IIE Transactions
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713772245

Knowledge discovery from observational data for process control using causal Bayesian networks

Jing Li [a]; Jianjun Shi [a]
[a] Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA

Online Publication Date: 01 June 2007
To cite this Article: Li, Jing and Shi, Jianjun (2007) 'Knowledge discovery from observational data for process control using causal Bayesian networks', IIE Transactions, 39:6, 681 - 690
To link to this article: DOI: 10.1080/07408170600899532
URL: http://dx.doi.org/10.1080/07408170600899532

IIE Transactions (2007) **39**, 681–690
Copyright © "IIE"
ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170600899532

Knowledge discovery from observational data for process control using causal Bayesian networks

JING LI and JIANJUN SHI*

Department of Industrial and Operations Engineering, University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109-2117, USA
E-mail: jinglz@umich.edu or shihang@umich.edu

Received March 2004 and accepted March 2006

This paper investigates learning causal relationships from the extensive datasets that are becoming increasingly available in manufacturing systems. A causal modeling approach is proposed to improve an existing causal discovery algorithm by integrating manufacturing domain knowledge with the algorithm. The approach is demonstrated by discovering the causal relationships among the product quality and process variables in a rolling process. When allied with engineering interpretations, the results can be used to facilitate rolling process control.

Keywords: Bayesian network, process control, knowledge discovery, causal modeling, rolling process

1. Introduction

Rapid advances in sensing and computing technologies have allowed numerous sensors to be imbedded in manufacturing systems to collect on-line production data. Analysis of the extensive datasets could lead to more effective process control, unfortunately, however, the data analysis techniques currently used are limited in scope. Most of them focus on correlation or association analysis, which concerns how to predict certain features (e.g., product quality) of a system from other features (e.g., process parameters). However, to establish effective process control strategies, it is important to find the "causal" relationships among the features, e.g., which process parameters lead to the changes in quality levels. It is well known that "correlation does not imply causation". Significant process parameters for quality prediction may include those, but may not be exactly the ones, causing the quality problem. Thus, there is a need to identify the causal relationships among variables, which go beyond correlation or association for more effective process control.

Causal discovery from observational (i.e., uncontrolled nonexperimental) data is a challenging issue. Various research efforts have been made to develop generic causal discovery algorithms (Cooper and Herskovits, 1992; Lam and Bacchus, 1993; Spirtes, Glymour and Scheines 1993; Heckerman, 1998). Most of the implementations and applications of these algorithms are in the fields of genetics (Friedman *et al.*, 2000; Rodin and Boerwinkle, 2005), ecology (Marcot *et al.*, 2001; Borsuk *et al.*, 2004), social sciences (Dai *et al.*, 1997), medical sciences (Mani and Cooper, 1999), and physical sciences (Gutiérrez *et al.*, 2004). However, few have been found to conduct causal modeling from manufacturing data, especially for process control purposes (Lerner, 2002).

In a manufacturing system, the causal relationships are complicated, nonlinear, and dynamic, which generates considerable difficulties in the causal modeling of the underlying system. It is almost impossible to develop a universal method without specific manufacturing domain knowledge. In this research, therefore, emphasis is placed on developing a causal modeling approach that integrates a generic statistical causal discovery algorithm with manufacturing domain knowledge. In a manufacturing system, information flow is determined by the nature of each physical action and the topology of the physical system. The information related to key process/product features evolves in the system following engineering principles. From product/process designs, some engineering knowledge exists that can be used to help identify the key variables and potential causal relationships (with different confidence levels, or even qualitatively only). Meantime, the data captured by in-process sensors record the process changes and interrelationship among the variables during the operations. By integrating these two sets of information (information flow and data), a causal model can be discovered from observational data which can then be used to develop effective process control strategies.

*Corresponding author

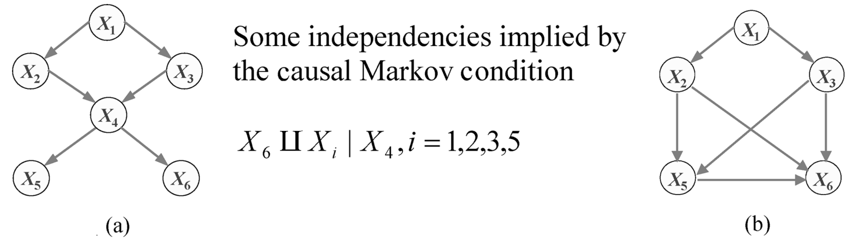


Fig. 1. (a) A causal network; and (b) the causal network with X_4 omitted.

In this paper, an integrated modeling approach is proposed to discover the causal relationships, represented by a causal Bayesian network (or causal network for short), in which engineering domain knowledge is embedded in various critical stages of a generic causal discovery algorithm. The approach is demonstrated by application to a rolling process control problem based on real production data. In this study, the product quality is measured in terms of surface defects. The process parameters include 22 variables, collected from two major manufacturing stages, continuous casting (i.e., prerolling) and rolling. With a causal network representation, the causal relationships among the variables can be identified and they can then be used to facilitate diagnosis, prediction, and control of the rolling process.

The rest of this paper is divided into three sections. Section 2 discusses how to obtain the causal network from observational data by integrating manufacturing domain knowledge with a generic learning algorithm. Section 3 illustrates how to derive the causal network from rolling data and utilize the results for process control. We give our conclusions in Section 4.

2. Causal modeling of observational data for application to process control

The causalities in a system can be represented by a causal network. Detailed concepts and properties of causal networks can be found in Glymour and Cooper (1999) and Korb and Nicholson (2003). In this section, notation and concepts relevant to process control applications are highlighted in Section 2.1. Section 2.2 illustrates how to integrate manufacturing domain knowledge with the generic Peter-Clark (PC) learning algorithm.

