Causality and Explanations in Databases*

Alexandra Meliou
University of Massachusetts
Amherst, MA, USA
ameli@cs.umass.edu

Sudeepa Roy
University of Washington
Seattle, WA, USA
sudeepa@cs.washington.edu

Dan Suciu
University of Washington
Seattle, WA, USA
suciu@cs.washington.edu

ABSTRACT

With the surge in the availability of information, there is a great demand for tools that assist users in understanding their data. While today's exploration tools rely mostly on data visualization, users often want to go deeper and understand the underlying causes of a particular observation. This tutorial surveys research on causality and explanation for data-oriented applications. We will review and summarize the research thus far into causality and explanation in the database and AI communities, giving researchers a snapshot of the current state of the art on this topic, and propose a unified framework as well as directions for future research. We will cover both the theory of causality/explanation and some applications; we also discuss the connections with other topics in database research like provenance, deletion propagation, why-not queries, and OLAP techniques.

1. MOTIVATION

With the surge in the availability of information, there is great need for tools that help users *understand* data. There are several examples of systems that offer some kind of assistance for users to understand and explore datasets. Humans typically observe the data at a high level of abstraction, by aggregating or by visualizing it in a graph, but often they want to go deeper and understand the ultimate causes of their observations. Over the last few years there have been several efforts in the Database and AI communities to develop general techniques to model causes, or explanations for observations on the data, some of them enabled by Judea Pearl's seminal book on *Causality*[1]. Causality has been formalized both for AI applications and for database queries, and formal definitions of *explanations* have also been proposed both in the AI and the Database literature. Given the importance of developing general purpose tools to assist users in understanding data, it is likely that research in this space will continue, perhaps even intensify.

Depth and Coverage. This 1.5-hour tutorial aims at establishing a research checkpoint: its goal is to review, summarize, and systematize the research so far into causality and explanation in databases, giving researchers a snapshot of the current state of the art on this topic, and at the same time propose a unified framework for future research. We will cover a wide range of work on causality and explanation from the database and AI communities, and we will discuss the connections with other topics in database research.

Intended audience. The tutorial is aimed both at active researchers in databases, and at graduate students and young researchers seeking a new research topic. Practitioners from industry might find the tutorial useful as a preview of plausible future trends in data analysis tools.

Assumed Background. Basic knowledge in databases will be sufficient to follow the tutorial. Some background in Datalog, provenance, and/or OLAP would be useful, but is not necessary.

2. COVERED TOPICS

Our tutorial is divided in three thematic sections. First, we discuss the notion of causality, its foundations in AI and philosophy, and its applications in the database field. Second, we discuss how the intuition of causality can be used to explain query results. Third, we relate these notions to several other topics of database research, including provenance, missing results, and view updates.

2.1 Causality

Understanding causality in a broad sense is of vital importance in many practical settings, e.g., in determining legal responsibility in multi-car accidents, in diagnosing malfunction of complex systems, or in scientific inquiry. The notion of causality and causation is a topic in philosophy, studied and argued over by philosophers over the centuries. On a high level, causality characterizes the relationship between an event and an outcome: the event is a cause if the outcome is a consequence of the event. The notion of *counterfactual* causes, which can be traced back to Hume (1748) and is analyzed later by Lewis (1973), explains causality in an intuitive way: if the first event (cause) had not occurred, then the second event (effect) would not have occurred.

Several philosophers explored an alternative approach to counterfactuals that employs structural equations. Judea Pearl's landmark book on causality defined the state-of-the-art formulation of this framework. Pearl's and Halpern

*Partially supported by NSF Awards IIS-0911036 and CCF-1349784.

[1] All references are omitted and will appear in the tutorial due to space limitations.

Articles from this volume were invited to present their results at the 40th International Conference on Very Large Data Bases, September 1st - 5th 2014, Hangzhou, China.
Proceedings of the VLDB Endowment, Vol. 7, No. 13
Copyright 2014 VLDB Endowment 2150-8097/14/08.

and Pearl's work distilled the generally accepted aspects of causality into a rigorous definition.

Closer to home, the study of causality in databases was motivated by the need to find *reasons* for surprising observations, or simply to trace observations on the outputs back to the inputs. In a database context, they would like to find the causes of answers or non-answers to their queries, *e.g.*, "What caused my personalized newscast to have more than 50 items today?" or, "What caused my favorite undergrad student to not appear on the Dean's list this year?"

Causality in databases aims to answer the following question: *given a query over a database instance and a particular output of the query, which tuple(s) in the instance caused that output to the query?* Meliou *et al.*(2011) answer this question by extending the notions of causality and responsibility to database queries. The intuition is to quantify the contribution that each input tuple has to an output, and identify the input tuples with the highest contribution.

