where the causal SEM graph and parameters both changed after the first 1000 datapoints. We generated 500 datasets for each $\langle \#variables, MaxDegree \rangle$ of $\langle 4, 3 \rangle$, $\langle 8, 3 \rangle$, $\langle 10, 3 \rangle$, $\langle 10, 7 \rangle$, $\langle 15, 4 \rangle$, $\langle 15, 9 \rangle$, $\langle 20, 5 \rangle$, and $\langle 20, 12 \rangle$, where each dataset used two different, randomly generated causal SEMs of the specified size and degree.

Figures 2(a-c) show the mean edge addition, removal, and orientation errors (respectively) by DOCL as a function of time, and Figures 2(d-f) show the mean of $\#errors_{PC} - \#errors_{DOCL}$ for each error type (i.e., higher numbers imply DOCL outperforms PC). In all graphs, each $\langle variable, degree \rangle$ pair is a distinct line. As expected, DOCL was basically indistinguishable from PC for the first 1000 datapoints; the lines in Figures 2(d-f) for that interval are all essentially zero. After the underlying causal SEM changes, however, there are significant differences. The PC algorithm performs quite poorly because the full dataset is essentially a mixture from two different distributions which induces a large number of spurious associations. In contrast, the DOCL algorithm finds large Mahalanobis distances for those datapoints, which lead to higher weights, which lead it to learn (approximately) the new underlying causal model. In practice, DOCL typically stabilized on a new causal model by roughly 250 datapoints after the changepoint.

The second set of simulations was identical to the first (500 runs each for various pairs of variable number and edge degree), except that the graph was held constant throughout and only the causal SEM parameters changed after 1000 datapoints. Figures 3(a-c) and 3(d-f) report, for these sim-

[4] DOCL relearned graphs and PC was rerun after datapoints $\{25, 50, 100, 200, 300, 500, 750, 1000, 1025, 1050, 1100, 1200, 1300, 1500, 1750, 2000\}$.



(a) (b)

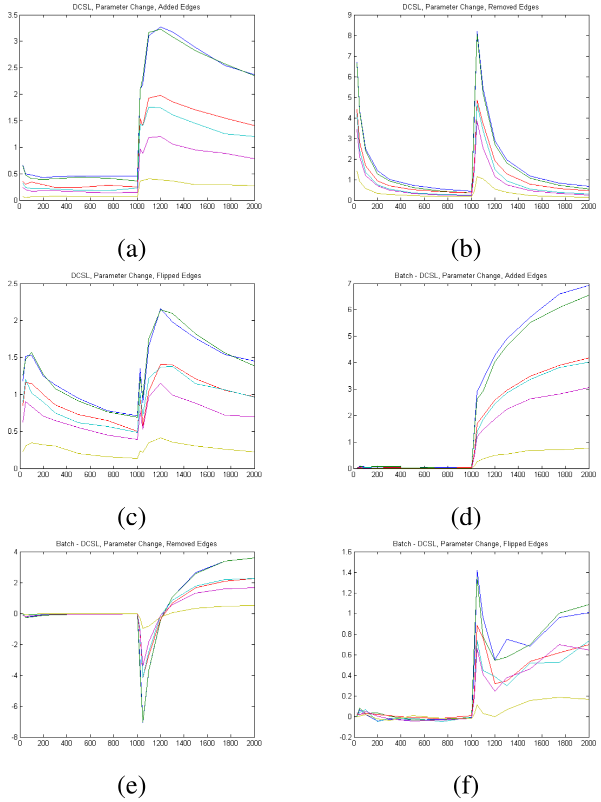
(c) (d)

(e) (f)

Figure 2: (a-c) DOCL Errors; (d-f) DOCL Improvement Over Batch When Structure Changes

ulations, the same measures as Figures 2(a-c) and 2(d-f). Again, DOCL and PC performed basically identically for the first 1000 datapoints. Performance after the parameter change did not follow quite the same pattern as before, however. DOCL again does much better on edge addition and orientation errors, but performed significantly worse on edge removal errors for the first 200 points following the change. Recall, however, that the *graph* does not change after 1000 datapoints, only the parameters. And since the parameter change leads PC to initially add edges, not remove them, it temporarily outperforms DOCL. That is, the difference in performance is more because PC is "lucky" that the graph does not change, rather than being due to a failure by DOCL.