2.1. *Key notations and concepts of causal networks in process control applications*

A causal network is a **Directed Acyclic Graph** (DAG), namely, a set of nodes $\{X_1, \ldots, X_n\}$ connected by directed arcs. A directed graph is **acyclic** if there is no directed path $X_i \to \cdots \to X_j$ such that $X_i = X_j$, which implies that there are no feed-back loops in the system. The nodes represent random variables. In process control, a node X_i can be a process variable, a product quality variable, or any feature extracted from a process or product quality variables. If there is a directed arc from X_i to X_j, X_i is called a **parent** of X_j and the directed arc is interpreted as a direct causal influence that X_i exerts on X_j. X_i and X_j are **adjacent** if and only if X_i is a parent of X_j or X_j is a parent of X_i. For each node X_i with parents $\{Pa_1(X_i), \ldots, Pa_{m_i}(X_i)\} \subset \{X_1, \ldots, X_n\}$, the effects that the parents have on the node are quantified by the conditional probability distribution $P(X_i|Pa_1(X_i), \ldots, Pa_{m_i}(X_i))$. An example causal network is given in Fig. 1(a). Assuming that it describes a manufacturing system with six variables where X_6 measures the product quality and the rest are process parameters, this causal network implies, for example, that X_2 and X_3 can causally influence X_4, which in turn can causally influence the quality X_6, but X_5 has no causal influence on X_6.

The following causal network concepts are important in this application.

2.1.1. *Causal Markov condition*

The DAG encodes the causal Markov condition, i.e., a variable is independent of its nondescendants given its parents. X_j is a **descendant** of X_i if there is a directed path from X_i to X_j. With the causal Markov condition the independencies in a DAG can be identified, which is important for evaluating the consequence of a process control action. For example, in Fig. 1(a), if X_4 is controlled (e.g., set to be a certain value), then a change in X_i ($i = 1, 2, 3$) will not lead to any change in X_6. In other words, controlling X_4 blocks the paths through which X_i ($i = 1, 2, 3$) can impact the quality X_6. None of the other process variables have this property in this causal network. For instance, if only X_2 is controlled, the quality X_6 may still be affected by X_1, X_3 or X_4.

2.1.2. *Causal sufficiency*

The variables in a DAG satisfy causal sufficiency if and only if all the common causes of pairs of variables are included in the DAG. The violation of causal sufficiency may result in ineffective/inefficient control actions. For example, if X_4 (the common cause of X_5 and X_6) in Fig. 1(a) is omitted, the remaining variables will form a different DAG (shown in Fig. 1(b)) in which X_5 is falsely considered as a cause of X_6. However, any attempt to control X_5 is in fact ineffective in terms of quality improvement since X_5 does not impact the quality variable X_6.

To identify a set of causally sufficient variables, denoted by **V**, the following procedure can be used in a process control application.

1. Targeting a particular quality concern, determine the quality variable and include it in **V**.
2. For the quality variable in **V**, include all its potential causes in **V**. For each newly added variable in **V**, include all its potential causes in **V**.
3. Repeat step 2 until all newly added variables are believed to have no causes.

It should be pointed out that for a variable X_i in **V**, adding a false cause to **V** has little negative impact because the learning algorithm will correct it (e.g., remove it through causal discovery). However, omitting a true cause may violate causal sufficiency and generate a problematic DAG. Thus, it is recommended to include all potential causes of X_i in **V**, i.e., the variables are suspected to be the causes with at least some confidence.

2.2. *Incorporating manufacturing domain knowledge with causal discovery*

Causal discovery from observational data consists of two steps: learning the structure (i.e., the DAG) and learning the parameters (i.e., the conditional probability distributions) given a structure. A typical algorithm to learn the structure is the PC algorithm (Spirtes, Glymour and Scheines, 1993; Meek, 1995), which relies on the hypothesis testing of independence. However, although numerous tests of independence exist for discrete variables, establishing statistical tests applicable to mixed discrete/continuous variables or a variety of distributions on continuous variables remains an open problem (Glymour and Cooper, 1999). Therefore, only discrete variables are considered in this paper, i.e., each variable can only take the values from a finite set. Each value is called a **state** of this variable.

After the structure is obtained, the parameters can be learned based on the structure and the data. Learning algorithms exist for the cases of both complete and incomplete datasets (Lauritzen, 1995). A summary and comparison of typical algorithms can be found in Buntine (1996).

This section briefly introduces the PC algorithm in Section 2.2.1. We draw attention to Section 2.2.2 in which we illustrate how to integrate manufacturing domain knowledge with the PC algorithm so as to discover causal relationships that can subsequently be used for process control purposes.

2.2.1. *The PC algorithm*

Before presenting the PC algorithm, some additional notations are defined here. Let $Adj(\mathbf{V}, X_i)$ denote the set of variables in **V** adjacent to X_i, and $Adj(\mathbf{V}, X_i)\backslash\{X_j\}$ denote the complement of $\{X_j\}$ with respect to $Adj(\mathbf{V}, X_i)$. Let *Cardinality* $(\mathbf{A})$ denote the cardinality of a set **A** and denote $Cardinality(Adj(\mathbf{V}, X_i))$ by n_i. Let $Subset^d(\mathbf{A})$ denote the operation of searching for all the subsets with cardinality d of a set **A** $(0 \leq d \leq Cardinality(\mathbf{A}))$ and denote each of these subsets by

$$\mathbf{T}_l,\ l = 1, \ldots, \frac{Cardinality(\mathbf{A})!}{d!(Cardinality(\mathbf{A}) - d)!}.$$

For example, $Subset^2(\{X_1, X_2, X_3\}) = \{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$, where $\mathbf{T}_1 = \{X_1, X_2\}$, $\mathbf{T}_2 = \{X_1, X_3\}$ and $\mathbf{T}_3 = \{X_2, X_3\}$.

