AIBD'13: The First International Workshop on Artificial Intelligence for Big Data

(in conjunction with ICJAI 2013, Beijing, China)

Organizers: Barry O'Sullivan, Vijay Saraswat and Roland H.C. Yap

5 August, 2013

Preface

Big Data is an emerging research area of great interest across many communities in computer science, for example, systems, programming languages, parallel and distributed computing, artificial intelligence, social computing. Big Data is characterized by large amounts of data being generated continuously by interconnected systems of people and things – click data, audio/speech data, natural language text (in multiple languages), images/video data. The challenge is to analyze the information content in these vast, continuous data streams, use them for descriptive and predictive analytics in various domains, build more robust and intelligent learning systems. Big Data offers incredible opportunities in a very diverse set of fields ranging from Consumer Marketing to politics (campaigning).

The objective of this workshop is to bring together a multi-disciplinary group of researchers and technologists from academia and industry to explore the opportunities of Big Data focusing both on applications of artificial intelligence to Big Data problems and on the use of Big Data in AI (e.g. in modeling, learning, problem-solving, multi-modal analytics).

Papers in the workshop cover topics such as:

- constraints
- data analytics
- data cleaning
- machine learning
- ontologies
- relational learning
- semantic modelling

Organization

Workshop Papers

Learning Bayes Nets with Link Uncertainty for Relational Data Sets

Oliver Schulte and Zhensong Qian

School of Computing Science
Simon Fraser University
Vancouver-Burnaby, Canada

Abstract

Many if not most big data sets are maintained in relational databases. We describe Bayes net learning methods that can discover knowledge about correlations among both link types and node attributes in big relational data. A key scalability challenge for relational learning is to compute event counts in a relational database (sufficient statistics), especially when these involve negated relationships. We provide empirical evidence that the fast Möbius transform provides a scalable solution for this problem.

1 Introduction: Link Correlations

Scalable link analysis for relational data with multiple link types is a challenging problem in network science. We describe a method for learning a Bayes net that captures simultaneously correlations between link types, link features, and attributes of nodes.

Building a Bayes net model is useful for big data analysis because such models provide a compact summary of the statistical relationships in the data. The model supports both descriptive and predictive analytics. Correlations are presented to the user in a graphical way, and queries about probabilistic relationships can be answered quickly using Bayes net inference rather than via database queries run against a large dataset.

Previous work on learning Bayes nets for relational data was restricted to correlations among attributes given the existence of links [4]. The larger class of correlations examined in our new algorithms includes two additional kinds:

1. Dependencies between two different types of links.
2. Dependencies among node attributes given the *absence* of a link between the node.

Contributions include the following:

1. To our knowledge this is the first implementation of Bayes net learning for modelling correlations among different types of links.
2. Using the Möbius transform to make the computation of sufficient statistics for negated relationships tractable [4].

2 Background and Notation

Poole introduced the Parametrized Bayes net (PBN) formalism that combines Bayes nets with logical-relational syntax [2]. A **population** is a set of individuals. A **population variable** is capitalized. A **functor** represents a function or a Boolean predicate. A predicate with more than one argument is called a **relationship**; other functors are called **attributes**. A **Parametrized random variable** (PRV) is of the form $f(X_1, \ldots, X_a)$, where the populations associated with the variables are of the appropriate type for the functor. A **Parametrized Bayes Net (PBN)** structure is a directed acyclic graph whose nodes are PRVs.

We assume that data are represented in a standard **relational schema** containing a set of tables, each with key fields, descriptive attributes, and possibly foreign key pointers. The powerset of relationship tables can be ordered as a lattice (e.g., $\{Reg(S, C)\} \sqsubseteq \{Reg(S, C), Teaches(C, P)\}$. For each relationship set, there is a **data table** whose columns consist of: (1) the attributes of all entities/relationships involved in the set, and (2) *a Boolean relationship node* for each relationship, that records whether the relationship holds between two entities. For an illustration of these concepts see Figure 1.

Methods Compared We compared the following methods.

Flat Applies a single-table Bayes net learner to the maximal data table comprising all relationship sets in the database. The results of [3] provide a theoretical justification for this procedure.

