Talking About Data: Web Text Can Aid Causal Structure Learning

Daniel J. Butler and Mausam
Department of Comp. Science and Engineering
University of Washington, Seattle, WA
{djbutler, mausam}@cs.washington.edu

ABSTRACT

In order to learn causal models from data, human prior knowledge is often required to constrain causal structure. In practice a domain expert often supplies commonsense or factual knowledge (e.g., “smoking causes lung cancer” or “number of cylinders affects gas mileage”) to be used alongside data in the learning algorithm. Unfortunately, this human expertise is hard to get — an expert may be costly and for large domains the cost of entering such knowledge may be prohibitive.

We ask the following question: can the vast knowledge present in text be used to supplement causal structure learning by supplying cues similar to those of a human expert?

We introduce a technique for combining knowledge extracted automatically from text with numerical data to yield more accurate causal networks. Our method uses sentence-level facts extracted from the Web by the Open IE [9] system, allowing us to leverage recent advances in this burgeoning area.

We evaluate our system on a variety of open datasets, providing evidence that our hybrid approach yields networks that are more accurate than purely numerical approaches.

1. INTRODUCTION

There has been much interest in inferring causal relationships from observational data. Although correlation does not imply causation, there is a persistent intuition that the two are intimately, though subtly, related.

While controlled experiments and randomized control trials are still the gold standards for inferring causal relationships, observational studies have certain advantages that make them unlikely to go away any time soon. Benson and Hartz [4], discussing the role of observational studies in medicine, write:

> Observational studies have several advantages over randomized, controlled trials, including lower cost, greater timeliness, and a broader range of patients.

Many other authors have drawn similar conclusions [1] [31] [30].

In domains in which experiments are expensive or infeasible (domains such as public health, epidemiology, economics, and public policy, to name a few) observational data that on their own may be insufficient to make causal claims are typically combined with causal background knowledge to argue for particular causal conclusions [19] [11]. Some background knowledge is based on common sense (e.g. nothing causally influences the age of a person) or on prior scientific evidence (e.g. nicotine causes cancer in rats). In either case, if a human data analyst can encode his or her prior structural knowledge in the form of a causal graph, statistical methods can combine that graph with observational data to predict the effects of interventions on the system, for example how much lung cancer rates might decrease by if cigarette taxes were increased by a certain amount [24].

In the age of large, freely available datasets through services like Data.gov and the Linked Open Data ecosystem, the sheer quantity of data that could benefit from causal analysis has exploded.

This points to a “knowledge bottleneck” [23]: we want to automate the causal analysis of large datasets to pick out potential causal relationships, but in many domains where observational studies are prevalent, causal analysis typically relies on causal background knowledge [11].

We ask the following question: can the vast knowledge present in text be used to supply causal background knowledge similar to that of a human expert? If the answer is yes, then such a system could have access to far more background knowledge than any one human, and could be run on datasets with far more—and far more diverse—variables than a human could practically analyze.

Our approach relies on modern information extraction (IE)—a family of techniques that transform freeform text into structured tabular knowledge. We used knowledge extracted by the Open IE system ([9]), which produces extractions of the form (**subject**, **verb phrase**, **object**) (see Table 1 for examples). The database we used was created by the Open IE system run on several large web corpora, and it contains over 5 billion of extractions.

Previous work has shown that curated domain-specific data sources, such gene expression databases, can be useful for structure learning [2]. In contrast, our work uses knowledge extracted automatically from the web, a much larger and more diverse source of information, but also full of algorithmic errors, factual errors, and omissions. In addition, most prior work in causal structure learning has assumed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

```
Each 100 pounds of weight decreases fuel
efficiency by 1%.

This engine produces 160 horsepower, [...].
```

↓

argument 1	relation	argument 2
weight	decreases	fuel efficiency
engine	produces	horsepower

Table 1: Sample extractions from Open IE.

that background knowledge comes in the form of hard constraints (PC, Eberhardt, CITATION). As part of ongoing work, we have formulated an optimization schemes that integrates soft causal constraints and is therefore applicable to noisy automatic extractions.

Furthermore, most previous work on using text for structure learning has focused on data-predictive tasks like log likelihood or mean squared error, without regard to causal structure.[1]While this is useful for making predictions about, for example, whether a person with a given set of attributes has a particular disease, it is not useful for determining whether a particular factor causes a disease, or how a new law might affect some measurable outcome. A classic example of this distinction is medical symptoms: symptoms of a disease are strong predictors of the disease but do not cause it. For a more formal explanation of the difference between causal and non-causal models, see Appendix A.

2. BACKGROUND

Most approaches to learning causal structure from data are based on the idea of learning a Bayesian network. These methods fall broadly into constraint-based methods and score-based methods [24].

