

3-17-2005

## Exploring Causal Influences

Eric M. Neufeld  
*University of Saskatchewan*

Sonje K. Kristtorn  
*University of Saskatchewan*

Qingjuan Guan  
*University of Saskatchewan*

Manon Sanscartier  
*University of Saskatchewan*

Colin Ware  
*University of New Hampshire, Durham, colin.ware@unh.edu*

Follow this and additional works at: <https://scholars.unh.edu/ccom>

 Part of the [Oceanography and Atmospheric Sciences and Meteorology Commons](#)

---

### Recommended Citation

Eric M. Neufeld ; Sonje K. Kristtorn ; Qingjuan Guan ; Manon Sanscartier and Colin Ware "Exploring causal influences", Proc. SPIE 5669, Visualization and Data Analysis 2005, 52 (March 17, 2005); doi:10.1117/12.588790; <http://dx.doi.org/10.1117/12.588790>

This Conference Proceeding is brought to you for free and open access by the Center for Coastal and Ocean Mapping at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Center for Coastal and Ocean Mapping by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact [nicole.hentz@unh.edu](mailto:nicole.hentz@unh.edu).

# Exploring causal influences

Eric Neufeld<sup>\*a</sup>, Sonje Kristtorn<sup>a</sup>, Qingjuan Guan<sup>a</sup>, Manon Sanscartier<sup>a</sup> and Colin Ware<sup>†b</sup>

<sup>a</sup>Department of Computer Science, University of Saskatchewan, Saskatoon, SK S7K 5A9

<sup>b</sup>University of New Hampshire Center for Coastal and Ocean Mapping Durham, NH, 03824

## ABSTRACT

Recent data mining techniques exploit patterns of statistical independence in multivariate data to make conjectures about cause/effect relationships. These relationships can be used to construct causal graphs, which are sometimes represented by weighted node-link diagrams, with nodes representing variables and combinations of weighted links and/or nodes showing the strength of causal relationships. We present an interactive visualization for causal graphs (ICGs), inspired in part by the Influence Explorer. The key principles of this visualization are as follows: Variables are represented with vertical bars attached to nodes in a graph. Direct manipulation of variables is achieved by sliding a variable value up and down, which reveals causality by producing instantaneous change in causally and/or probabilistically linked variables. This direct manipulation technique gives users the impression they are causally influencing the variables linked to the one they are manipulating. In this context, we demonstrate the subtle distinction between *seeing* and *setting* of variable values, and in an extended example, show how this visualization can help a user understand the relationships in a large variable set, and with some intuitions about the domain and a few basic concepts, quickly detect bugs in causal models constructed from these data mining techniques.

**Keywords:** causality, causal perception, interactive causal graph, visualization, multivariate data.

## 1. INTRODUCTION

Tweedie *et al* [17] describe an engineering design tool they call the Influence Explorer (IE) that enables interactive, visual exploration of relationships between parameter and performance values of an engineering artifact. The fundamental model for a design task is a set of equations that give performance measures as functions of parameter values. Tweedie *et al* illustrate the parameter-performance relationships as a directed bipartite graph (Figure 1). IE uses this mathematical model to generate an exploration database, a subset of the design space consisting of more than 500 randomly generated points from a user-selected region of parameter space. Each point represents a design instance, that is, it comprises values for each of the parameter and performance variables. The frequency distributions of the parameter and performance variables over this data are represented as histograms.

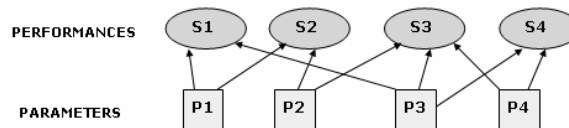


Figure 1: Performance-parameter correspondence in the Influence Explorer (adapted from [17])

Figure 2 shows the IE interface. Performance histograms are displayed horizontally on the left and parameter histograms are shown vertically on the right. The white bars show variable frequencies for the entire exploration database. Sliders beneath each variable histogram are used to limit the range of the variable, in effect selecting a subset of the data. The white circles indicate the mean of the selected values for each variable. In Figure 2, a range of values for the performance variable S4 has been selected. (All values of the other variables are selected by default.) Black histogram bars show the frequency distributions of the variables over the selected subset of the data. The user can

---

\* eric@cs.usask.ca

† cware@ccom.unh.edu

interactively explore relationships between parameter and performance variables by moving the sliders around and observing the effects on the black histograms.

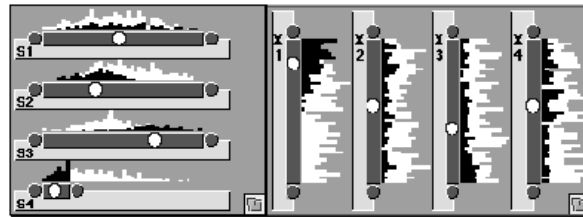


Figure 2: The Influence Explorer Visual Interface

On the one hand, the IE interface reflects the specific graphical topology of the engineering design problem. On the other, it abstracts the ontology of engineering design, which even in specific domains can contain many kinds of objects, to just two salient entities – parameter variables and performance variables. IE uses a bipartite graph to define the cause/effect relationships between parameter and performance values that problem solvers presumably have in mind while exploring design space. Moreover, IE’s approach is essentially statistical. Users select a desired range of performance and parameter values and IE computes, by summing over the appropriate subpopulation of the exploration database, new distributions for all variables.