The PC algorithm (or PC for short) inputs the dataset and a fully connected undirected graph in which each variable is adjacent to all variables in **V** other than itself. Thus, there are $n(n-1)$ ordered pairs of adjacent variables in this initial graph. In the following description of the PC algorithm, each of these pairs is denoted by $\mathbf{M}_h$, $h = 1, \ldots, n(n-1)$, i.e., $\mathbf{M}_h = (X_i, X_j)$, $i = 1, \ldots, n$; $j = 1, \ldots, n$; and $i \neq j$. $X_i \coprod X_j | \mathbf{T}_l$ denotes that X_i is independent of X_j given $\mathbf{T}_l$. PC searches for the causal network structure through two stages, discovering all the adjacencies and identifying the arc directions, as shown in Fig. 2.

In Fig. 3(a–c), the progression of steps 1 to 12 in Fig. 2 is further illustrated through an example.

2.2.2. *Integrating manufacturing domain knowledge with PC for process control*

Domain knowledge plays an important role in causal discovery as it can effectively constrain the model search, reduce the computational complexity, increase model accuracy, and also help validate and interpret the results. In manufacturing systems, domain knowledge can include such things as an understanding of the process/product variables especially their physical meanings, distributions and interactions, the production flow and sensor placements, the procedure for data collection, data quality, and the engineering specifications that support decision-making. Figure 4(a and b) is a flow chart showing how to integrate domain knowledge with PC; in the figure, blocks with single lines represent the original operations in PC and blocks with double lines represent the operations induced by domain knowledge. In what follows, each block with double lines is interpreted.

(1) *Variable selection*

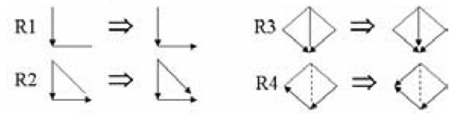
In complex manufacturing systems, usually more than one variable is measured to assess the product quality. The measured variables can include dimensions, mechanical properties and number of defects. If the study is focused on one particular type of quality problem, instead of including the whole set of process variables in the model, only a subset of these variables is selected. The choice of the variables considered to have potential causal relationships with the target quality problem is based on domain knowledge. In other words, domain knowledge can be used to define the objective of the causal modeling and also to partition a complex problem into specific problems by prescreening the variables. Furthermore, it is also important that the selected

(a) Steps to discover the adjacencies

1. Set $d = 0$.
2. Set $h = 1$.
3. If X_i and X_j in $\mathbf{M}_h$ are adjacent, go to 4. Otherwise, increment h by 1 and redo 3.
4. Find $Adj(\mathbf{V}, X_i)$.
5. If $n_i > d$, go to 6. Otherwise, increment h by 1 and go to 3.
6. Find $Subset^d\left(Adj(\mathbf{V}, X_i) \setminus \{X_j\}\right)$
7. Set $l = 1$.
8. If $X_i \perp\!\!\!\perp X_j \mid \mathbf{T}_l$, remove the adjacency between X_i and X_j, record $\mathbf{T}_l$ in $Indcondset(X_i, X_j)$, increment h by 1 and go to 3. Otherwise, increment l by 1 and redo 8.
9. Increment d by 1 and go to 2.

(b) Steps to identify the arc directions

10. For each triple of variables (X_i, X_j, X_k) where the pair X_i, X_j and the pair X_j, X_k are adjacent but the pair X_i, X_k is not, orient it as $X_i \rightarrow X_j \leftarrow X_k$ if and only if $X_j \notin Indcondset(X_i, X_k)$.
11. Apply the orientation rules R1~R4, i.e., if a set of variables satisfy the pattern to the left of the $\Rightarrow$ then orient the arcs according to the pattern to the right of the $\Rightarrow$. (The dash line in R4 stands for $\uparrow, \downarrow$, or $|$.)



12. Repeat 11 until no more adjacencies can be oriented.

Fig. 2. The PC algorithm.

variables satisfy causal sufficiency for effective process control purposes.

(2) *Continuous variable discretization*

In the presented PC algorithm, all variables are discrete. However, some of the sensors in manufacturing systems produce continuous data. Thus, discretization is needed. Some discretization methods can be found in Dougherty *et al.* (1995). In this section, a specific method is proposed to discretize the continuous variable with a special distribution.

In manufacturing systems, sensor measurements of continuous variables can generally be decomposed into two parts, $X = \mu_i + \varepsilon_i (i = 1, \ldots, t)$, where μ_i represents the ith nominal physical setting of the variable and ε_i reflects the random variation around μ_i. For example, in the continuous casting (prerolling) stage of a rolling process, electromagnetic stirring is adopted to ensure the complete mixing of the raw material. The speed of the stirring is often set to be one of two levels, either low or high. However, due to variations in material hardness, sensor measurements of the stirring speed are always continuous. In this paper, this type of continuous variable is termed as a **clustered continuous variable** since its distribution histogram often consists of several separate clusters. Under the assumptions that: (i) $\varepsilon_i = \varepsilon$ for $\forall i$; and (ii) ε follows a normal distribution with zero mean, i.e., $\varepsilon \sim N(0, \sigma^2)$, a procedure to discretize a clustered continuous variable X with m samples $x_1, \ldots, x_m$ is proposed as follows.

Step 1. Find $\mu_i, i = 1, \ldots, t$, based on domain knowledge.
Step 2. $\tilde{x}_j = \min_{i \in \{1,\ldots,t\}} (|x_j - \mu_i|), j = 1, \ldots, m.$
Step 3. $\hat{\sigma}^2 = \frac{1}{m-1} \sum_{j=1}^{m} \tilde{x}_j^2$.
Step 4. If $(1/(t-1)) \sum_{j=1}^{t-1} |\mu_j - \mu_{j+1}| > \gamma \hat{\sigma}$, go to Step 5. Otherwise, go to Step 6.
Step 5. Assign x_j to the ith state if $|x_j - \mu_i| \leq |x_j - \mu_k|$ for $\forall k \in \{1, \ldots, t\}, j = 1, \ldots, m.$
Step 6. Use the Equal Frequency Intervals method (Dougherty *et al.*, 1995).

The purpose of Steps (2) to (4) is to justify the discretization in Step (5). In other words, the proposed discretization method is valid only when the difference between physical settings overrides the random variation. The strength of the override is measured by a predefined value γ.