Constraint-based structure learning methods try to detect conditional independences in the data using various statistical tests, and then search for networks that obey these constraints. Some of them have the feature that they will mark the presence of certain edges as being ambiguous if there exist multiple Bayesian network structures that describe the data equally well but disagree on the presence of that edge, which may be an advantage for causal learning (on the basis that "don't know" is better than a wrong answer) [24].

One of the earliest constraint-based algorithms was the PC algorithm [28], which interleaves conditional independence tests with edge deletions to minimize the number of tests it has to perform. Other constraint-based methods, such as IAMB [32], pursue different search strategies. Recently, search based on SAT-solvers has been proposed [17].

Score-based algorithms optimize an objective function that combines the likelihood of the data with some complexity-penalizing prior which favors networks with fewer edges [24]. These methods include greedy hill-climbing, in which all single-edge additions and deletions are evaluated to see if any will increase the score, and this process is repeated until a local optimum is reached. Typically these methods output a complete Bayesian network in which all edges are oriented.

There have been successful applications where Bayesian network structure learning has been use to learn causal networks directly from data, particularly in the area of gene networks. Friedman et al [10] was one of the first groups to use a score-based optimization scheme to learn a causal Bayesian network describing a gene network. Maathuis et al [22] used a constraint-based algorithm based on the PC algorithm to learn a gene network in the high-dimension setting with many genes and few data points.

In addition to learning causal relationships from data, there has been work in extracting causal knowledge from text.

Girju and Moldovan [12] extracted cause-effect pairs from WordNet glosses and used these pairs to learn a set of causal verb phrases from the TREC-9 corpus. They followed a bootstrapping procedure: any verb phrase in the text connecting a known cause to a known effect was marked as causal, and these verb phrases were later used to do extraction.

Radinsky, Davidovich, and Markovitch [25] extracted causal relations from news headlines and used them to predict possible effects of a given event. For example, the algorithm predicted that the event "Magnitude 6.5 earthquake rocks the Solomon Islands" could cause the event "Tsunami-warning will be issued in the Pacific Ocean".

Most similar to our work, Sanchez-Graillet and Poesio [27] developed a system for constructing Bayesian networks using causal facts extracted from a source document. However, their system did not combine relate these facts to numerical data in any way.

In the realm of information extraction more generally, the Open IE that we use in the work distinguished itself from other systems in that it is not restricted to extracting a fixed set of relations [9] [3], making it well suited to applications like ours in which the relations of interest are not well represented in most knowledge-extraction systems. There are other web-scale IE systems such as NELL [5], for instance, which is based on an iterative bootstrapping scheme with weak human supervision. However, the NELL system does not appear to have causal knowledge relevant to our task.

The idea of combining data with external information to learn Bayesian networks has been around for a long time. Heckerman et al [14] and Castelo and Siebes [6] developed generic mathematical formulations for learning Bayesian network structures with prior probabilities defined on the space of all networks and parameters.

There has been work aimed at using text as a source of prior information to improve Bayesian network structure learning from data, mostly in the medical domain. Antal et al [2] use the concurrence statistics of medical terms in MEDLINE medical abstracts and a collection of journals selected by experts to learn an improved predictive model of ovarian cancer. Imoto et al [18] developed a Bayesian network model and optimization procedure for combining gene microarray data with prior knowledge to learn more accurate gene networks.

Along similar lines, there has been work combining knowledge elicited from a human expert with numerical data to learn better Bayesian networks. Richardson and Domingos [26], for instance, developed a model for incorporating noisy knowledge from multiple experts into a Bayesian network structure learning algorithm.

[1] Gene interaction networks are the exception, where the goal is not merely prediction but to uncover the true interaction network structure [33] [10].

3. OUR APPROACH

In our problem formulation, the input is a tabular dataset with m samples (which we also refer to as rows) and n variables (columns). Each column c_i comes with a text label s_i related to its semantic meaning. In the datasets we worked with, typical column labels are shown in Table 2. Some discrete variables take on values in a set of strings (for example, the variable “make” in the Automobile dataset can take on the values “audi”, “bmw”, etc). When available, these strings are also provided to the algorithm, but they are not required.

We also assume that we are given a knowledge base containing text extractions of the form given in Table 1, although these are permitted to be noisy or incomplete.

The desired output of the system is a directed graph where the presence of edge (i, j) indicates that variable c_i is a direct causal influence on variable c_j with respect to all the variables in the dataset $\{c_1 \ldots c_n\}$. We do not permit self-edges.

In our approach, we frame the problem as binary classification on each pair of columns (c_i, c_j). Our basic strategy for generating text features is to query Open IE for tuples related to the column labels s_i and s_j and count the results in various ways.

3.1 Entity linking and query expansion

The first challenge in carrying out this strategy is that facts in the knowledge base may not refer directly to the column labels s_i or s_j but rather to synonyms or semantically related terms. Since Open IE has a named entity linker that maps tuple arguments to Wikipedia entries, a natural approach is to map the column labels to Wikipedia entries as well, then query Open IE for tuples that refer to the same concepts. We include counts results from these types of queries as part of our feature sets.