Semantically, this is similar to computing conditional probabilities or conditional expectations, and it occurred to us that this approach would be useful in almost any domain where relationships are statistical. Thus, we further generalized the IE abstraction. First, we eliminated the distinction between parameters and performances and replaced both with generic variables. Any variable is a performance indicator for its in-directed neighbours and a parameter for its out-directed neighbours. Similarly, we generalized the simple bipartite model in Figure 1 of the engineering design problem to arbitrary directed graphs in more general domains. (Though these are restricted to acyclic graphs here, as are IE’s models, it is possible to extend the present ideas to certain cyclic graphs, but that is beyond the scope of this presentation.) As in the IE problem domain, arcs represent cause-effect relationships. The result is a graphical visualization that lets us extend the elegant core simplicity of IE to the representation of cause and effect in a large and general class of interesting statistical datasets.

Figure 3 illustrates an instance of the kind of problem we have in mind. It shows a path diagram for success in academic publishing, as measured by publications and citation rate. The graph was based on data from 162 academics [12]. The path model incorporates seven variables: ABILITY, SEX, GPQ (Graduate Program Quality), QFJ (Quality of First Job), PREPROD (Pre-PhD Production rate), PUBS (Publication rate), and CITES (Citation rate).

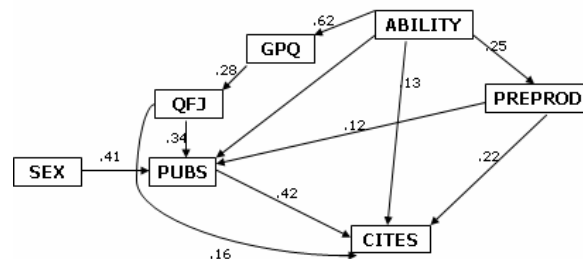


Figure 3: Path analysis example from Rodgers and Maranto [12]

This graph was constructed using *path analysis* [19] by a program called LISREL. The path coefficient on each path is a standardized regression coefficient between the connected variables.

Figure 4 shows another example, this time from the *structural equation modeling* (SEM) literature. In structural equation modeling (SEM), social scientists model a domain with a hierarchy of regression equations or a path diagram that defines the relationships between the domain variables. The path diagram in Figure 4 depicts a SEM model for data collected by Holzinger and Swineford [6]. The data consisted of scores for 6 intelligence tests for 63 girls. The rectangles represent observed variables – scores for the 6 intelligence tests. The ovals represent latent or unobserved variables. These include error terms for each of the test scores and two hypothesized latent variables – spatial ability and verbal ability. The double-headed arrow between these last indicates correlation with no causal attribution. Estimates for

the parameters of the model, including path weights and variances and covariances for latent variables, are derived from the data by various statistical methods. How well the model fits the data can then be assessed using a range of statistical tests. The model may be modified in light of this assessment and the process repeated; SEM is an iterative process.

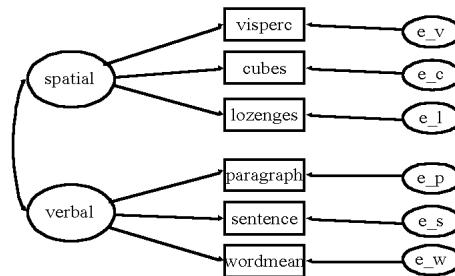


Figure 4: Structure Equation Modeling

These graphs share basic structural properties with the graph underlying IE—they are both directed and acyclic. Moreover, both trends make causal interpretations of their arcs, as is the case in IE although this is not explicit. The causality is transitive, in the sense that the effect of one variable might be the cause of another. (We do not discuss undirected arcs in the present work.) One could imagine generalizing instances from a statistical model space corresponding to these diagrams and providing interfaces for these models that are very similar to the IE interface. However, the number of variables would not have to become very large before the interface became confusing and unwieldy. Therefore, we make a further simplification. We take the histogram distribution of the IE interface, approximate it with a normal distribution, and project it onto the nodes of a node-link diagram. Variable distributions are projected onto nodes as follows: Each node is a narrow rectangle, where the center of the bar is labeled with the prior expected value of that variable. The visualization represents a standard deviation as a shaded region above and below the mean. The distribution of the variable is normalized so that the bar represents three standard deviations from the prior expected value. This conveys approximately the same information as IE does. As in the preceding examples, variables are connected by directed arcs to indicate direct causality. Tversky *et al* [16] suggest this is a natural interpretation for an arrow.

Direct manipulation of variables is achieved by sliding values up or down. This produces instantaneous changes in linked variables that indicate quantitative probabilistic and causal relationships. Although causality is difficult to define, we are most interested in understanding the consequences of interventions that are difficult to test experimentally, e.g., whether fluoridation of the water supply causes a decrease in cavities.