LAJ The previous hierarchical learn-and-join method [4] without relationship nodes in the data table and hence without link correlations. Conducts bottom-up search through the lattice of relationship sets. Dependencies (Bayes net edges) discovered for smaller sets are propagated to larger sets.

LAJ+ The new LAJ method with relationship data that has the potential to find link correlations.

(a)

Student		
s-id	Intelligence	Ranking
Jack	3	1
Kim	2	1
Paul	1	2

(b)

Registration			
s-id	C.id	Grade	Satisfaction
Jack	101	A	1
Jack	102	B	2
Kim	102	A	1
Kim	103	A	1
Paul	101	B	1
Paul	102	C	2

(c)

Course					
c-id	Prof	Rating	Difficulty	T-a-prof	P-prof
101	Oliver	3	1	1	3
102	David	2	2	1	2
103	Oliver	3	2	1	3

(d)

attribute-relation table								
Intelligence	Rank	Rating	Diff	P-prof	t-a-prof	Reg(S,C)	R.Grade	R.Sat
3	1	3	1	3	1	T	A	1
3	1	2	2	2	1	T	B	2
3	1	3	2	3	1	F	⊥	⊥
2	1	3	1	3	1	F	⊥	⊥
2	1	2	2	2	1	T	A	1
2	1	3	2	3	1	T	A	1
1	2	3	1	3	1	T	B	1
1	2	2	2	2	1	T	C	2
1	2	3	2	3	1	F	⊥	⊥

(e)

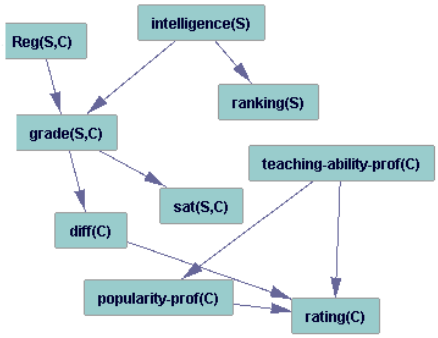

Figure 1: Database Table Instances: (a) *Student*, (b) *Registered* (c) *Course*. To simplify, we added the information about professors to the courses that they teach. (d) The data table for *Registered*(S, C), which lists for each pair of entities their descriptive attributes, whether they are linked by *Registered*, and the attributes of a link if it exists. $\perp$ means "not applicable." (e) A Parametrized Bayes Net for the university schema.

Dataset	**Flat**	**LAJ+**	**LAJ**
University	1.916/x	1.183/x	0.291/x
Movielens	38.767/x	18.204/x	1.769/x
Mutagenesis	3.231/x	3.448/x	0.982/x
Small-Hepatitis	9429.884/x	8.949/x	10.617/x

Table 1: Model Structure Learning Time in seconds. The first number refers to a simple SQL query implementation, the second to an implementation with database indexes and the fast Möbius transform.

3 Evaluation

For the details of the system setup, the datasets, and the fast Möbius transform please see [4]. We report learning time, log-likelihood, Bayes Information Criterion (BIC), and the Akaike Information Criterion (AIC) [1].

3.1 Results

Learning Times Table 1 provides the model search time for each of the link analysis methods. The combination of indexing and the Möbius transform provides substantial speedups. On the medium-size and more complex datasets (Hepatitis, MovieLens), hierarchical search is much faster due to its use of constraints.

Statistical Scores On the medium-sized dataset MovieLens, which has a simple structure, all three methods score similarly. LAJ and LAJ+ return the same model. The most complex dataset, Hepatitis, is a challenge for flat search, which overfits severely. Because of the complex structure of the Hepatitis schema, the hierarchical constraints are effective in combating overfitting. The situation is reversed on the Mutagenesis dataset where flat search does much better than hierarchical search. The reason for that is that, unusually, links in Mutagenesis are dense. As a result, we find strong correlations between attributes conditional on *the absence of relationships*. Our current version of the LAJ+ algorithm cannot detect such correlations; we leave an appropriate extension for future work.