However, we found that under such a scheme, there are relatively few matches between Open IE and the datasets. This is because the requirements for named entity linking are quite strict—two strings should only be linked to the same concept if there is a very high confidence that they refer to the same thing.

By contrast, in our setting we found it is acceptable to match quite loosely. For example, consider the causal relationship “fly ash → concrete compressive strength” (fly ash is an ingredient in concrete). It turns there are no tuples in Open IE relating the Wikipedia concepts **Fly_ash** and **Compressive_strength**, but there are text matches between “ash” and “strength” (which turn out to be correct causal tuples).

Consequently, in addition to search based on concepts, we also adopted a query-expansion approach in which we generated many possible text strings from each of the original column labels, including splitting all words apart into individual queries (we also tried just taking the head word, but the results were no better). An even richer set of queries was obtained by using the Cross Wikis dataset [29], which provides a huge collection of strings that people have used as anchor text on the web to link a Wikipedia article. We first use a Google search to map from column name to Wikipedia entry, then use the Cross Wikis dataset to generate the top five anchor text strings that link to that entry (filtering out certain words that commonly appear as noise in that dataset, such as “wikipedia”).

3.2 Causal text features

As stated earlier, we transform the Open IE query results into features by counting. We have two ways of counting — by instance and by tuple — and we apply each of these counting schemes to the query results filtered in different ways.

In Open IE, each tuple of the form (**argument1**, **relation**, **argument2**) is associated with a number of instances, which are sentences from *different webpages* that all express the same fact. The more times the fact is mentioned on different pages, the more confident we should be that it is correct, so we use the total number of instances returned by a query as one type of counting feature.

The other type of counting feature is based on the number of tuples. This feature is meant to capture the strength in a slightly different, way, based on the number of *different phrasings* of the relationship between arguments.

Up to this point, we have not said anything about the relation, only the arguments.

We generate two different types of features based on the relation.

The first, which we call **all_exts** counts all instances or tuples, without filtering based on the relation. Say we are generating count features for a particular pair of strings (a, b). For **all_exts** we generate queries of each of the following forms:

$$(a, \_, b), (b, \_, a), (a, \_, \_), (\_, \_, b), \tag{1}$$

where the underscore indicates a wildcard field that can match anything. The latter two are intended to allow the classifier to down-weight strings that are mentioned frequently on the web regardless of their relationship to one another.

The second type of feature based on the relation, which we call **causal_exts**, counts instances or tuples of the form $(a, rel_{\rightarrow}, b), (b, rel_{\leftarrow}, a)$, which we call rightward and leftward causal relations respectively. The strings $rel_{\rightarrow}$ and $rel_{\leftarrow}$ in these queries must match a predetermined list of strings that suggest a causal relationship from left to right or right to left, respectively.

We generated these lists of causal strings via a bootstrapping procedure similar to that of [12]. We started by generating a list of pairs of strings where we knew the first causally influenced the second. Then we searched Open IE for relations connecting each of these pairs, in both the rightward and leftward configuration. Finally we filtered the results of this bootstrapping by hand to remove obvious noise.

3.3 Confounder features

If two variables are correlated in a dataset, mathematical theory and human intuition tell us that one variable may causally influence the other or there may be a third variable (possibly in the dataset, but possibly not) that causally influences both of the correlated variables [24]. Common causes that are *not* in the dataset are a big potential problem for causal structure learning, and many causal learning methods explicitly assume there are none.

Open IE provides an interesting capability: given a pair of variables in a dataset, we can search for entities in the world that causally influence both of them. If there are a lot of such entities, presumably we should decrease our confidence that the pair of variables directly influence one another.

dataset	cols	rows	domain	col labels
Automobile	26	205	mechanical	curb weight, city mpg, engine size
Hepatitis	20	155	health	fatigue, ascites, alk phosphate
Concrete	9	1030	industrial	fine aggregate, superplasticizer, water
Adult[2]	15	2000	economic	age, education, agrossincome
NHEFS[3]	17	1746	health	bronch, pepticulcer, diabetes

Table 2: Dataset characteristics.

We perform both of the following queries:

$$(\_, rel_{\rightarrow}, a) \cup (a, rel_{\rightarrow}, \_) \quad (2)$$

$$(\_, rel_{\rightarrow}, b) \cup (b, rel_{\leftarrow}, \_) \quad (3)$$

and count the number of arguments that match a wild card on both the first line (causes of a) and the second line (causes of b).

3.4 Correlation feature

Although correlation is not causation, in practice, large correlations are trusted as more likely to be causal than small correlations [19].

We tried mutual information as a simple measure of statistical correlation, but it did not generalize well between datasets. We ended up using a statistic based on the G^2 test of independence, which is similar the χ^2 test. In essence, G^2 computes the expected counts in each bin in the joint probability table of two variables under the hypothesis of independence, and compares the observed counts to the expected counts. We converted the p-value from the G^2 test into the log-odds ratio ($\log(p)/\log(1-p)$) and used that as our feature.