This is consistent with our most immediate impressions of causality, which in everyday life are associated with the direct manipulation of objects. When we manually move an object in our environment, we correctly perceive that we cause its displacement. This feeling of direct control is considered to be a central principle of contemporary direct manipulation interfaces [18]. We move a computer mouse and through a complex train of events, a cursor or other object moves on the screen or a window is resized. Because of the immediacy of the visual change corresponding to the hand movement, we feel that we have directly manipulated something even though there is nothing direct about it. The key requirements for this illusion of direct manipulation are rapid visual feedback (< 100 msec.) and a natural visual relationship between the user's action and the onscreen event, e.g., we move our hand left, the object moves left. Though the optimal correspondence is by no means known, the result of successful direct manipulation is a feeling of transparency such that "the user is able to apply intellect directly to the task; the tool itself disappears" [13]. It is not true that any onscreen change concomitant with a user action is perceived as direct manipulation and the circumstances under which the illusion is achieved are likely to be complex. But certainly object movement linked to hand movement is most likely to promote the illusion.

In IE, the user interactively selects a variable range and immediately sees the corresponding changes in the histograms. This enables a rapid exploration of the design space but, because of the rather complex pattern of change in the IE histograms, it is unlikely that there is a strong feeling of direct manipulation. We believe that the simpler node-link visualization we describe provides this kind of perception of causality. When the user manipulates a variable, the subsequent changes have a clear causal structure. The user can see the overall model as well as responses of the model to changes in variable values.

## 2. CAUSALITY AND DIMENSIONALITY

A key motivation for information visualization is the so-called curse of dimensionality. As the examples of Figures 3 and 4 show, many datasets have dimensionality greater than 2 or 3. Although the human eye can use differences in colour, movement or size to distinguish objects, employing too many of these techniques may result in a noisy and cluttered visualization. Yet collapsing of dimensions using clustering techniques results in loss of information.

The goal of many data mining and data visualization techniques is to help the user discover and understand key cause-effect relationships. Indeed, this motivates many studies, despite the common caution against interpreting correlation as causation—researchers want to know the cause (or predictors) of cancer, the reasons for retention of undergraduate students, or the likely outcome of a marketing or campaign strategy. We tackle dimensionality here by explicitly representing cause and effect. Sometimes, cause-effect relationships are known at the outset, as in the case of IE, and the user mainly wishes to use the tool to quantitatively explore the consequences of observations and actions. This works well in cases where each variable has relatively few direct causes. For instance, in IE, we would not expect every parameter to directly influence every performance. When the cause-effect relationships are not known, we can use data mining tools like TETRAD to construct plausible causal models from raw data. In such cases, our visualization tool, coupled with human intuitions, can be used to confirm, refute, or debug aspects of such models, as we show below.

The success of this representation depends on an assumption that the number of cause-effect relationships in a dataset is small relative to the number of variables. A typical objection is that almost all variables are related to each other. But causal graphs can be simplified by explicitly representing only direct causes. Formally, if the number of direct causes of any variable is bounded by some constant independent of  $n$ , the number of nodes, the number of relationships to be explicitly represented goes from  $O(n^2)$  to  $O(n)$ . This permits a significant simplification, without collapsing variables away, and indirect relationships can be inferred from arrow sequences, so no information is lost. In fact, if the domain is not amenable to this simplification, (that is, if each variable is tightly bound to every other variable in direct cause-effect relationships), it is unlikely that any visualization technique will fare better.

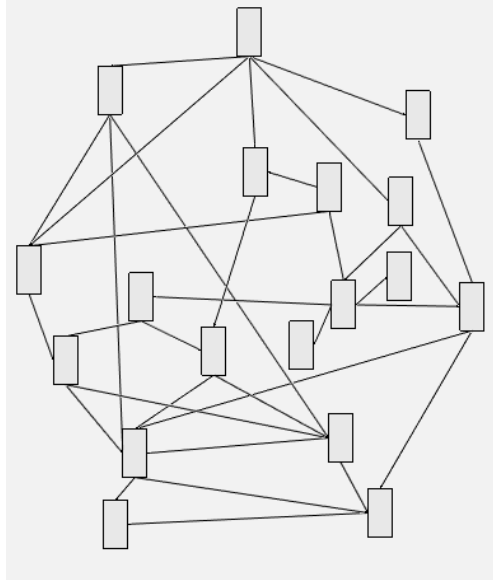


Figure 5: Student retention model from Druzdel and Glymour [4] showing only cause-effect relationships.

Figure 5 illustrates a medium view of a large set of variables from an earlier version of our implementation. Though the graph contains 18 variables, the number of direct causal relationships is relatively small.

The graph is busy, but not hopelessly so. Our implementation also incorporates a semantic zoom [1,2], allowing the user to zoom in and see quantitative relationships, as shown in Figure 6.

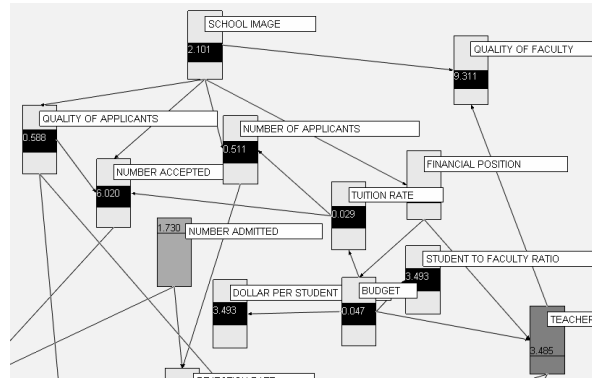


Figure 6: Semantic Zoom close shot (detail). Expected values of variables, along with indications of standard deviations, are instantly propagated through the network as nodes are manipulated. The blue node indicates an observed variable and the red node indicates a set variable.