The third set of simulations was designed to explore in detail the performance of the probabilistic relearning scheduler. We randomly generated a single dataset with 10,000 datapoints, where the underlying causal SEM graph and parameters changed after every 1000 datapoints. Each causal SEM had 10 variables and maximum degree of 7. We then ran DOCL with the probabilistic relearning schedule 100 times on this dataset. Figure 4(a) shows the (observed) expected number of "relearnings" in each 25-datapoint window. As expected, there are substantial relearning peaks after each structure shift, and the expected number of relearn-

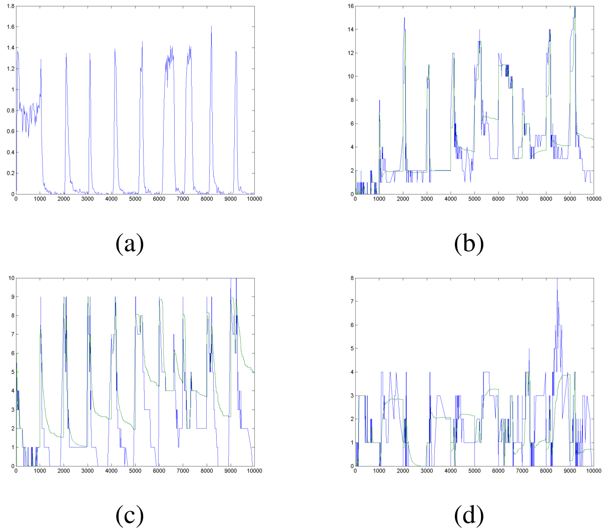


(a) (b)

(c) (d)

(e) (f)

Figure 3: (a-c) DOCL Errors; (d-f) DOCL Improvement Over Batch When Parameters Change



(a) (b)

(c) (d)

Figure 4: (a) DOCL Expected Relearnings for 25-Point Window; (b-d) Errors and Optimal Performance

ings persisted at roughly 0.05 per 25 datapoints throughout the 1000-datapoint stable period. Figures 4(b-d) provide error information: the smooth green lines indicate the mean edge addition, removal, and orientation errors (respectively) during learning, and the blocky blue lines indicate the DOCL errors if graph relearning occurred after every datapoint (i.e., optimal DOCL performance). Although there are many fewer graph relearnings with the probabilistic schedule, overall errors did not significantly increase.

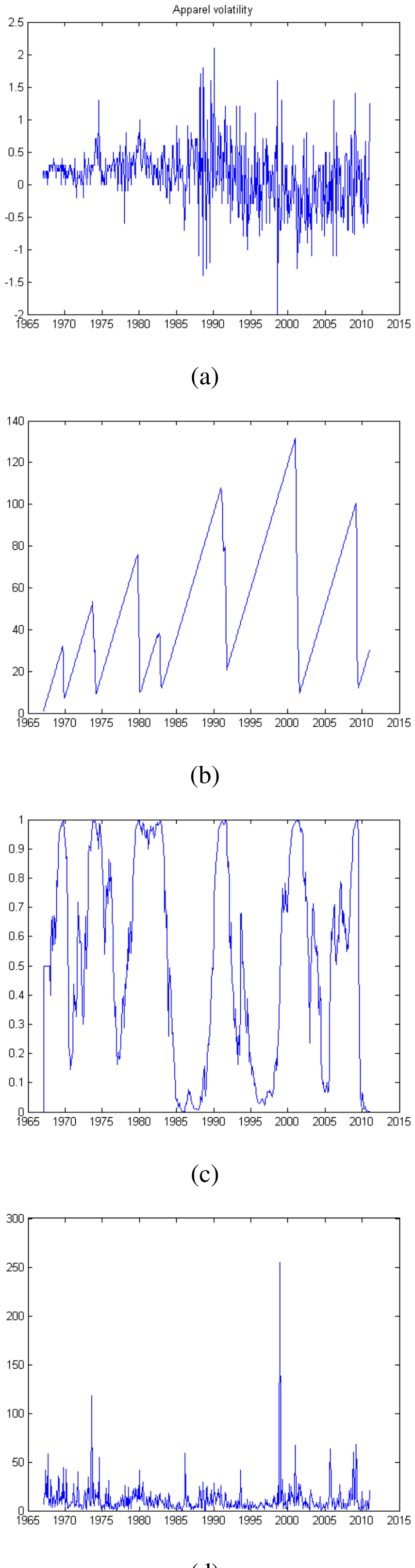
## 5 APPLICATION TO US PRICE INDEX VOLATILITY

We have applied DOCL to seasonally adjusted price index data available online from the U.S. Bureau of Labor Statistics. We limited the data to those commodities/services which had data going back to at least 1967, resulting in a data set of 6 variables: Apparel, Food, Housing, Medical, Other, and Transportation. The data was collected monthly from 1967 until 2011, resulting in 529 data points. Because of significant trends in the indices over time, we used the month-to-month differences. Figure 5(a) shows the relevant data for the Apparel index.