4. EXPERIMENTS

4.1 Data

Our experiments focused on four datasets drawn from the UCI repository and another made available by Miguel Hernan at the Harvard School of Public Health [15]. In keeping with our goal to be as domain-general as possible, they cover areas ranging from medicine to automobiles (see Table 2). The criteria used to select them were that the variables had to have semantically meaningful labels, and there had to be at least one extraction in Open IE that related to the dataset.

The number of samples m in each dataset ranged from 155 to 2000, and the number of columns n ranged from 9 to 26. We constructed ground-truth causal graphs by hand, using a combination of common sense and web resources. Every ordered pair of nodes (A, B) in each dataset was labeled as "causal", "non-causal", or "uncertain", and evaluations were over the causal and non-causal pairs only.

4.2 Main results

Our main result is shown in Figure 1. We compare the best algorithms based on: semantic Open IE data, numerical data, and combination of both. We used a linear SVM with a 70-30 train-test split, and tuned the hyper parameters via 5-fold cross validation on the training set.

The combination of both Open IE data and numerical data performed the best ($AUC = 0.81$), followed closely by the Open IE data alone ($AUC = 0.78$). Surprisingly the PC algorithm, the best numerical structure learning algorithm among those we evaluated ($AUC = 0.58$), did worse than the simple numerical statistic G2 ($AUC = 0.64$).

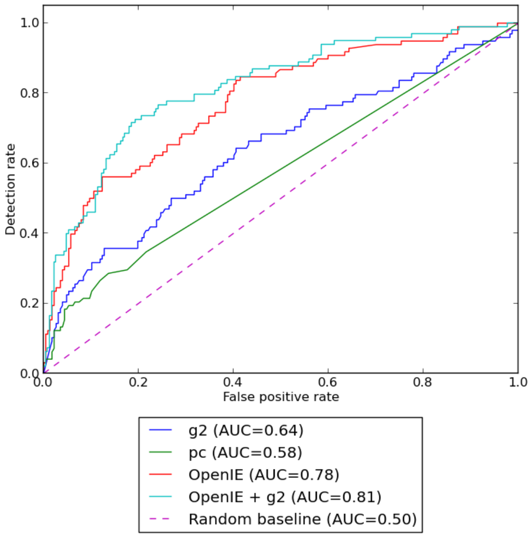

Figure 1: Structure learning with Open IE data, numerical data, and the combination of both.

Figure 2 shows the relative performance of the pure numerical structure learning algorithms we tried. We evaluated three different methods representing three major strategies in the literature: the **PC** algorithm [] (representing constraint-based algorithms), greedy hill-climbing (**HC**) with the BDE prior of Heckerman et al [] (representing score-based algorithms), and **IAMB** [] (representing Markov-blanket-based methods).

In Figure 3 we show the relative contribution of each type of text feature to the overall performance of **Open IE+G2**. About 50% of the overall increase above the baseline random performance is accounted for by the *all_exts* features ($AUC = 0.65$), which count Open IE extractions without taking causal relations into account. The *causal_exts* features, which count Open IE extractions involving a causal relation, improve performance to about 84% of total ($AUC = 0.76$). The *confounder* features bring performance up a modest 6%, with the numerical G2 statistic making up the final 10%.

[2] Original Adult dataset contains 32,562 samples. We selected a subset for efficiency reasons and because prior work indicates that in simulations, 2000 samples is sufficient to learn causal networks with 15 nodes [7].

[3] Original NHEFS dataset contains 61 variables—we selected a subset to make labeling the groundtruth graph more manageable.

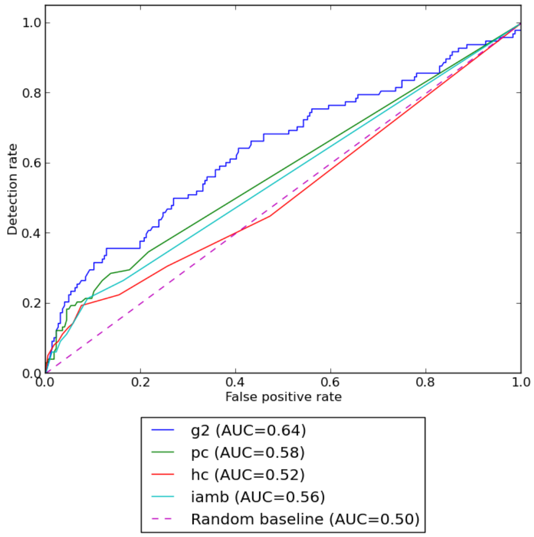

Figure 2: Comparison of numerical structure learning algorithms.