As mentioned earlier, the user can select the mean of any variable and drag it with the mouse. In the example shown in Figure 6, changes propagate instantly. Variables causally proximate to the manipulated variables are most affected by changes, but this animation allows the eye to easily detect causal paths, and stronger relationships.

A trivial example of how direct causality assists the computation is given by the three variable graph shown in Figure 7. This model represents a domain where *rain* makes streets *wet* and where *wet* streets result in *slippery* conditions. Knowing whether or not *rain* has occurred allows us to compute the probability it is *slippery*, even if we cannot observe whether it is *wet*. However, if the street is *wet*, the probability it is *slippery* is independent of *rain*, or any other cause (indicated by ellipsis). The cause of *wetness* is not relevant to *slippery* once the outcome of *wet* is known. This does not limit the expressiveness of the graphical representation. Should *rain* impact *slippery* directly, we can always add an arrow from *rain* to *slippery* but at extra cost of readability and computation.

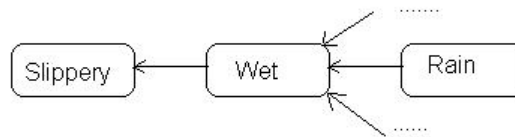


Figure 7: The rain-wet-slippery graph

This sort of calculation is efficiently handled by a Bayes net [10] computational engine. For our implementation, we used the method of Lauritzen and Jensen [7] for systems of linear Gaussian variables. Their fast and stable algorithms allow instant propagation of conditional expectations throughout the net diagram. They perform well, even on graphs of more than 35 variables.

These algorithms derive their effectiveness from the independence relationships implied by the relationships of direct causality. That is, given fixed outcomes for all of its direct causes (parents), a variable is independent of the outcomes of any predecessor causes. This permits a divide and conquer approach to a computation that is otherwise always exponential. (For example, in the worst case of discrete variables, all probabilities must be computed by summing up over the set of all joint outcomes, which is exponential.) Neufeld and Guan [8] give some intuitions behind the computational efficiency of this algorithm, but also see, for example, Pearl [10] for a deeper introductory discussion. Lauritzen and Jensen's method uses sophisticated data structures to further optimize, but we do not give details here. In particular, although our implementation limits all relationships to linear Gaussians, this is not a limitation of the visualization framework itself. If the relationships in the data are more complex, the inference engine can be modified.

Thus, conditional expectations based on regression models can be replaced with neural net relationships or relationships discovered with genetic algorithms, so long as there is an associated inference engine that can propagate changes in expectations in both directions. The visualization framework is a tool to help users understand cause and effect relationships; it does not presuppose a commitment to a particular algorithm.

The power of this paradigm is that we can represent arbitrarily complex relationships if we are prepared to pay the cost of less readability and less computational efficiency. However, where the domain permits, we can simplify the representation and achieve both readability and computational advantages. Furthermore, this can be done without loss of information.

### 3. SEEING AND SETTING

Dennis *et al* [3] report an interesting problem regarding sunscreen use and melanoma. Though most doctors believe that sunscreen use prevents melanoma, some reports suggest that sunscreen use increases the risk for melanoma. However, these reports don't take into account different sunscreen usage rates by light-skinned people, who are more susceptible to melanoma, and by dark-skinned people, who are less susceptible to melanoma in the first place. Figure 8 shows the qualitative causal relationships involved between sunscreen use (*S*), melanoma (*M*) and darkness of skin (*D*), but does not indicate the sign or magnitude of the causal relationship.

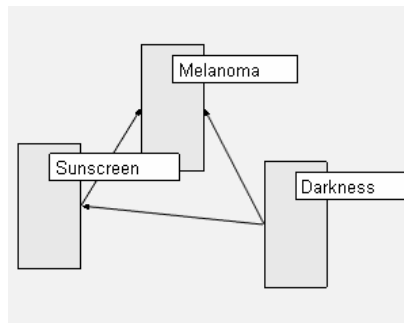


Figure 8: Qualitative causal model of sunscreen, melanoma and darkness of skin

In the sequel, we are interested in two basic inference problems in multivariate datasets constructed from observational data; we call these *seeing* and *setting*. (Freedman [5] uses the terms *observation* and *intervention*.) *Seeing* is the common procedure of computing a conditional expectation—given a subpopulation of a known population, what is the posterior distribution of all related variables in the subpopulation? For example, we may wish to compute  $p(M/see(S))$ , the probability that someone we *see* using sunscreen might develop melanoma, or we might also be interested in the inverse probability  $p(S/see(M))$ , the probability that someone with melanoma uses sunscreen. These are straightforward probabilistic inferences, routinely handled by any probability inference engine, and for any variables,  $p(X/see(Y))=p(X/Y)$ . (The reason for using the *see* functor will become apparent.) Thus, computing the consequences of an observation is equivalent to computing an ordinary conditional probability. For this illustrative example, we use discrete probabilities, but the argument generalizes to continuous variables.

Someone who hears about the sunscreen studies described above wants to know whether to use sunscreen. The probabilistic relationships alone do not reveal the best course of action. This is where we need the distinction between *seeing* and *setting*.