Figure 5(b) shows the change in effective sample size, where significant drops indicate the detection of change. Figures 5(c) and 5(d) show the drivers of those changes: the pooled p-value and Mahalanobis distance for each month, respectively. Notably, DOCL appears to detect (much of) the Great Moderation, a well-known macroeconomic phenomena between 1980 and 2007 in which the U.S. financial market underwent a slow but steady reduction in volatility. DOCL detects a shift in the volatility of the causal relationships among these price indexes around 1980, and then detects comparatively few such changes until shortly after 2000.[5] This real-world case study also demonstrates the importance of using pooled p-values, as DOCL does not respond to the single-month spike in Mahalanobis distance in 1995, but does respond to the extended sequence of slightly above average Mahalanobis distances around 1980.

## 6 DISCUSSION AND FUTURE RESEARCH

The DOCL algorithm will clearly have difficulty if the underlying causal structure changes very rapidly during data collection, or if the datapoints are a random-order mixture from multiple causal structures. The algorithm will find continually high Mahalanobis distance in the CMCD module, and so produce high learning rates in the OCME module. Of course, it is unclear whether *any* causal learning algorithm could be successful in a sufficiently non-stationary



(a)

(b)

(c)

(d)

Figure 5: (a) Sample of BLS Volatility Data; (b) Effective Sample Size During DOCL Run; (c) Pooled p-value; (d) Mahalanobis Distance

[5] The shift detected in 1991 is likely due to the recession of the early 1990s, and the shift in 2001 is almost certainly due to the U.S. recession that occurred in March to November of that year.

environment. Nonetheless, an important future research direction is to improve DOCL's performance on rapidly varying causal structures.

The DOCL algorithm also allows theoretically for the possibility that the current learned causal model can be used to influence the $a_r$ weights. If there are certain causal connections that have not changed over a long period of time, or have been stable over multiple relearnings, then one might plausibly conclude that those connections are stable and less likely to change. Thus, much greater error should be required to substantially change the estimates for those connections. In practice, implementing this intuition requires allowing for the $a_r$ weights to vary across $\langle V_i, V_j \rangle$ pairs. The mathematics of the OCME become much more complicated when this is allowed, and much more information must be tracked. It is currently unknown whether the (presumably) improved tracking would compensate for the additional computational and memory cost.

We have focused on causal models represented by SEMs, but there is also a long history of successful causal modeling using causal Bayes nets defined over discrete variables with conditional probability tables. Tracking the sufficient statistics for causal Bayes net structure learning is substantially more costly, and we are currently investigating ways to learn the necessary information in a tractable, online fashion. Similarly, we have focused on constraint-based structure learning since the relevant scores in score-based methods (such as (Chickering, 2002)) do not decompose in a manner that is suitable for online learning. We are thus investigating alternative scores, as well as heuristic approximations to principled score-based search.

There are many real-world contexts in which batch-mode causal structure learning is either infeasible or inappropriate. The online causal structure learning algorithm that we presented has great potential to perform well in a range of challenging contexts, and at little cost in "traditional" settings.

### Acknowledgments

Thanks to Clark Glymour, Peter Spirtes, and two anonymous UAI referees for their helpful comments. EK was supported by a grant from the James S. McDonnell Foundation. DD was partially supported by a James S. McDonnell Foundation Scholar Award.

## References

R. P. Adams and D. J. C. MacKay. Bayesian online changepoint detection. Technical report, University of Cambridge, Cambridge, UK, 2007. arXiv:0710.3742v1 [stat.ML].

D. M. Chickering. Learning Bayesian networks is NP-complete. In *Proceedings of AI and Statistics*, 1995.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 8:2961–2974, 2005.

D. Heckerman, C. Meek, and G. Cooper. A bayesian approach to causal discovery. In C. Glymour and G. Cooper, editors, *Computation, Causation, and Discovery*, pages 141–165. MIT Press, 1999.

T. Liptak. On the combination of independent tests. *Magyar Tud. Akad. Mat. Kutato Int. Kozl.*, 3:171–197, 1958.

P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55, 1936.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

M.R. Siracusa and J.W. Fisher III. Tractable bayesian inference of time-series dependence structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.

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

R. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

M. Talih and N. Hengartner. Structural learning with time-varying components: tracking the cross-section of financial time series. *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 67(3):321–341, 2005.