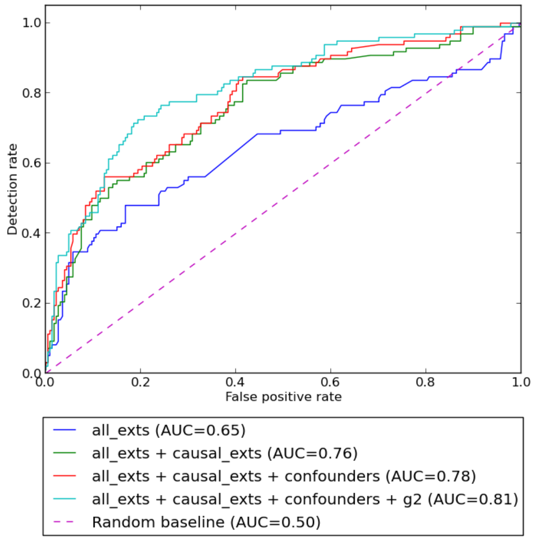

Figure 3: Ablation study showing contribution of each type of text feature to Open IE performance.

4.3 External validation

Since both the ground-truth labels and causal relation features were created by us, a natural question is whether we are over-fitting to these five datasets.

It was challenging to find external datasets that contained causal labels on real data. The datasets of the Causality Workbench [13], a major developer of causal learning competitions, intentionally do not include semantic labels, making the OpenIE approach inapplicable.

Instead, we use the MPI causal pairs benchmark [16], which is relatively small (86 pairs initially, 57 pairs after we removed semantic pairs that already appear in our 5 datasets) and unfortunately only contains data and labels for pairs of variables, not entire datasets. As a result, we did not evaluate the PC algorithm and other structure numerical learning methods on this dataset. However, we were able to use the MPI data to test whether our OpenIE-based method could perform well on datasets besides our own.

The results from this evaluation are shown in Figure 4, and indeed performance is similar ($AUC = 0.71$ compared with $AUC = 0.76$ on our data)[4], suggesting that over-fitting is not occurring to a large extent.

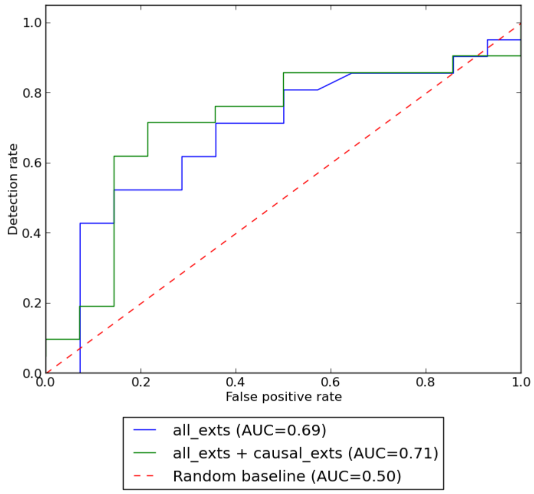

Figure 4: Results on the MPI causal pairs dataset.

4.4 Why numerical methods are failing

Another natural question is why are the numerical structure learning methods doing so badly compared to previously reported results?

Part of the reason seems to be that most evaluations we found in the literature are either: a) on synthetic data generated from Bayesian networks, or b) on gene networks, where the statistical assumptions of Bayesian networks are potentially more accurate [33]. As a sanity check, we ran **PC** on synthetic data. The results were striking: on the Lucas network [13], a network with 11 nodes and 2000 samples, the PC algorithm was able to achieve a detection rate of 100% at a false-positive rate of 1% — essentially perfect.

This result hints that existing structure learning algorithms are quite sensitive to distributional assumptions, in agreement with results reported in [17].

5. CURRENT AND FUTURE WORK

5.1 Global joint graph optimization

One unsatisfying aspect of causal structure learning with prior knowledge is that we may not be learning new causal knowledge, but rather simply putting what is already known into a model. This is fine if the application you have in mind is to learn parameters from the data, but if you actually want to learn something new about causal structure, we would hope for more.

[4] The MPI dataset consists of pairs (A, B) and the task is to determine direction as $A \rightarrow B$ or $A \leftarrow B$. The G^2 and confounder features do not make much sense in this case (they are both symmetric), so we did not include them in our comparison.

The theory of causal Bayesian networks suggests a natural sort of constraint propagation should be possible, in which prior knowledge orients some edges, and these constraints are propagated to other edges with then also become oriented, thus representing new knowledge about causal structure.

The simplest case of this is the situation depicted below.

$$
\begin{gathered}
A \rightarrow B \rightarrow C \\
A \leftarrow B \rightarrow C \\
A \leftarrow B \leftarrow C
\end{gathered}
$$

Each of these Bayesian networks encodes the same set of conditional independences: namely, $A \perp\!\!\!\perp C \mid B$. As a result, any structure learning method based on conditional independence relationships observed in the data ail not be to determine which is correct. However, if prior knowledge tells us that there is a path from A to B, then the ambiguity disappears: there is only one member of the equivalence class consistent with $A \perp\!\!\!\perp C \mid B$ and $A \rightarrow B$—namely top case, where $B \rightarrow C$. Thus we have learned the direction of (B, C) from prior knowledge about (A, B) combined with numerical data.