Whereas *seeing* is about the consequences of observations, *setting* is about the consequences of actions (or, as Freedman would say, interventions) and requires a subtler calculation. An individual wants to know the net effect of using sunscreen, for that individual, whether sunscreen use decreases or increases the overall probability of melanoma, in light of conflicting opinions. The probability of interest is  $p(M/set(S))$ , where we use the *set* functor to distinguish *setting* from *seeing*. People expect the probability of *setting* to be the same as the probability of *seeing*, namely, the conditional probability  $p(M|S)$ . In this discrete formulation, *D* indicates dark skin and  $\neg D$  indicates light skin. If we use  $p(M|S)$  to compute our target probability, using the ordinary axioms of probability, we get

$$P(M/set(S))=p(M|S)=p(M|DS)p(D/S) + p(M/\neg DS)p(\neg D/S).$$

But this formulation is wrong! Why? Because although  $P(M|DS)$  correctly counts the incidence of melanoma among dark-skinned sunscreen users, multiplying it by  $p(D|S)$  will not predict the posterior incidence of melanoma among dark-skinned sunscreen users in the new scenario, since it does not count dark-skinned persons currently not using sunscreen, who will use it in the new scenario. Thus,  $p(M|DS)$  should be multiplied by the proportion of all persons who

are dark-skinned. A symmetric argument applies to the second multiplicative term, and thus, the complete correct formulation is

$$P(M|\text{set}(S)) = p(M|DS)p(D) + p(M|\neg DS)p(\neg D).$$

When  $D$  and  $S$  are independent,  $p(M|\text{set}(S)) = p(M|\text{see}(S))$ , but otherwise they are different. (In the restricted topology of the IE, there is no practical need for explicating this distinction.) Although this formulation goes back to Stotz and Wold [15], it has a very simple and elegant implementation in terms of our causal graphs. We just erase arcs incoming to the set variable, and change all distributions involving the set variable accordingly. For correctness, see Pearl [11]. We believe this visualization is the only causal graph implementation that lets users distinguish between seeing and setting, which makes it useful in policy prediction. Below, we give an extended example to show its usefulness.

#### 4. DETAILED SUNSCREEN EXAMPLE

Neufeld and Guan [8] demonstrate basic functionality of ICGs. Here we would like to demonstrate how ICGs can help users to explore and debug causal models with counterintuitive relationships. This is more so when the user has become comfortable with the meaning of *setting*, but even when this is not the case, *setting* can still help the user to understand relationships in the domain.

In this example, we extend the sunscreen example of the preceding section to continuous variables. We therefore assume some continuous methodology for representing usage of sunscreen, incidence of melanoma, and skin type. This is because our implementation presently only handles continuous variables. For purpose of illustrating the usefulness of ICGS, we will suppose that reports showed that sunscreen had *no* effect on incidence of melanoma. (This is almost as surprising as the actual reports, and lets us extend the example somewhat.) This data is typical of pedagogical examples sometimes used to illustrate the concept of suppressors in social science courses. Our contrived model corresponds to intuitions, but with a twist. That is, people believe that sunscreen has a negative effect on melanoma, and that melanoma decreases with darkness of skin. In our model, these causal relationships hold, but the twist is that heavy users of sunscreen tend to have lighter skin. This is reasonable, but we have stretched the data so far that in the aggregate data, sunscreen use appears to have *no* effect on melanoma. It would be possible to stretch the example even further, so that sunscreen usage appears to have a positive effect on melanoma in the aggregate data, consistent with the reports, which would provide a continuous instance of Simpson's Paradox [9], where associations in aggregate data reverse themselves in subsets.

Simpson's Paradox is not about the domain, but about the numbers. It is easy enough to contrive a domain where sunscreen does in fact have a positive causal influence on melanoma, for instance, if sunscreens, although preventing exposure to UV radiation, were carcinogenic in other ways. So let's avoid this issue by supposing that, although the aggregate data appears to conflict with expectations, sunscreen is in fact not carcinogenic.

##### 4.1 Debugging the model

We begin by supposing that the user uses a data mining tool such as TETRAD [14] to construct a first cut of the causal relationships in a set of data.

TETRAD works by using conditional independencies in the data to construct causal relations. The following is an oversimplification, but gives the basic idea: Suppose that in some context, variables  $A$  and  $B$  are independent, but that both are associated with  $C$  in all contexts. Tetrad uses the fact that there is always an association between  $A$  and  $C$  to draw an edge from  $A$  to  $C$ , and similarly from  $B$  to  $C$ . Since there is no association between  $A$  and  $B$ , they are not connected. Next, TETRAD decides how to orient the edges. Combinatorially, there are only four possibilities, which we might write as  $A \rightarrow C \rightarrow B$ ,  $B \rightarrow C \rightarrow A$ ,  $B \leftarrow C \rightarrow A$ ,  $B \rightarrow C \leftarrow A$ . The first two possibilities would imply an association (correlation) between  $A$  and  $B$ , since the causal relationship is transitive. The same is true of the third possibility—effects of a common cause are normally correlated. The only possible orientation of arcs that does not contradict the independence of  $A$  and  $B$  is  $B \rightarrow C \leftarrow A$ .