(3) *Dimension reduction of the state space*

Even though all variables are discrete with finite states, domain knowledge can be used to combine the states thus reducing the dimension of the state space and improving the computational efficiency. In Section 3.2.1, one specific approach based on clustering is discussed for combining the states of the variables in a rolling process.

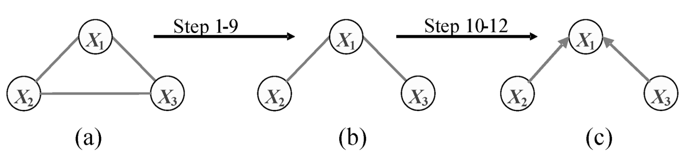


Fig. 3. The progression of steps 1 to 12 of PC: (a) the fully connected graph; (b) the partially connected graph; and (c) the final causal network.

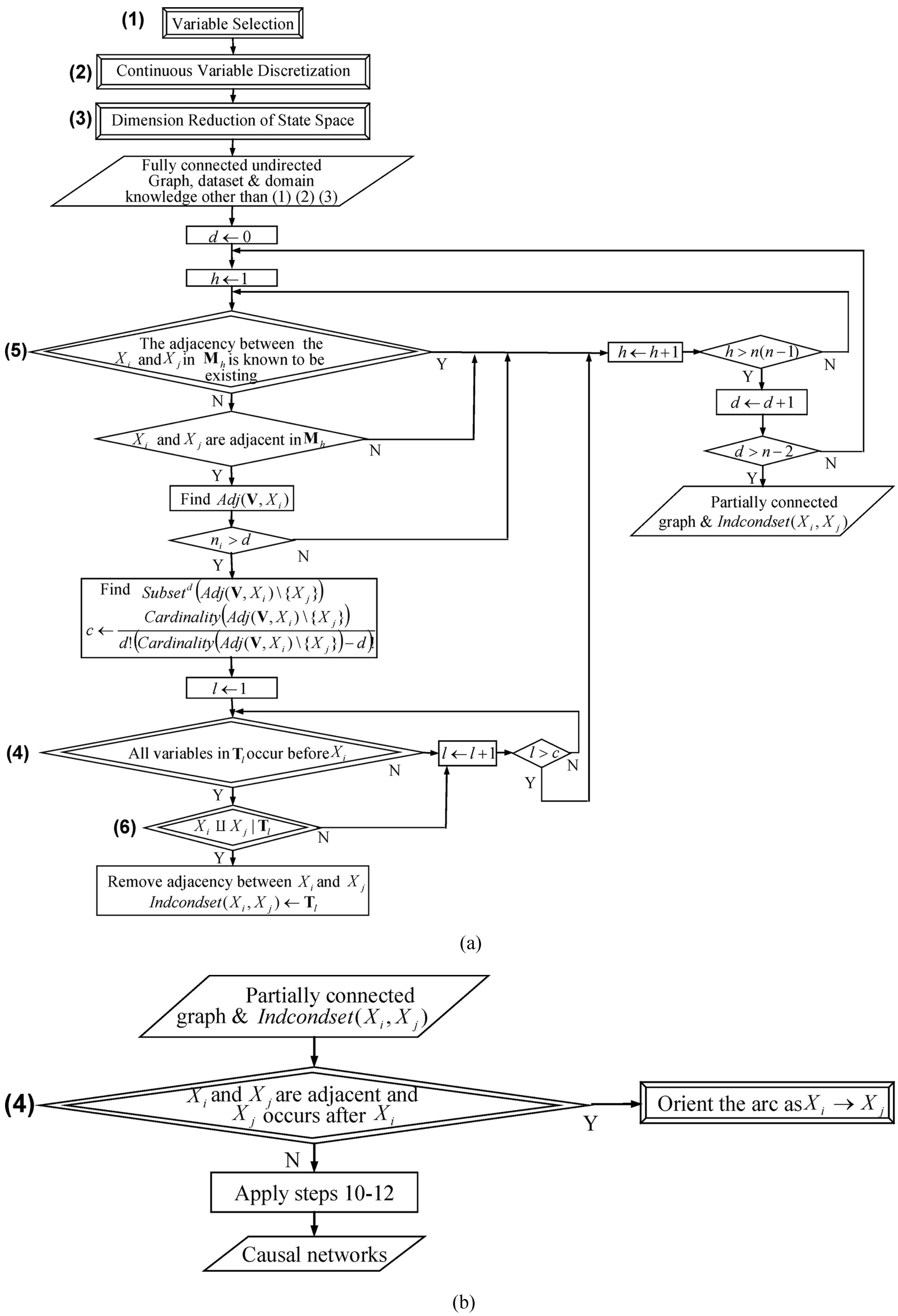


Fig. 4. (a) Integration of domain knowledge with steps 1–9 of PC; and (b) integration of domain knowledge with steps 10–12 of PC.

(4) *Temporal order of variables*

The relative position of sensors in the production flow provides information on the temporal order of variables, that is, a complete or partial order of variables according to the time or sequence that they are measured. Based on this information, any pair of variables can be either temporally distinguishable or indistinguishable. If X_i is measured prior to X_j such that their temporal order is known, they are called **temporally distinguishable**, denoted by X_i P.T. X_j (P.T. stands for "Prior To"). Otherwise, if X_i and X_j are measured simultaneously, or the time delay is not significant, they are called **temporally indistinguishable**, denoted by (X_i, X_j) or equivalently (X_j, X_i). Two basic properties associated with temporally distinguishable or indistinguishable variables are:

1. Transitional property: If X_i P.T. X_j, and X_j P.T. X_k, then X_i P.T. X_k. Similarly, If (X_i, X_j) and (X_j, X_k), then (X_i, X_k). In the latter case, the set of temporally indistinguishable variables can be included in one bracket, i.e., (X_i, X_j, X_k).
2. Distributive property: Suppose $\mathbf{S}_1$ and $\mathbf{S}_2$ are two sets of temporally distinguishable variables. If $\mathbf{S}_1$ P.T. $\mathbf{S}_2$, then for $\forall\ X_i \in \mathbf{S}_1$ and $\forall\ X_j \in \mathbf{S}_2$, X_i P.T. X_j.