In light of this, it seems desirable to formulate the problem not as a per-edge detection task, but as a joint optimization procedure over all edges simultaneously.

As part of ongoing work, we have formulated this as a discrete optimization which we solve in a logic programming language called answer set programming [21]. This is similar to recent work by Hyttinen [17], who applied SAT solvers to estimate causal structure (they use hard constraints to encode prior knowledge but suggest MAX-SAT as a way to incorporate soft constraints.)

One nice feature of the logic programming approach is it is easy to encode the fact that the sentence "A causes B" may mean $A \rightarrow Z \rightarrow B$.

We incorporate terms in the objective function to reward paths that agree with OpenIE extractions, penalize double edges, penalize violations of conditional independence constraints (v-structures)

Early results are promising, and the solver is often able to find the optimum graph within a few minutes. However, we've found that it is hard to find a setting of the penalty parameters that generalizes well.

5.2 Data prediction

We have also been conducting experiments in which our objective is not to learn a graph structure which matches a causal ground-truth, but instead to maximize the log-likelihood of a non-causal predictive model.

The fact that the constraint-based algorithms such as PC and IAMB are not doing very well suggests that the naive assumptions about conditional independence may not hold in our datasets, and thus those constraints in the optimization may need to be altered in some way. There are other structure learning algorithms that relax the statistical assumptions somewhat (like the FCI algorithm [24]), and building on these are an interesting direction for future work.

6. CONCLUSION

We developed a system that combines the information in web text with numerical data to reconstruct more accurate causal graphs. Since our method is fully automated and is connected to an ever-growing web-scale knowledge base, it can potentially scale up to large datasets involving knowledge from many different domains. We envision possible applications in data mining and data visualization, in which the relationships deemed most likely to be causal are brought to the users attention.

AI systems with large amounts of knowledge have not historically played a major role in data analysis, since those systems that did incorporate knowledge have mostly been fairly application-specific. We speculate that approaches similar to the one presented here may one day bring about knowledge-rich automated data analysis systems that work on a wide variety of domains.

7. REFERENCES

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

[2] P. Antal, G. Fannes, D. Timmerman, Y. Moreau, and B. De Moor. Using literature and data to learn bayesian networks as clinical models of ovarian tumors. *Artificial Intelligence in medicine*, 30(3):257–281, 2004.

[3] M. Banko, O. Etzioni, and T. Center. The tradeoffs between open and traditional relation extraction. In *ACL*, volume 8, pages 28–36, 2008.

[4] K. Benson and A. J. Hartz. A comparison of observational studies and randomized, controlled trials. *New England Journal of Medicine*, 342(25):1878–1886, 2000.

[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[6] R. Castelo and A. Siebes. Priors on network structures. biasing the search for bayesian networks. *International Journal of Approximate Reasoning*, 24(1):39–57, 2000.

[7] D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321, 2012.

[8] A. P. Dawid. Beware of the dag! *Journal of Machine Learning Research-Proceedings Track*, 6:59–86, 2010.

[9] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[10] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[11] A. Gelman. Causality and statistical learning. `http://www.stat.columbia.edu/~gelman/presentations/causaltalk3_handout.pdf`, 2013.

[12] R. Girju and D. I. Moldovan. Text mining for causal relations. In *FLAIRS Conference*, pages 360–364, 2002.

[13] I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, J.-P. Pellet, P. Spirtes, and A. Statnikov. Design and analysis of the causation and prediction challenge. 2008.

[14] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[15] M. Hernan and J. Robbins. Causal inference book. `http://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/`, October 2013.

[16] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. 2009.

[17] A. Hyttinen, P. O. Hoyer, F. Eberhardt, and M. Järvisalo. Discovering cyclic causal models with latent variables: A general sat-based procedure. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

[18] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. *Journal of Bioinformatics and Computational Biology*, 2(01):77–98, 2004.

[19] D. Kaplan. *The Sage handbook of quantitative methodology for the social sciences*. Sage, 2004.

[20] S. L. Lauritzen and T. S. Richardson. Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):321–348, 2002.

[21] V. Lifschitz. Answer set programming and plan generation. *Artificial Intelligence*, 138(1):39–54, 2002.

[22] M. H. Maathuis, M. Kalisch, and P. Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009.

[23] R. T. OÕdonnell, A. E. Nicholson, B. Han, K. B. Korb, M. J. Alam, and L. R. Hope. Incorporating expert elicited structural information in the camml causal discovery program.

[24] J. Pearl. *Causality: models, reasoning and inference*, volume 29. MIT press Cambridge, 2000.

[25] K. Radinsky, S. Davidovich, and S. Markovitch. Learning causality for news events prediction. In *Proceedings of the 21st international conference on World Wide Web*, pages 909–918. ACM, 2012.