University	BIC	AIC	log-likelihood	# Parameter
Flat	-17638.27	-12496.72	-10702.72	1767
LAJ+	-13495.34	-11540.75	-10858.75	655
LAJ	-13043.17	-11469.75	-10920.75	522

MovieLens	BIC	AIC	log-likelihood	# Parameter
Flat	-4912286.87	-4911176.01	-4910995.01	169
LAJ+	-4911339.74	-4910320.94	-4910154.94	154
LAJ	-4911339.74	-4910320.94	-4910154.94	154

Mutagenesis	BIC	AIC	log-likelihood	# Parameter
Flat	-21844.67	-17481.03	-16155.03	1289
LAJ+	-47185.43	-28480.33	-22796.33	5647
LAJ	-30534.26	-25890.89	-24479.89	1374

Hepatitis	BIC	AIC	log-likelihood	# Parameter
Flat	-7334391.72	-1667015.81	-301600.81	1365357
LAJ+	-457594.18	-447740.51	-445366.51	2316
LAJ	-461802.76	-452306.05	-450018.05	2230

Table 2: Performance of different Model Search Algorithms by dataset.

4 Conclusion

We described different methods for extending relational Bayes net learning to correlations involving links. Statistical measures indicate that Bayes net methods succeed in finding relevant correlations. There is a trade-off between statistical power and computational feasibility (full table search vs constrained search). Hierarchical search often does well on both dimensions, but needs to be extended to handle correlations conditional on the absence of relationships.

References

[1] D. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

[2] David Poole. First-order probabilistic inference. In *IJCAI*, pages 985–991, 2003.

[3] Oliver Schulte. A tractable pseudo-likelihood function for Bayes nets applied to relational data. In *SIAM SDM*, pages 462–473, 2011.

[4] Oliver Schulte and Hassan Khosravi. Learning graphical models for relational data via lattice search. *Machine Learning*, 88(3):331–368, 2012.

[5] Yizhou Sun and Jiawei Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*, volume 3. Morgan & Claypool Publishers, 2012.

Semantic Modeling for Big Data Integration

Craig A. Knoblock and Pedro Szekely
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

We are developing an environment where developers compose various sources and tools to analyze a wide range of big datasets. This environment will be effective to the extent that data can flow seamlessly from one tool to another. But this is no easy task. The problem is that most analytic tools require data to be organized in a particular way using specific formats. For example, network analysis tools require data to be represented as network graphs; time-series analysis tools require data to be organized around the temporal information in the data; visualization components typically have their own requirements for the data. There is typically a mismatch between the input and output requirements of the datasets and tools.

The problem of data cleaning and transformation is a huge problem today, even when dealing with small to moderate sized datasets, but especially for big datasets. The usual approach is to perform the transformations manually or write programs to do this, but this requires tremendous time and effort, is prone to errors, and does not scale to the large datasets. In prior work, we used Karma to solve data preparation problems for environmental scientists working on simulating stream metabolism. They use data from a variety of sensors and data repositories from various government agencies. Even though the datasets are small, containing only thousands of records, prior to our involvement it took scientists weeks and months to prepare the data. They used a combination of manual editing and custom scripts to clean, normalize, resample, integrate and restructure the data to satisfy the requirements of their simulation models. Using Karma we were able to generate scripts that automate the process, enabling them to automatically run their analysis every day with the most recent data [1].

We are developing a data cleaning and transformation approach that will free developers from the time-consuming and error-prone work of cleaning and reshaping the data to satisfy the data assumptions of the analysis and visualization tools. The overall approach is shown in Figure 1. Our tool will (1) automate much of the data transformation, (2) provide an easy to use interface that allows developers to specify the parts of the process that need to be refined, and (3) efficiently execute the required transformations on big datasets. Our tool will reduce the time and effort it takes for developers to define a required set of data transformations and support the execution of those transformations at scale, thus enabling developers to focus on the analysis workflows, trying different tools, different parameters, etc. in order to optimize the analyses for the end users.