We have intentionally fooled TETRAD, using the knowledge that it builds causal models from statistical independencies, by hiding an spurious independency in the data that is not due to "true" independence but arises from two causal influences canceling each other out. TETRAD will generate the model shown in Figure 9.



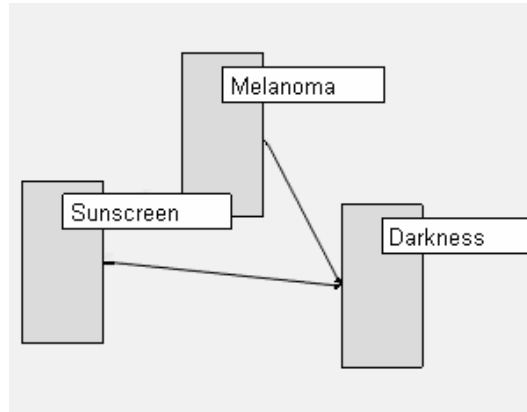


Figure 9: First TETRAD Model of Sunscreen Problem

The model shows a world where *melanoma* and *sunscreen* are independent common potential causes of *darkness* of skin, and mathematically is a minimal representation of possible causal relationships in a world coherent with the raw data. (In this model, low values for *darkness* mean lighter skin.) Although this model is graphically faithful to the independencies in the distribution, the user should immediately recognize that the model is incorrect because of the nature of the variables involved. (In domains where the variables are not so well understood, the user can use the model to see what predictions it makes about interventions, and test these predictions in the actual domain. This would involve experimenting with *setting*, which we discuss in a more practical setting below.)

In any event, suppose the user eventually constructs the correct model of causal relationships shown in Figure 10.

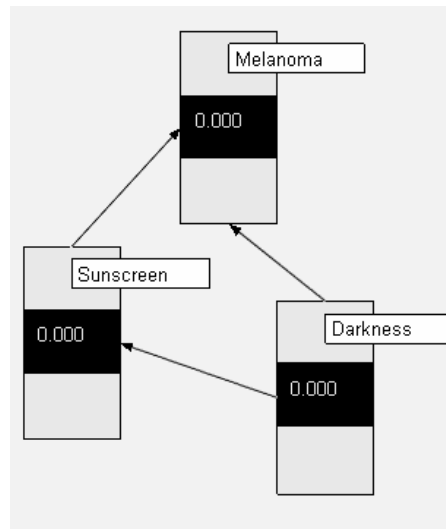


Figure 10: Sunscreen-Melanoma-Darkness graph before manipulation.

The user has also built a coherent hierarchy of regression equations for all variables using normalized values. Figure 10 shows initial expected values and standard deviation of all variables. Having constructed the right model, the user wishes to explore the relationship between *sunscreen* and *melanoma*.

## 4.2 Exploring causal influences with seeing

Figure 11 shows the results of the user's first explorations with seeing. The user grabs the value of the sunscreen variable, drags it up and down, and finds from the contrived data that sunscreen and melanoma are unrelated! Changes to sunscreen change darkness significantly, but melanoma barely moves. (This is not a surprise, since TETRAD found those two variables independent.)

The statistical explanation is that darkness is a suppressor variable or confound. That is, light-skinned individuals are more likely to use sunscreen, but they are also more likely to develop melanoma than dark-skinned people.

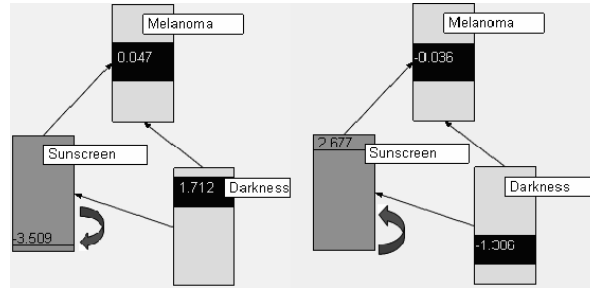


Figure 11: Melanoma does not respond much to sunscreen. Added red arrows indicate mouse movement during interaction

An experienced data analyst would revisit the data and ask what happens when *sunscreen* is manipulated after *darkness* is first fixed at some value. This data analysis technique is easy to do with an ICG. Figure 12 illustrates what needs to be done.

The user must first choose to *see* a fixed value for the *darkness* variable, and then *see* a range of values for *sunscreen* by dragging it up and down. The pairs of before and after images in Figure 12 now reveal the correct relationship between *sunscreen* and *melanoma*. Whether *darkness* is high or low, fixing *darkness*, and then increasing *sunscreen* results in a decrease to *melanoma*. Manipulating a variable by *seeing* a value corresponds to computing conditional expectations of all other variables in the domain, given the observed value. So when the user fixes *darkness*, expected values for *sunscreen* and *melanoma* are calculated based on the selected value for *darkness*. When the user then sees different values for *sunscreen*, *melanoma* responds as expected. Moreover, the same result will obtain for any values the user chooses for the *darkness* variable.

The reason the user needs to perform a double *seeing* operation is that there are two paths of probabilistic influence from *sunscreen* to *melanoma* that cancel each other out. One is the direct unit path, and the other is the indirect path from *sunscreen* to *darkness* to *melanoma*. This is all the more confusing because this probabilistic path goes “against the arrows”. That is why, in real world examples, a zero net effect, or net effect of the wrong sign, is so puzzling. By fixing the *darkness* variable, the positive probabilistic path of influence from *sunscreen* to *melanoma* via *darkness* is effectively blocked, and the ICG can reveal the relationship correctly.

