Contributions to Personalizable Knowledge Integration

Maria Vanina Martinez
University of Maryland College Park
College Park, MD 20783, USA
mvm@cs.umd.edu

Abstract

Inconsistency and partial information is the norm in knowledge bases used in many real world applications that support, among other things, human decision making processes. In this work we argue that the management of this kind of data needs to be context-sensitive, creating a synergy with the user to build useful, flexible data management systems.

1 Introduction

Large repositories of data are used daily as knowledge bases (KBs) feeding computer systems that support decision making processes, such as in medical or financial applications. Unfortunately, the larger a KB is, the harder it is to ensure its consistency and completeness, and many times this might not be an option or even a desirable action. The problem of handling KBs of this kind has been studied in the AI and Databases communities, but most approaches focus on computing answers locally to the KB, assuming there is some single, epistemically correct solution. An important aspect to consider is that for some applications, as part of the decision making process, users consider far more knowledge than that which is contained in the knowledge base, and that sometimes inconsistent data may help in directing reasoning, *e.g.*, inconsistency in taxpayer records can serve as evidence of a possible fraud. Thus, the handling of this type of data needs to be context-sensitive, creating a synergy with the user in order to build useful, flexible data management systems.

The goal of my work is to attack particular problems of knowledge integration and provide personalizable approaches to handle them. Specifically, I focus on (1) inconsistency management in relational databases, general KBs, and a special kind of KBs designed for news reports; and (2) management of incomplete information in the form of null values.

Using the proposed frameworks, users can specify when and how they want to manage/solve the issues that the integration of several heterogeneous knowledge bases yield, in the way that best suits their needs.

2 Progress Made to Date

1a) Policy-based Inconsistency Management in Relational Databases: The process of knowledge integration involves combining knowledge bases residing in different sources and providing users with a *unified view* of these data. Researchers both in AI and databases, as well as in information retrieval, have been working on the problems that arise with the integration of heterogeneous knowledge bases for decades [Baral *et al.*, 1991; Benferhat *et al.*, 1997; Besnard and Schaub, 1998; Arenas *et al.*, 1999; Bohannon *et al.*, 2005]. However, almost all past approaches proceeded under the assumption that there was some single epistemically correct way of resolving inconsistencies or reasoning in the presence of inconsistency. To see why it is important to take into account the user's context, let us consider a database containing employees' salary data. Let us assume that salaries are uniquely determined by names but there is more than one record for a certain employee **John**: two records stating that he earns $70K$, and one more stating that his salary is $80K$.

Clearly, there is an inconsistency w.r.t to John's salary and, in this case, a user may want to resolve the inconsistency in many different ways. (C1) If he were considering John for a loan, he might want to choose the lowest possible salary of John to base his loan on. (C2) If he were assessing the amount of taxes John has to pay, he may choose the highest possible salary John may have. (C3) If he were just trying to estimate John's salary, he may choose some number between 70K and 80K (*e.g.*, the average of the three reports of John's salary) as the number. (C4) if he had different degrees of confidence in the sources that provided these salaries, he might choose a weighted mean of these salaries. (C5) He might choose not to resolve the inconsistency at all, but to just let it persist until he can clear it up. (C6) He might simply consider *all* the data about John unreliable and might want to ignore it until it can be cleared up – this is the philosophy of throwing away all contaminated data.[Baral *et al.*, 1991; Arenas *et al.*, 1999; Bohannon *et al.*, 2005] can handle cases C1 and C2, but not the other cases.

In [Martinez *et al.*, 2008], we proposed to provide users with tools to manage their data in a personalized way in order to reason about it according to their needs. These tools are called *inconsistency management policies* or IMPs. IMPs are defined with respect to functional dependencies and generalize other efforts in the database community by allowing policies to either remove inconsistency completely or to allow part or all of the inconsistency to persist, depending on the users' application needs. In the example above, each of

the cases C1 through C6 reflects a *policy* that the user is using to resolve inconsistencies.

1b) Unified Inconsistency Management Framework in KBs: We also propose a unified framework for reasoning about inconsistency that extends [Subrahmanian and Amgoud, 2007]. This framework applies to any monotonic logic, including ones for which inconsistency management has not been well studied (*e.g.*, temporal, spatial, and probabilistic logics), and the main goal is to allow end-users to bring their domain knowledge to bear by taking into account their preferences. In the example above, neither the bank manager nor the tax officer are making any attempt to find out the truth (thus far) about John's salary; however, both of them are making different decisions based on the same facts. The basic idea in this framework is to construct options and then, using a preference relation defined by the user, compute compute the set of *preferred options*, which are intended to support the conclusions to be drawn from the inconsistent knowledge base. Intuitively, an option is a set of formulas that is both consistent and closed with respect to consequence in a given monotonic logic. Note that preferred options are not necessarily consistent subsets of the KB; instead, they are consistent subsets of the deductive closure of the KB.