[26] M. Richardson and P. Domingos. Learning with knowledge from multiple experts. In *ICML*, volume 20, pages 624–631, 2003.

[27] O. Sanchez-Graillet and M. Poesio. Acquiring bayesian networks from text. In *LREC*, 2004.

[28] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*, volume 81. The MIT Press, 2000.

[29] V. I. Spitkovsky and A. X. Chang. A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175, 2012.

[30] D. F. Stroup, J. A. Berlin, S. C. Morton, I. Olkin, G. D. Williamson, D. Rennie, D. Moher, B. J. Becker, T. A. Sipe, S. B. Thacker, et al. Meta-analysis of observational studies in epidemiology. *JAMA: the journal of the American Medical Association*, 283(15):2008–2012, 2000.

[31] S. Suissa and E. Garbe. Primer: administrative health databases in observational studies of drug effectsÑadvantages and disadvantages. *Nature Clinical Practice Rheumatology*, 3(12):725–732, 2007.

[32] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov. Algorithms for large scale markov blanket discovery. In *FLAIRS Conference*, volume 2003, pages 376–381, 2003.

[33] I. Tsamardinos and S. Triantafillou. Causal discovery from mass cytometry data. `http://www.mensxmachina.org/files/presentations/cmu_tsamardinos.pdf`, 2013.

APPENDIX

A. CAUSAL BAYESIAN NETWORKS

The goal of causal modeling is to predict what will happen to a system when a certain action is performed. In all frameworks for predicting the outcomes of actions (henceforth *causal models*), an important object of study is the probability distribution of the outcome X conditioned on the action taken Z, which we write as $p(X \mid Z)$.

One the most popular formalisms for causal modeling, popularized by Pearl [24], is the causal Bayesian network, also known as a causal directed acyclic graph (causal DAG). Causal Bayesian networks take advantage of a particular factorization of $p(X \mid Z)$ in certain systems. Causal Bayesian networks are closely related to the normal Bayesian networks which are widely known in computer science, and in fact, causal Bayesian networks can be expressed as traditional Bayesian networks with extra nodes representing "interventions" [8] (see Figure 5).

In a causal Bayesian network, the outcome variable X is defined as a product of n outcome variables $(x_1, \ldots, x_n)$. Each individual variable could represent something like whether a particular person smokes cigarettes in a certain timeframe or whether they get lung cancer in another timeframe. In the simplest interpretation of causal models, these variables are interpreted as "manipulatable" variables, which will be defined below.

The first assumption usually made is that **an action corresponds to setting a subset of the outcome variables to be certain values**. More formally, let $S(x_i)$ denote the possible values of an outcome variable x_i. The action variable Z is then a product of n individual variable manipulations $(z_1, \ldots, z_n)$. Each variable manipulation z_i is either the null action $\varnothing$ or a "set-variable" action which fixes x_i to one of its possible values. Formally, we have

$$z_i \in S(z_1) = (S(x_i) \cup \{\varnothing\}), \tag{4}$$

$$Z \in S(z_1) \times ... \times S(z_n). \tag{5}$$

Already this assumption hints at how causal modeling can learn the outcomes of actions that are never performed during learning: the non-null actions are in 1-to-1 correspondence with different the values a variable can assume spontaneously. Intuitively, this correspondence suggests we may be able to learn something about what effect a particular action would have just by waiting until its corresponding outcome variable assumes the appropriate value, instead of actively setting it.

The second assumption is that the distribution $p(X \mid Z, \Theta)$ factorizes according to the directed acyclic graph (DAG) in Figure 5), in the sense of a classical Bayesian network (cf. Pearl). Each action variable z_i has a single directed edge connecting it to its corresponding outcome variable x_i.

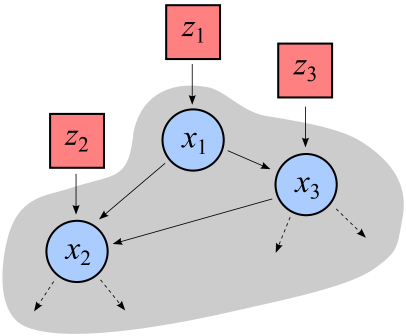

Figure 5: The augmented graph representation of a causal Bayesian network. Action variables are red squares, observable variables are blue circles.

The final assumptions involve the conditional probability distribution of each node given its parents. Intuitively, there is a "default" distribution $p(X \mid Z = \varnothing)$ (the outcome distribution given the null action) from which the model derives other distributions conditional on non-null actions. More formally, we assume:

$$p(x_i = j \mid X - \{x_i\}, Z = z) = \begin{cases} 1 & \text{if } z_i = j \\ p(x_i = j \mid \mathrm{pa}_X(x_i),\ Z = \varnothing) & \text{if } z_i = \varnothing \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

where $X - \{x_i\}$ represents all outcome variables besides x_i and $\mathrm{pa}_X(x_i)$ represents $\mathrm{pa}(x_i) \cap X$, the intersection of the parent set of x_i and X.