Unfortunately, quantifying these causal contributions is generally NP-hard. With respect to data complexity, there are two approaches to the problem: In the **instance-based** approach, the focus is on the complexity of computing the causal contributions of tuples to a query result *for a given data instance*. The results by Eiter and Lukasiewicz (2002) show that deciding causality for general Boolean expressions is NP-complete, but they identify several tractable cases. The **query-based** approach focuses on the complexity of computing the causal contributions of tuples to a given query over *any* data instance. For the class of conjunctive queries without self-joins, Meliou *et al.* showed a complete dichotomy between the NP-complete and the polynomial-time cases. The problem remains open for other classes of queries.

Aside from analyzing the complexity of computing the causal contribution of a tuple to a query, there are two practical considerations to the problem: (a) the impact of an input to multiple queries, and (b) a practical approach to computing causality even in the NP-hard cases. View-conditioned causality (Meliou *et al.* 2011) extends the notion of causality to account for the effect of a tuple on multiple outputs (views) and describes a reduction to SAT, which allows for the use of SAT-solvers to compute causal contributions.

Halpern and Pearl's work on actual causes and its extension in databases rely on a framework of structural equations that describes the causal structure of a system. In the database domain, this causal structure may be defined using the *lineage* or *provenance* of an answer (*e.g.*, Cui *et al.* 2000, Green *et al.* 2007) or integrity constraints (Roy and Suciu 2014). A different direction has focused on deriving causal relationships directly from the data (Silverstein *et al.* 2000).

2.2 Explanations in Databases

According to its strict interpretation, causality can only be established by a controlled experiment, where one changes one single variable while keeping all others unchanged, and observes a change of the output. This is not possible using data alone. *Explanation* lowers the bar and aims at finding inputs that are best correlated with the outputs. While less well formalized than causality, explanation is more appealing in practice when we don't have full control over the inputs.

Explanations for general database query answers. Traditional data analysis techniques using OLAP focus on viewing aggregate information over multi-dimensional data to find interesting information. On the other hand, the new trend in *explanations in databases* aims to provide answers to more complex questions on query outputs, typically for aggregate queries, where the results are visualized with the help of simple plots. Here the explanations are formulated as predicates on the input attributes. These predicates, similar to the notion of intervention in the causality literature, cause the output aggregate values to change when applied to the input, and are ranked according to a score that measures how much they affect the outputs. Following this approach, Wu and Madden (2013) proposed the Scorpion system: given an aggregate query over a single relation and a set of outlier points in the output, it returns top predicates which make the outliers disappear. Roy and Suciu (2014) proposed a formal framework for finding explanations to complex SQL queries over database schemas involving multiple relations and foreign key constraints. Adopting the notion of *causal paths* in the causality literature, each such explanation is now associated with an intervention – a set of tuples to be removed from the database – which includes all tuples defined by the explanation, plus all tuples implied by causal relationships through foreign key constraints and their extensions.

Explanations for specific database applications. Several research projects in databases have aimed at explaining query answers focusing on interesting applications. For instance, Khoussainova *et al.* (2012) proposed the system PerfXplain where the users specify the expected and observed performance of pairs of MapReduce jobs as well as a *despite* clause stating how similar the jobs are, and are shown top explanations based on relevance, precision, and generality. Das et. al. (2011) studied *Meaningful Ratings Interpretation* to help a user easily interpret ratings of items on Yelp or IMDB, (*e.g.*, "male reviewers under 30 from NYC love this movie"). Fabbri and LeFevre (2011) and Bender *et al.* (2014) studied explanation-based auditing of access log. Re and Suciu (2008) and Kanagal *et al.* (2011) studied computing top-k explanations for conjunctive queries on probabilistic databases (for questions like "why a tuple is in the output" or "why a tuple has higher probability than another one").

2.3 Related topics in Databases

Causality and explanations are related to several other topics that have been studied in the literature. This tutorial will summarize them and explain their connection to causality and explanations. **Data provenance** studies formalisms that capture why a particular data item is in the output, whereas **deletion propagation** aims at finding which tuples in the database need to be deleted in order to remove a certain tuple from the view, with minimum *side-effects* on the input and output. The problem of explaining **missing query results** aims to answer "why a certain tuple does not appear as an answer", in terms of base tuples or query predicates. **Data mining** aims at finding common patterns, which are different from causality, but tools from data mining have also been deployed to find causal dependencies; *e.g.*, Silverstein *et al.* (2000) studied the problem of efficiently determining causal relationship (*i.e.*, not simple association or correlation) for mining market basket data. Sarawagi and Sathe (2000) proposed new operators for efficient data analysis in **OLAP** data cubes, *e.g.*, *RELAX*, *DIFF*, and *SURPRISE*.

2.4 Conclusions

We will conclude the tutorial with a discussion of open problems and challenges for database research in the area of causality and explanation.