1c) Inconsistency in News Reports: A real world domain that is heavily affected by integration techniques is that of news reports, especially since millions of reports can be extracted daily automatically from different web sources. Oftentimes, even the same news source may provide widely varying data over a period of time about the same event. Past work on inconsistency management and paraconsistent logics assume that we have "clean" definitions of inconsistency. However, when reasoning about news, there is an extra layer of uncertainty, that comes from the following two phenomena: (i) do two reports correspond to the same event or different ones?; and (ii) what does it mean for two event descriptions to be mutually inconsistent, given that these events are often described using linguistic terms that do not always have a uniquely accepted formal semantics? In [Martinez *et al.*, 2010], we proposed a probabilistic logic programming language called PLINI (Probabilistic Logic for Inconsistent News Information) within which users can write rules specifying what they mean by inconsistency in situation (ii) above. Extensive work has been done in duplicate record identification and elimination. The main difference between our approach and previous work is the fact that the user is able to specify the notion of inconsistency that is of interest to him; furthermore, news reports are in general unstructured data containing complex *linguistic modifiers* which different users may interpret differently.

2) Management of Incomplete Information. The problem of representing incomplete information in relational databases and understanding its meaning has been extensively studied. Early work on this problem appears in [Grant, 1980; Lipski, 1981]. Many data modeling and analysis techniques deal with missing values by removing from consideration whole records if one of the attribute values is missing, or using ad hoc methods of estimation for such values. Even though a wide variety of methods to deal with incomplete information have been proposed, which are in general highly tuned for particular applications, no tools exist to allow end-users to easily specify different ways of managing the data according to their needs and based on their expertise.

Towards this end, we proposed the general concept of a *partial information policy* (PIP), for dealing with null values in relational databases. We focus on missing information that corresponds to either an *unknown value*, *inapplicable null*, or *no-information null*. The rationale behind the PIP framework is that any method to reason about incomplete information must adequately consider the nature of the missing data rather than adopt a "one size fits all" policy. Users specify how to replace (some) null values in a relation, and these policies can be combined with relational algebra operators. We propose index structures for efficiently applying PIPs and experimentally assess their effectiveness on a real world data set. Furthermore, we proposed and analyzed augmenting relational algebra operators with PIPs and study how they interact with one another, specifically under which conditions the property of commutativity holds.

3 Proposed Plan of Research

Research towards obtaining a PhD is expected to be complete by August 2011. The work described in the Progress section will be integrated into a thesis describing personalized methods for knowledge integration and managing inconsistency and incompleteness for several types of knowledge bases.

References

[Arenas *et al.*, 1999] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *PODS*, pages 68–79, 1999.

[Baral *et al.*, 1991] C. Baral, S. Kraus, and J. Minker. Combining multiple knowledge bases. *TKDE*, 3(2), 1991.

[Benferhat *et al.*, 1997] S. Benferhat, D. Dubois, and H. Prade. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study part 1: The flat case. *Studia Logica*, 58(1):17–45, 1997.

[Besnard and Schaub, 1998] P. Besnard and T. Schaub. Signed systems for paraconsistent reasoning. *J. Autom. Reas.*, 20(1):191–213, 1998.

[Bohannon *et al.*, 2005] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD*, pages 143–154, 2005.

[Grant, 1980] John Grant. Incomplete information in a relational database. *Fund. Informaticae III*, 3:363–378, 1980.

[Lipski, 1981] Witold Lipski. On databases with incomplete information. *J. ACM*, 28(1):41–70, 1981.

[Martinez *et al.*, 2008] M. V. Martinez, F. Parisi, A. Pugliese, G. I. Simari, and V.S. Subrahmanian. Inconsistency management policies. In *KR 2008*, pages 367–376, 2008.

[Martinez *et al.*, 2010] M. V. Martinez, M. Albanesse, M. Broecheler, J. Grant, and V.S. Subrahmanian. Plini: a probabilistic logic program framework for inconsistent news information. *Proc. of Constructive Mathematics in Computer Science*, 2010.

[Subrahmanian and Amgoud, 2007] V. S. Subrahmanian and L. Amgoud. A general framework for reasoning about inconsistency. In *IJCAI*, pages 599–504, 2007.