There are a few important properties to notice about this model class.

First, if some node x_j is not a descendent of x_i, then $x_j \perp\!\!\!\perp z_i$. Informally, this means that the action z_i only influences the descendants of x_i. This can be proven by using the properties of d-separation. Specifically, it can be shown that z_i is d-separated from all non-descendants of x_i conditioned on the empty set.[5]

Second, the DAG relating the outcome variables can be regarded as a Bayesian network in its own right: the distribution $p(X \mid Z = \varnothing, \Theta)$ is the pure "passively observed" distribution. In fact Pearl, in his later work, eliminates the action variables z_i from his notation entirely, and the subgraph relating the outcome variables X is the DAG under study. We will call this subgraph the *outcome subgraph.*

Third, the direction of the edges in the outcome subgraph always matters. In fact two models of this form express the same set of probability distributions (i.e. they are Markov equivalent) if and only if their outcome subgraphs are identical. Consider the case of a network with just two outcome variables x_1 and x_2 connected by an edge $x_1 \to x_2$ (see Figure 6). In this Bayesian network, the direction of the edge contains no information: in either case, the set of conditional independences implied by the graph is the same (namely, there are no implied conditional independences). Contrast this with the augmented graph containing action variables z_1 and z_2. In this case, the direction of the edge between x_1 and x_2 does matter. If the edge is $x_1 \to x_2$, then z_1 and x_2 are d-separated (and thus independent) conditioned on x_1. However, this is not the case if $x_1 \leftarrow x_2$.

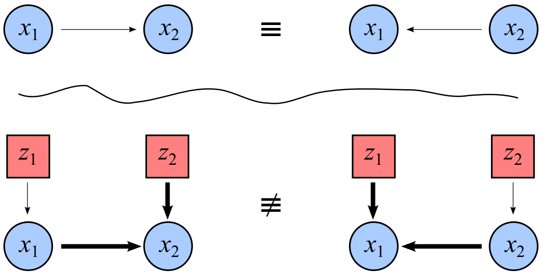

Figure 6: Two models that are Markov equivalent without action nodes, but are no longer Markov equivalent when action nodes are added. V-structures highlighted for emphasis.

These properties are interesting because they more closely match human reasoning and intuition. If a variable is set to a value, it makes intuitive sense that any variables which are descendants of that variable should be affected. In the classical Bayesian network setting, if we take conditioning on a variable as a proxy for "setting it", we get the slightly counter-intuitive result that distribution of the variable's parents changes. Cognitive scientists such as Steven Sloman have shown that humans make systematic errors when handling probabilistic models and statistical conditioning — errors that reveal a bias towards the model properties described above.

What we are doing here is expressing causal semantics using Bayesian networks. Pearl, one of the progenitors of the Bayesian network formalism, claims that his original Bayes net formulation was in fact inspired by causal intuition, but that in retrospect *the original Bayes net formulation failed to capture those intuitions correctly.* In light of this, it is slightly odd to express the more intuitive mechanics of causal DAG models using the non-intuitive mechanics of Bayes nets. I would claim, however, that most computer scientists are more familiar with the formal properties of Bayesian networks than with the various formalizations of causal intuition. Thus for this exposition, which is intended to somewhat demystify causal modeling, it makes sense to express our causal models in the language of Bayes nets. Pearl's later works introduce a collapsed form of these Bayesian networks in which each action node is merged with its corresponding outcome variable, and each action event $z_i = k$ is denoted by $\mathrm{do}(x_i = k)$.

One possible criticism of this model class is that in real life, one cannot force a variable have a particular value with

[5] This is because z_i has no parents, and exactly one edge, $z_i \to x_i$. Clearly any active path between z_i and another node x_j must include that edge. Since we are conditioning on the empty set, this forces any active path between z_i and another node x_j to in fact be a directed path from z_i to x_j, through x_i. The non-descendants of x_i are exactly the nodes for which no such paths exist, making them exactly the nodes which are d-separated from z_i.

probability 1 — there is always some noise. In fact, this model class can express such noisy relationships by introducing hidden outcome variables. The observable outcome variables can then be made to depend noisily on the hidden ones. In some sense, the action variables only represent an idealized modeling construct, not the real actions that people can perform.[6]

Another problem is that the acyclicity constraint is awkward for modeling some systems. Dawid [8] gives two examples of this: the ideal gas law relating pressure, temperature, and volume; and the relationship between supply, demand, and price. In these cases, the acting on variable x_1 influences x_2, and acting on x_2 also influences x_1. In a DAG-based formalism this directionality cannot be expressed. However, there are alternative models that can represent causal cycles, for example chain graph models ([20]).

[6] In practice, the distinction between an "idealized action" and a noisy action may be moot if the level of noise is sufficiently low, in which case there is no need to introduce extra variables.