According to those two properties, the temporal relationship between any pair of variables in a manufacturing system can be identified based on their sensor locations. This information can be further used to constrain the model search and reduce the computational complexity of PC. Specifically, the following two operations are incorporated into PC (Spirtes, Glymour, Scheines, Meek and Richardson, 1993), corresponding to block (4) in both Fig. 4(a) and Fig. 4(b).

1. In testing whether X_i and X_j are independent given $\mathbf{T}_l$, if there exists an $X_k \in \mathbf{T}_l$ such that X_i P.T. X_k, do not test for their independence conditional on $\mathbf{T}_l$.
2. If X_i and X_j are adjacent and X_i P.T. X_j, orient the adjacency as $X_i \rightarrow X_j$.

(5) *Engineering-specified adjacencies*

Adjacency is different from dependence. If two variables are adjacent, they must be dependent. However, if two variables are dependent, they are not necessarily adjacent unless one is a parent of the other. The dependence between some variables can be ascertained based on an understanding of the manufacturing system. For example, in rolling processes, each type of material is assigned a specific range of rolling speeds and any speed beyond this range is not allowed. Thus, material and rolling speed are always dependent. To ascertain the adjacency between two dependent variables, additional conditions have to be satisfied. In what follows, a sufficient condition to justify the adjacency is proposed.

A sufficient condition of adjacency: X_i and X_j are adjacent if all the following three conditions are satisfied: (i) their dependence is justified by domain knowledge; (ii) they are temporally distinguishable; and (iii) measured successively.

If the existence of certain adjacencies can be ascertained based on engineering knowledge, there is no need to check them by conducting independence tests. The strategy of incorporating such knowledge into PC is illustrated in block (5) of Fig. 4(a).

(6) *Identification of structural zeros*

To conduct a conditional independence test, the data needs to be organized into a contingency table. If a cell with zero counts is present, it is important to distinguish between two different types of zeros before performing the test. A "sampling zero" is introduced by an insufficient sample size. A "structural zero" occurs in a cell in which the combination of the states of the variables is inherently impossible to generate any counts. Structural zeros are not uncommon in a manufacturing system. They usually result from limitations imposed by the engineering operations. For instance, a certain type of material can only be processed at a certain temperature. Cells containing structural zeros can be identified with the aid of domain knowledge. Then in the corresponding test of conditional independence, the test statistic can be calculated by simply ignoring these cells since they do not really exist, however, computation of the degrees of freedom does require those cells to be taken into account (Christensen, 1997).

3. Causal discovery for rolling process control

A causal network can serve as an effective tool to identify the causal factors producing quality problems and facilitate the formulation of better process control strategies. This section reports a procedure and initial results of applying this technique to rolling process control problems. Section 3.1 gives a brief introduction to the rolling process, quality and process variables. Section 3.2 illustrates how to derive the causal network based on rolling data by integrating domain knowledge with the PC algorithm. Section 3.3 presents two case studies to illustrate how to use the obtained casual network for process control.

3.1. *Rolling process, quality and process variables*

Rolling is a high-speed bulk deformation process that reduces the thickness or changes the cross-section of a long work piece by compressive forces applied through a set of rolls (Kalpakjian and Schmid, 2003). Surface defects are generally used as a measure of product quality. Among all the possible surface defects, the seam defect is one of the most critical quality concerns. A seam is a crack that is aligned parallel to the worked metal surface, and is usually the result of nonmetallic inclusions, cracking or porosity. In this study, the quality variable is the number of seams on each rolled bar. Also, 22 process variables are gathered from two manufacturing stages, continuous casting (pre-rolling) and rolling. The dataset consists of 100 000 records collected from a real production environment. Each record corresponds to a rolled bar.

Table 1. Process and quality variables

Process variables	*Batch*	Melt shop batch number	Nominal
	Tap	Time from the tapping of one batch to the tapping of the next batch at the melt furnace (minutes)	Continuous
	Vac	The time that the steel is exposed to a high vacuum at the degasser (minutes)	Continuous
	Vad	The number of degrees F above the liquidious temperature for this grade of steel	Continuous
	Cast	The absolute temperature of the steel in the tundish	Continuous
	Liq	The temperature at which the steel begins to solidify upon cooling	Continuous
	Zone2	Spray water zone 2 flow (gallon/minute)	Continuous
	Mold	Distance of liquid steel level from top of mold	Continuous
	SpeedC	Average casting speed	Continuous
	Ems	Electromagnetic stirring (amps)	Continuous
	Grade	Grade of steel	Nominal
	SpeedR	Speed of rolling	Continuous
Quality variable	*Defect*	Number of seams on each rolled bar	Ordinal

3.2. *Causal discovery by integrating domain knowledge with PC*

This section illustrates how to integrate domain knowledge on the rolling process with PC to derive the causal network.

3.2.1. *Utilization of domain knowledge*

Each of the six types of domain knowledge discussed in Section 2.2.2 is reviewed for the specific application of a rolling process.

(1) *Variable selection*

Among the 22 process variables, 12 of them are considered to have potential causal impacts on seam production while the others are believed to be related to other types of defects. These 12 variables, together with the quality variable *Defect*, were selected according to the procedure outlined in Section 2.1 to satisfy causal sufficiency. Table 1 lists the variables, together with a description of their physical meanings and attributes.

(2) *Continuous variable discretization*

Except for *Defect*, *Batch* and *Grade*, all the other variables are continuous, with *Mold*, *Ems* and *SpeedR* being clustered continuous variables. These three variables are discretized according to Steps (1)–(4) in Section 2.2.2 (the second type of domain knowledge), where γ is set to have a value of two. *Zone2* and *SpeedC* are regular continuous variables. They are all discretized using the Equal Frequency Intervals method (Dougherty, 1995), into three states. *Tap*, *Vac*, *Vad*, *Cast* and *Liq* need not be discretized. The reason is explained below.