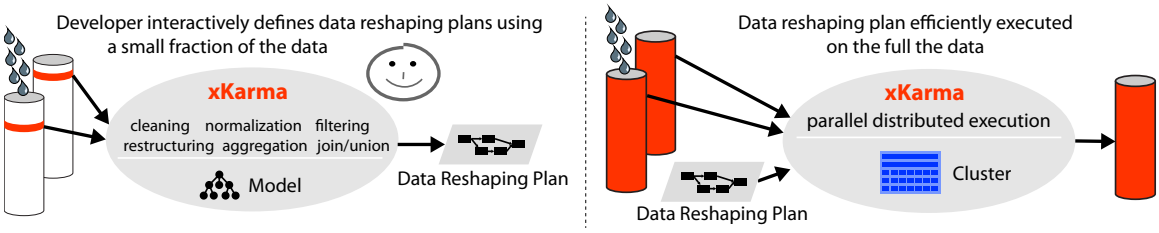

Figure 1: xKarma allows a user to rapidly build and execute data integration plans

Our approach is based on our existing Karma semantic integration tool [4, 2, 3] that already solves significant aspects of this problem. The key insight in our approach is that in Karma we developed techniques to semi-automatically build semantic descriptions (models) of the data. Furthermore, our techniques build these models of datasets based on small samples of a few hundred records. Users of our tool will define their data shaping tasks using examples in an interactive easy to use interface, reaping the benefits of these semantic models that enable Karma to provide a level of assistance that is not possible based on purely syntactic models. Karma then generates the plans that implement the transformations that can be run offline on large datasets. For example, consider the situation where an analysis tool requires a set of data in a particular format, but the required data is actually spread over multiple datasets and in a very different format. Our system would use the knowledge of the input requirements of the analysis tool to interactively work with a user to quickly define a transformation plan on a small subset of the data and then convert that into a general transformation plan that is then efficiently executed on the input datasets.

References

[1] Yolanda Gil, Pedro Szekely, Sandra Villamizar, Thomas C. Harmon, Varun Ratnakar, Shubham Gupta, Maria Muslea, Fabio Silva, and Craig A. Knoblock. Mind your metadata: Exploiting semantics for configuration, adaptation, and provenance in scientific workflows. In *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[2] Craig A. Knoblock, Pedro Szekely, Jose Luis Ambite, , Aman Goel, Shubham Gupta, Kristina Lerman, Maria Muslea, Mohsen Taheriyan, and Parag Mallick. Semi-automatically mapping structured sources into the semantic web. In *Proceedings of the Extended Semantic Web Conference*, Crete, Greece, 2012.

[3] Mohsen Taheriyan, Craig A. Knoblock, Pedro Szekely, and Jose Luis Ambite. Rapidly integrating services into the linked data cloud. In *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, 2012.

[4] Rattapoom Tuchinda, Craig A. Knoblock, and Pedro Szekely. Building mashups by demonstration. *ACM Transactions on the Web (TWEB)*, 5(3), July 2011.

High Quality Data Generation: An Ontology Reasoning based Approach * (Extended Abstract)

Yue Ma **Julian Mendez**
Institute of Theoretical Computer Science, TU Dresden, Germany
{mayue,mendez}@tcs.inf.tu-dresden.de

Abstract

As Big Data is getting increasingly more helpful for different applications, the problem of obtaining reliable data becomes important. The importance is more obvious for domain specific applications because of their abstruse domain knowledge. Most of the Big Data based techniques manipulate directly datasets under the assumption that data quantity can lead to a good system quality. In this paper, we show that the quality can be improved by automatically enriching a given dataset with more high-quality data beforehand. This is achieved by a tractable reasoning technique over the widely used biomedical ontology SNOMED CT. Our approach is evaluated by the scenario of formal definition generation from natural language texts, where the average precision of learned definitions is improved by 5.3%.

Domain specific Big Data is of particular interest for domain specific applications. SNOMED CT [SNOMED *Clinical Terms*, 2006,] is one of such data (a.k.a. ontology) and now a widely accepted international standard. It describes concepts such as anatomical structures, disorders, and organisms. It has been adopted worldwide as a standard for electronic health records and is also used in clinical decision support systems. Users can access SNOMED CT through browsers such as NIH Browser (cf. Table 1 for the formal definition of the concept Baritosis given in SNOMED CT).