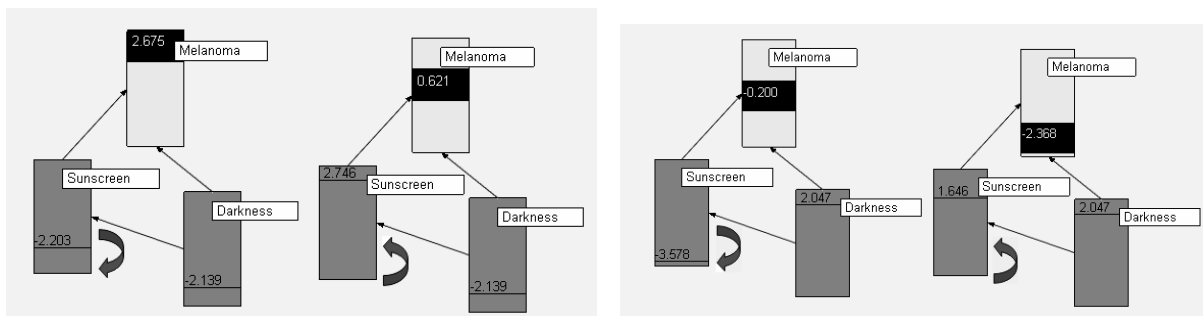


Figure 12: Melanoma responds to seeing changes in sunscreen within a particular skin type. The top pair of graphs shows the change in melanoma with sunscreen for light-skinned persons, and the bottom pair shows melanoma change for dark-skinned persons. Ranges differ, but melanoma incidence consistently decreases with increased sunscreen usage.

Because there are few potential confounds in this three node world, trying all see operations is not logistically difficult. However, in a richer dataset, this process may be cumbersome. This is where setting becomes important.

#### 4.4 Exploring causal influences with *setting*

If the user simply *sets* the *sunscreen* variable, *melanoma* responds appropriately, and more importantly, consistently with the cause-effect relationship that actually exists in this (contrived) world. The before/after graphs in Figure 13 show the visualization.

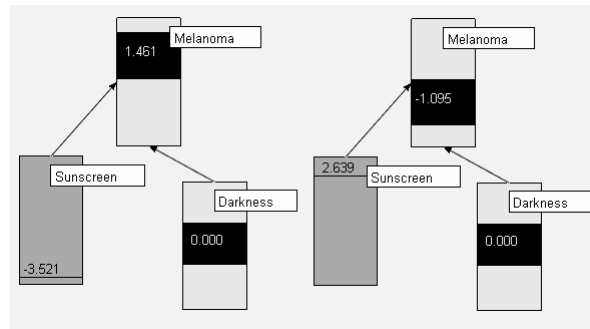


Figure 13: Melanoma responds appropriately to setting of the sunscreen variable.

To understand why this happens, revisit the semantics of setting as given in Section 3. The definition of *setting* cannot be derived mechanically from the axioms of probability, but rather uses probabilities from a known world in a novel way to make predictions about an unknown world resulting from an intervention. In terms of probabilistic influences, *setting* accomplishes the same thing as the double *seeing* operation by canceling the positive probabilistic influence *sunscreen* has via the path through *darkness*. Recall that in Section 3, we mentioned that this method has a simple and elegant interpretation in terms of the graphical models—the arc between *darkness* and *sunscreen* is erased, eliminating the positive causal path explicitly. (As a cue to the user, we show the arc disappearing in the visualization.)

## 5 DISCUSSION

We have generalized the ideas behind the Influence Explorer to almost arbitrary statistical domains. Where IE performs an essentially statistical strategy by summing over a subpopulation of a set of randomly generated model instances, we propagate statistical inferences efficiently by borrowing the Bayes net technology [7, 10]. We further simplify IE by projecting distributions onto a graphical representation of the domain, so that the user can maintain a graphical model of cause and effect while observing the consequences of actions. Our visualization of variable manipulation is linked to the deep notion of causality as the ability to predict the consequences of an intervention, but, just as IE abstracts engineering artifacts over domains, we abstract causality over domains as a directed graph. Where IE implicitly gives its users the ability to understand the consequences of seeing and setting different types of variables, our visualization makes this explicit.

The separate notions of *setting* and *seeing* come from Pearl [11]. *Setting* is presently a little known concept: the ideas behind it are not widely known even within the structure equation community, and we do not expect the idea to be absorbed quickly by statistically inexperienced users. However, because we believe our visual metaphor for *setting* is an excellent abstraction of causal effect, a visualization of this type has the potential to rapidly educate users who may not be statistically sophisticated about the kinds of subtle interactions variables may have, just as pie charts and bar graphs have proved to be vivid expressions of probability distributions to a wide audience.

The main contribution of this work is to demonstrate the feasibility of performing principled calculations quickly enough to allow real-time animations of both *seeing* and *setting* on graphs of modest size. The examples here (including the graphs of Figures 3, 5 and 6) performed well on a 2Ghz desktop machine. In ongoing work, we are refining the graphical presentation. Figure 14 shows a visualization of the graph that uses normal curves rather than vertical bars, as well as ghosts of the prior distributions. This more completely accomplishes the goal of projecting the prior and posterior distributions onto the nodes in the style of IE.