(3) *Dimension reduction of the state space*

Batch is a nominal variable. However, if each distinct value is assigned with a separate state, there will be thousands of states for this variable. Two problems can occur in this situation. Since the number of records belonging to each state is relatively small, the estimation of the conditional probability distribution may be biased. Furthermore, with a large number of states, conditional independence tests are involved, which may significantly decrease the modeling speed. Besides the computational concerns, there are also difficulties in interpreting the results. Different from other variables, the values of *Batch* do not correspond to any physical settings. Instead, they are indices used to label each outgoing batch in the melt shop. If some batches have quality problems they can be identified with the aid of causal modeling algorithms. However, to establish proper control strategies, there is a further need to understand why these batches are problematic.

Engineering knowledge of the rolling process suggests that five variables are particularly critical in terms of affecting the quality of each batch: they are *Tap*, *Vac*, *Vad*, *Cast* and *Liq*. To distinguish these five variables from other process variables, they are termed the **profile variables** of *Batch*. And the **profile** of a *Batch* is defined as a vector containing the values of the five profile variables corresponding to that particular *Batch*. It is reasonable to believe that batches with similar profiles have similar impacts on the product quality. Therefore, the profile variables are used to group the batches.

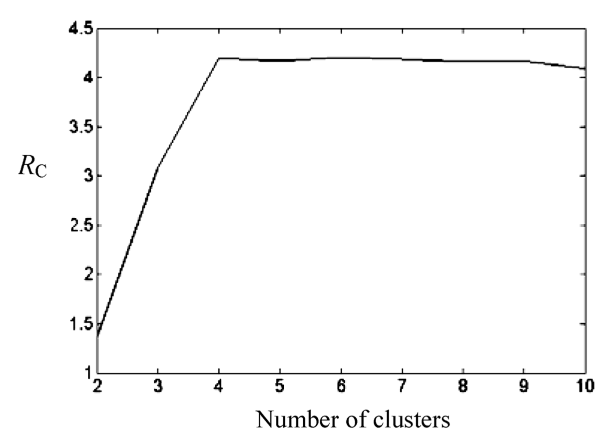


Fig. 5. Selection of an appropriate number of clusters.

Table 2. The variables used in the causal modeling and the number of states for each variable

Defect	*Batch*	*Zone2*	*Mold*	*SpeedC*	*Ems*	*Grade*	*SpeedR*
3	4	3	2	3	2	8	3

A hierarchical clustering approach with a Euclidian distance measure and Ward linkage (Arabie *et al.*, 1996) is adopted to group the batches and the profile variables are standardized before clustering. Since it is a hierarchical approach, the partition of the data is not unique unless the number of clusters is specified. One way to evaluate the partition (i.e., select the number of clusters) is to compare the between-cluster distance with the within-cluster distance. Figure 5 plots the ratio of average between-cluster distance to average within-cluster distance, denoted by R_C, versus the number of clusters. If the number of clusters is equal to two, then $R_C = 1.4$, implying that the between-cluster distance is not significantly larger than the within-cluster distance. If the number of clusters exceeds four, there is no significant gain obtained by increasing the number of clusters. Therefore, three or four clusters are considered to be adequate. Specifically, four clusters are used in this study and each is assigned with a separate state.

Similar to *Batch*, *Defect* also has a large number of distinct values although it is an ordinal variable. Because *Defect* is an indicator of the severity of the quality problem, its adjacent values can be combined according to engineering specifications. Therefore, "*Defect* = 0", "*Defect* = 1–7" and "*Defect* > 7" are assigned with three separate states, representing "satisfactory quality", "minor quality problems" and "severe quality problems".

Following the procedures in (2) and (3), eight variables are selected for causal modeling and the number of states for each variable is summarized in Table 2.

(4) *Temporal order of variables*

Figure 6 shows the production sequence of a rolling process and the locations where the variables are measured. Based on this knowledge, the temporal order of the variables can be specified as

Grade P.T. *Batch*
Batch P.T. (*Ems*, *Mold*, *Zone2*, *SpeedC*)
(*Ems*, *Mold*, *Zone2*, *SpeedC*) P.T. *SpeedR*
SpeedR P.T. *Defect*

Using the transitional and distributional properties proposed in Section 2.2.2 (the fourth type of domain knowledge), pairwise temporal relationships can be further identified.

(6) *Identification of structural zeros*

Because a particular grade of steel is only allowed to be rolled at a certain speed, the (conditional) independence tests involving *Grade* and *SpeedR* contain structural zeros. These tests are modified using a different way to calculate the test statistics and the degrees of freedom, as discussed in Section 2.2.2. Due to the large sample size, there are no sampling zeros in this study.

Table 3. $p(Defect|\ Batch, Grade, SpeedR)$

Batch	Grade	SpeedR	$p(Defect = 1\|$ Batch, Grade, SpeedR)	$p(Defect = 2\|$ Batch, Grade, SpeedR)	$p(Defect = 3\|$ Batch, Grade, SpeedR)
2	4	2	0.05	0.4086	0.5414
3	7	3	0.069	0.3448	0.5862
3	1	2	0.8558	0.0869	0.0573
3	6	2	0.0622	0.6378	0.3

3.2.2. *Results of learning*

The DAG obtained by following the steps of causal modeling outlined in Section 2, is shown in Fig. 7. Notice that the arc between *Zone2* and *SpeedC* is not oriented, implying that the DAG represents two Markov equivalent causal networks (Spirtes, Glymour and Scheines, 1993), in one of which the arc is oriented as $Zone2 \rightarrow SpeedC$ and in the other as $Zone2 \leftarrow SpeedC$.

Based on the DAG and data, the parameters (i.e., the conditional probability distribution of each variable given its parent) are further learned using the EM algorithm (Lauritzen, 1995). For illustrative purpose, Table 3 gives part of the conditional probability distribution of *Defect* given *Batch*, *Grade* and *SpeedR* as an example.