From Table 1, we can see that an important aspect for SNOMED CT development is to predict relationships among concepts, which can enable the formal definition generation of concepts (e.g. from texts) [Ma and Distel, 2013b; Tsatsaronis *et al.*, 2013; Ma and Distel, 2013a]. Unlike traditional text mining problems, due to the data quantity and complexity, it is unrealistic to manually build a training data to be used to predict new relationships among SNOMED CT concepts. Since the emergence of Big Data, such as Freebase or DBpedia, this can be partially solved by a new learning approach named *distance supervision* [Mintz *et al.*, 2009] based on the assumption that the quantity of Big Data can guarantee the quality of the automated annotation step. In this paper, we show that the system quality for predicting SNOMED CT relationships can be further improved by automatically enriching the SNOMED CT relationships before being used in the learning phase based on distance supervision.

Table 1: Baritosis as displayed by NIH Browser

Concept: [50076003] Baritosis

Relationships from *this* concept (9)

Baritosis	Causative agent	Barium dust	(Defining)
Baritosis	Associated morphology	Deposition of foreign material	
Baritosis	Finding site	Lung structure	(Defining)
Baritosis	Associated morphology	Inflammation	
Baritosis	Finding site	Lung structure	(Defining)
Baritosis	Is a	Pneumoconiosis due to inorganic dust	
Baritosis	Clinical course	Courses	(Qualifier)
Baritosis	Episodicity	Episodicities	(Qualifier)
Baritosis	Severity	Severities	(Qualifier)

The Distance Supervision based on Big Data

The general framework of distance supervision for text mining can be described as follows:

- Data projection: given a textual corpus $\mathcal{T}$ and a Big Data set $\mathcal{D}$, find the occurrences of instances of $\mathcal{D}$ in $\mathcal{T}$, thus obtaining the annotation data.
- Model learning: once the annotation data is ready, a supervised learning approach is applied to extract a model.
- Instance prediction: the learned model will be used to predict property of test data.

For the data projection process, when $\mathcal{D}$ is a Big Data, usually there are automated annotation tools with a good quality, such as the DBpedia Spotlight tool [Mendes *et al.*, 2011] for DBpedia data and the *Metamap* developed at the U.S. National Library of Medicine for SNOMED CT. Moreover, depending on the tasks, $\mathcal{D}$ can be of different forms besides mere concepts as handled by DBpedia Spotlight and Metamap. Since we aim to extract relations between concepts in this paper, $\mathcal{D}$ should be concept relationships. Due to the continuous development of SNOMED CT since 1965, there are already more than 1,360,000 relationships in SNOMED CT as of 2011. However, we distinguish two usages of these relationships, explicit and inferred as detailed below.

*We acknowledge financial support by DFG in the Research Unit FOR 1513 project B1 and in the Collaborative Research Center 912 "Highly Adaptive Energy-efficient Computing".

Explicit and Inferred Role Bases

The explicit role base ExpRB contains all relationships among concepts that are explicitly given in the description of concepts in SNOMED CT. For instance, by Table 1, we have Baritosis|Causative_agent|Barium_dust is an explicit one.

Reasoning provides a way to use implicit information encoded in SNOMED CT. The inferred role base InfRB is achieved through a tractable Description Logic (DL) reasoning engine as follows: $\mathsf{InfRB} = \{A|R|B : \text{SNOMED} \models A \sqsubseteq \exists R.B\}$, where $\models$ is the logical entailment under DL semantics which is tractable for $\mathcal{EL}$++ [Baader *et al.*, 2005], the logic language underlying SNOMED CT. By this, we have Baritosis|Causative_agent|Dust as an inferred relationship since Barium_dust is a subclass of Dust by SNOMED CT. We achieve this by the optimized reasoner `jcel`[1][Mendez, 2012] which can classify SNOMED CT in 13 minutes. By the monotonicity of DL semantics, we have ExpRB $\subseteq$ InfRB.