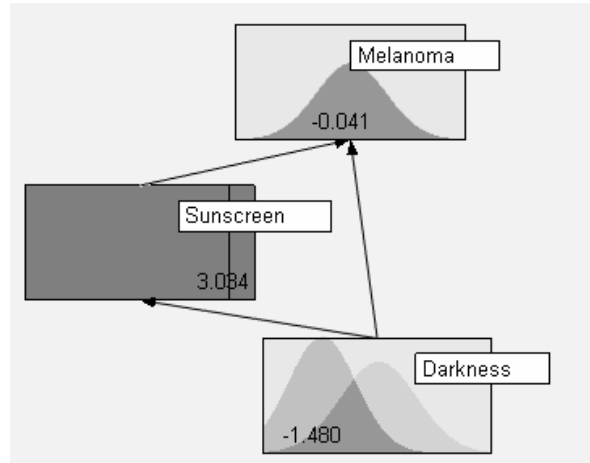


Figure 14: Bell curves represent distributions and ghosting is used to present both prior and posterior distributions.

This work did not discuss handling of cycles (systems with feedback) or undirected edges in path analysis, which indicate the presence of latent variables, i.e., hidden common causes. These problems are likewise the subject of current work.

## ACKNOWLEDGMENTS

The work of the first author was supported by a Discovery Grant from the Natural Science and Engineering Research Council of Canada. The bulk of the implementation was done as part of Qingjuan Guan's MSc thesis at the University of Saskatchewan, and that is where most of the ideas about visualizing *seeing* and *setting* began. Manon Sanscartier, PhD student at the University of Saskatchewan, improved the implementation.

## REFERENCES

1. G.W. Furnas and B.B. Bederson. Space-scale Diagrams: Understanding Multiscale Interfaces. Human Factors in Computing Systems, *CHI '95 Conference Proceedings*, ACM Press, pp. 234-241, 1995.
2. G.W. Furnas and X. Zhan. MuSE: A Multi-Scale Editor. *Proceedings of the UIST'98 Symposium*, ACM Press, 1998.
3. L.K. Dennis, L.E. Beane Freeman, M.J. VanBeek. Sunscreen use and the risk for melanoma: a quantitative review. *Annals of Internal Medicine*:139(12):966-7, 2003.
4. M.J. Druzdel and C. Glymour. Causal inferences from databases: Why universities lose students. In *Computation, Causation, and Discovery*, C. Glymour and G.F. Cooper (eds), AAAI Press, Menlo Park, CA 521-539, 1999.
5. D.A. Freedman. From association to causation via regression. In *Causality in crisis: statistical methods in the search for causal knowledge in the social sciences*, V.R. McKim and S.P. Turner (Eds).. South Bend: University of Notre Dame Press, 113-161, 1997.
6. K. J. Holzinger and F. Swineford. A study in Factor Analysis: The Stability of a Bi-Factor Solution. *Supplementary Educational Monographs*. Chicago, Ill.: The University of Chicago. 1939.
7. S.L. Lauritzen and F. Jensen. Stable local computation with conditional Gaussian distributions. *Statistics and Computing* 11, 191-203, 2001.
8. E. Neufeld and Q. Guan. Visualizing Multivariate Data with Cause and Effect Relationships. To appear in *Proceedings of 3IA 2004, International Conference on Artificial Intelligence and Graphics*, 2004.
9. E. Neufeld. Simpson's paradox in artificial intelligence and real life. *Computational Intelligence* 7:1 1-10, 1995.
10. J. Pearl. Fusion, Propagation, and Structuring in Bayesian Networks. *Artificial Intelligence*, Vol. 28, pp. 241-288, 1986.
11. J. Pearl. *Causality: Models, Reasoning, and Inference*. New York: Cambridge University Press, 2000
12. R. Rodgers and C. Maranto. Causal models of publishing productivity in psychology. *Journal of Applied Psychology*, 66:688-701, 1989.
13. C. Rutowsky. An Introduction to the Human Applications Standard Interface, Part 1: Theory and Principles. *BYTE* 7(11): 29-310, 1982.
14. P. Spirtes, C. Glymour and R. Scheines. *Causation, Prediction and Search*. Springer-Verlag, New York, 1993.
15. R.H. Stotz and H.O.A. Wold. Recursive versus nonrecursive systems: An attempt at synthesis. *Econometrica* 28, 417-427, 1960.
16. B. Tversky, J. Zacks, P. Lee and J. Heiser. Lines, Blobs, Crosses and Arrows: Diagrammatic Communication with Schematic Figures. *Diagrams* 2000: 221-230, 2000.
17. L. Tweedie, R. Spence, H. Dawkes and S. Hua. Externalising abstract mathematical models. In *Proc. of ACM CHI '96* (April 13-18, Vancouver, Canada), ACM Press, 406-412, 1996.
18. C. Ware. *Information Visualization: Perception for Design*, Morgan Kaufman, San Francisco, 2000.
19. S. Wright. Correlation and causation. *Journal of Agricultural Research* 20: 557-585, 1921.