3.3. *Process control based on the causal network*

Based on the causal network in Fig. 7, the causal relationships among variables can be identified both qualitatively and quantitatively. Two useful qualitative conclusions are:

1. Three process variables are discovered to be the causes of *Defect*: they are *Grade*, *Batch* and *SpeedR*.
2. Other process variables do not contribute to *Defect*, although they have dependencies with *Defect*. Specifically, the dependence between *Mold* and *Defect* is due to the common cause *Batch*; the dependence between

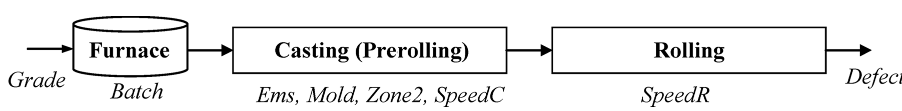


Fig. 6. The production sequence of a rolling process.

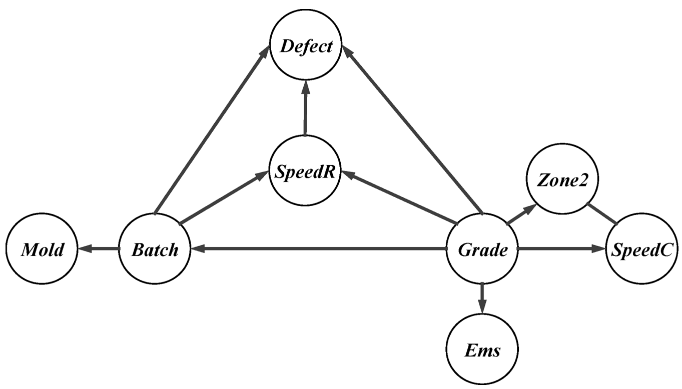


Fig. 7. The DAG (i.e., causal network structure) based on rolling data.

Zone2/SpeedC/Ems and *Defect* is due to the common cause *Grade.*

These conclusions imply that although many process variables have dependencies with *Defect*, only three of them causally impact the product quality. Thus, for effective process control, attention should be focused on the causal process variables. Controlling other dependent but noncausal process variables will not lead to changes in the quality conditions.

Furthermore, the quantitative results to facilitate process control are illustrated by two case studies.

Case study I (Diagnosis): If $Defect = 3$ is observed, i.e., severe quality problems occur, how to determine which state of *Batch*, *Grade* and *SpeedR* has the highest possibility to cause the problems?

One solution to this problem is to calculate three conditional probability distributions, i.e., $p(Batch|Defect = 3)$, $p(Grade|Defect = 3)$ and $p(SpeedR|Defect = 3)$. For example, $p(Batch|Defect = 3)$ can be obtained using:

$$p(Batch|Defect = 3) = \frac{p(Batch, Defect = 3)}{p(Defect = 3)} = \frac{p(Batch, Defect = 3)}{\sum_{Batch} p(Batch, Defect = 3)},$$

where $p(Batch = i, Defect = 3)$ $(i = 1, \ldots, 4)$ can be acquired by dividing the number of samples with $Batch = i$ and $Defect = 3$ by the total sample size. Figure 8 plots the obtained conditional probability distributions. Based on the figure, if only *Grade* is considered, $Grade = 7$ has the highest possibility to cause severe quality problems. For the same reason, attention needs to be paid to $Batch = 2$ *and* 3 and $SpeedR = 2$ and 3.

Case Study II (Forward prediction): If a certain combination of the process variables are observed, for example, $Batch = 3$, $Grade = 1$ and $SpeedR = 2$, how to determine the severity of the quality problems it may lead to?

This problem requires $p(Defect|Batch, Grade, SpeedR)$, which can be directly obtained from the results of the parameter learning. The $p(Defect|Batch, Grade, SpeedR)$ for several combinations of the three causal process variables are plotted in Fig. 9. The combinations resulting in an increasing trend are $Batch = 2$, $Grade = 4$, $SpeedR = 2$ and $Batch = 3$, $Grade = 7$, $SpeedR = 3$. In other words, at these two combinations, it is highly possible that severe quality problems will occur. Conversely, the combination $Batch = 3$, $Grade = 1$, $SpeedR = 2$ results in an decreasing trend with $Defect = 1$ having a probability of 0.9 to occur. Thus, it can be considered as a quality-ensuring process setting. We suspect that the combination $Batch = 3$, $Grade = 6$, $SpeedR = 2$ can lead to minor quality problems. If there is a rigorous requirement for the product quality, this combination should also be avoided.

Special attention needs to be given to this inference with respect to zero probabilities. For example, $p(Defect = 3|Batch = i, Grade = j, SpeedR = k) = 0$ does not necessarily imply that (i, j, k) is a desirable setting of the process variables. This may be a result of the structural zeros associated with the particular combination of the states. Thus, it is important to utilize design specifications to inspect the setting and discern whether it is against any engineering constraints.

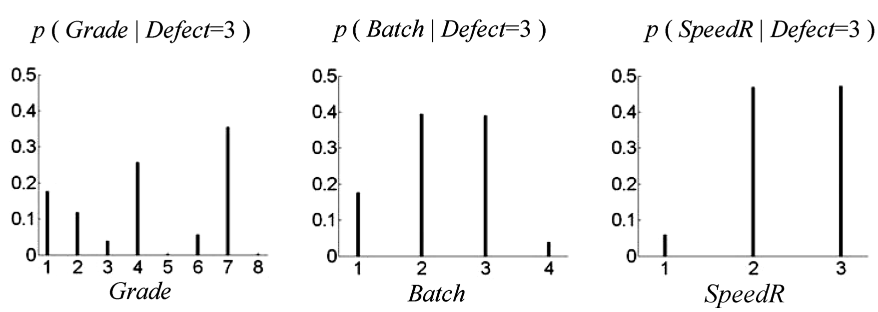


Fig. 8. Conditional distribution plots for causal process variables.

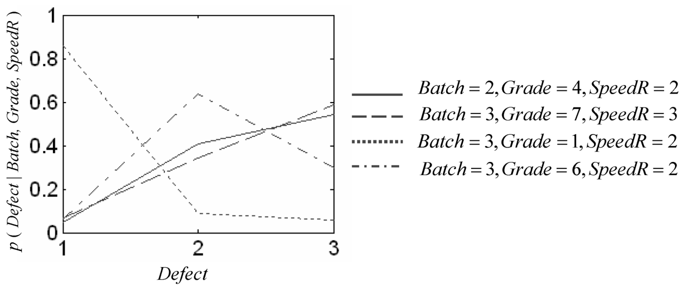


Fig. 9. Conditional distribution plots of *Defect* given certain combinations of the causal process variables.

4. Conclusions

This paper investigated a methodology to learn causal networks from observational data to facilitate process control. Theoretical details and representative algorithms were presented. Observational data collected from a rolling process were used to illustrate the use of the methodology in a real-world setting. With the aid of engineering knowledge, the results were analyzed and interpreted under two specific goals, process diagnosis and quality predication. Several problems are considered to be critical issues for the successful application of the proposed technique including variable preprocessing and interpretation of the results. We leave these topics to future studies. It is also of interest to compare causal networks with predictive models in terms of their prediction capacities.

Acknowledgements

The authors gratefully acknowledge the financial support of the NSF Engineering Research Center for Reconfigurable Machining Systems (NSF grant EEC95-92125) at the University of Michigan and the support from OG Technologies, Inc.

References

Arabie, P., Hubert, L.J. and De Soete, G. (1996) *Clustering and Classification*, World Scientific, River Edge, NJ.

Borsuk, M.E., Stow, C.A. and Reckhow, K.H. (2004) A Bayesian network of eutrophication models for synthesis, prediction, and uncertainty analysis. *Ecological Modelling*, **173**, 219–239.

Buntine, W. (1996) A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, **8**, 195–210.

Christensen, R. (1997) *Log-Linear Models and Logistic Regression*, Springer, New York, NY.

Cooper, G.F. and Herskovits, E. (1992) A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, **9**, 309–347.

Dai, H., Korb, K.B., Wallace, C.S. and Wu, X. (1997) A study of casual discovery with weak links and small samples, in *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI)*, Morgan Kaufman, San Francisco, CA, pp. 1304–1309.

Dougherty, J., Kohavi. R. and Sahami, M. (1995) Supervised and unsupervised discretization of continuous features, in *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufman, Los Altos, CA, pp. 194–202.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000) Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, **7**, 601–620.

Glymour, C. and Cooper, G.F. (1999) *Causation, Computation and Discovery*, MIT/AAAI Press, Cambridge, MA.

Gutiérrez, J.M., Cano, R. and Sordo, C.M. (2004) Applications of Bayesian networks in meteorology, in *Advances in Bayesian Networks*, Springer, New York, NY, pp. 309–327.

Heckerman, D. (1998) A tutorial on learning with Bayesian networks. In *Learning in Graphical Models*, Kluwer, Boston, MA, pp. 301–354.

Kalpakjian, S. and Schmid, S.R. (2003) *Manufacturing Processes for Engineering Materials*, Prentice Hall, Upper Saddle River, NJ.

Korb, K.B. and Nicholson, A.E. (2003) *Bayesian Artificial Intelligence*, Chapman & Hall/CRC, London, UK.

Lam, W. and Bacchus, F. (1993) Learning Bayesian belief networks: an approach based on the MDL principle. *Computational Intelligence*, **10**, 269–293.

Lauritzen, S.L. (1995) The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, **19**(2), 191–201.

Lerner, U. (2002) Hybrid Bayesian networks for reasoning about complex systems. Ph.D. Thesis, Stanford University, Stanford, CA.

Mani, S. and Cooper, G.F. (1999) A study in casual discovery from population-based infant birth and death records, in *Proceedings of the AMIA Annual Fall Symposium*, Hanley and Belfus, Philadelphia, PA, pp. 315–319.

Marcot, B.G., Holthausen, R.S., Raphael, M.G., Rowland, M. and Wisdom, M. (2001) Using Bayesian belief networks to evaluate fish and wildlife population viability under land management alternatives from an environmental impact statement. *Forest Ecology and Management*, **153**(1–3), 29–42.

Meek, C. (1995) Causal inference and causal explanation with background knowledge, in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, pp. 403–410.

Rodin, A. S. and Boerwinkle, E. (2005) Mining genetic epidemiology data with Bayesian networks I: Bayesian networks and example application (plasma apoE levels). *Bioinformatics*, **21**(15), 3273–3278.

Spirtes, P., Glymour, C. and Scheines, R. (1993) *Causation, Prediction and Search*, Springer, New York, NY.

Spirtes, P., Glymour, C., Scheines, R., Meek, C. and Richardson, T. (1993) *Tetrad 3 Tools for Causal Modeling User's Manual*. Available at http://www.phil.cmu.edu/projects/tetrad.

Biographies

Jing Li is currently a Ph.D. student in the Department of Industrial and Operations Engineering at the University of Michigan. She received her M.A. degree in Statistics from the University of Michigan in 2006. Her research interests include data fusion and causal discovery for optimal decision making such as monitoring, detection, diagnosis and control, with application to manufacturing and healthcare.

Jianjun Shi received his B.S. and M.S. in Electrical Engineering from the Beijing Institute of Technology in 1984 and 1987 respectively, and his Ph.D. in Mechanical Engineering from the University of Michigan in 1992. He is currently a Professor in the Department of Industrial and Operations Engineering and the Department of Mechanical Engineering at the University of Michigan. His research interests lie in the fusion of advanced statistics and engineering knowledge to develop in-process quality improvement methodologies to achieve automatic process monitoring, diagnosis, compensation, and their implementation in various manufacturing processes. He has received numerous awards and currently serves as a Department Editor of *IIE Transactions on Quality and Reliability*, and an Associate Editor of *International Journal of Flexible Manufacturing Systems*. He is a member of IIE, INFORMS, ASME, ASQC and SME.