Once a role base is fixed, annotated sentences can be aligned with a relationship if they contain two concepts that have a relationship according to the role base. For example, we have the following sentence annotated by Metamap: "*Baritosis*/Baritosis_(disorder) is pneumoconiosis caused by *barium dust*/Barium_Dust_(substance)", where "Baritosis" and "barium dust" are annotated with concepts Baritosis_(disorder) and Barium_Dust_(substance), respectively. Both ExpRB and InfRB contain the relationship Baritosis_(disorder)|Causative_agent|Barium_dust_(substance). The sentence is thus aligned with the role Causative_agent, with the latter being an aligned role. By using either ExpRB or InfRB as the role base, different training data are obtained. Since ExpRB $\subseteq$ InfRB, we can see that the training data by aligning with ExpRB can only be a subset of that by aligning with InfRB for any given set of textual sentences.

Evaluation and Discussion

For data projection, we use Metamap to identify SNOMED CT concepts in a sentence and the explicit and inferred role bases for sentence alignment as described above. Following [Ma and Distel, 2013b], we use Stanford classifier [Manning and Klein, 2003] to train a relation extraction model (cf. [Ma and Distel, 2013b] for more details). The aim is to test the effectiveness of different SNOMED CT role bases (see Table 2 for their values for three example roles) serving as the Big Data set for SNOMED CT concept formal definition construction.

Table 2: Sizes of the Explicit and Inferred Role Bases for Associated_morphology (AM), Causative_agent (CA), and Finding_site (FS)

	AM	CA	FS
ExpRB	503306	91794	1306354
InfRB	32454	13225	43079

In the experiment, we take the concepts that are descendants of Disease(disorder). Among the 65,073 descendants, 1305 concepts are mentioned in our text corpus and thus considered in the evaluation. A one-concept-leave-out evaluation is used, that is, each round of experiments removes one concept. The removed concept is used as the target concept in the learning process whose formal definition is to be predicted. The learned definition is then compared to the original as given in SNOMED CT to measure the system's quality.

Our text corpus is obtained by querying Wikipedia with one-word SNOMED CT concept names, resulting in around 53,943 sentences with 972,038 words. For each target concept, different training data are constructed by data projection from ExpRB and InfRB respectively. The test data are the same for a fair comparison of the effectiveness of different role bases. As the evaluation measure, we used the reasoning based precision as defined in [Ma and Distel, 2013a] for formal definition generation.

Under the experiment setting detailed above, we obtained a 66.71% average precision when InfRB is used and 61.41% average precision when ExpRB is used. The 5.3% average precision increment shows that the automatically enriched dataset improves the system's quality for predicting formal definitions for SNOMED CT concepts, which will be beneficial for automatic ontology learning [Ma and Distel, 2013a], which requires high quality extracted information from texts.

In the future, the differences between explicit and inferred role bases will be further tested with other algorithms as exploited for the same task [Tsatsaronis *et al.*, 2013]. We will also analyze the proposed approach for other Big Data based applications which involve large structured data sets.

[1]http://jcel.sourceforge.net/

References

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of IJCAI'05*. Morgan Kaufmann, 2005.

[Ma and Distel, 2013a] Yue Ma and Felix Distel. Concept adjustment for description logics. In *Proceedings of K-CAP'13*, 2013.

[Ma and Distel, 2013b] Yue Ma and Felix Distel. Learning formal definitions for Snomed CT from text. In *Proceedings of AIME'13*, 2013.

[Manning and Klein, 2003] Christopher Manning and Dan Klein. *Optimization, Maxent Models, and Cond. Est. without Magic*. Tutorial at HLT-NAACL'03 and ACL'03, 2003.

[Mendes *et al.*, 2011] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings of I-Semantics'11*, 2011.

[Mendez, 2012] Julian Mendez. jcel: A modular rule-based reasoner. In *Proc. of the 1st Int. Workshop on OWL Reasoner Evaluation (ORE 2012)*, vol. 858 of CEUR, 2012.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL/AFNLP'09*, pages 1003–1011, 2009.

[SNOMED *Clinical Terms*, 2006] SNOMED *Clinical Terms*. Northfield, IL: College of American Pathologists, 2006.

[Tsatsaronis *et al.*, 2013] George Tsatsaronis, Alina Petrova, and et al. Learning formal definitions for biomedical concepts. In *Proceedings of OWLED'13*, 2013.

Rethinking big data: the role of artificial intelligence and machine learning

Randy Goebel
Alberta Innovates Centre for Machine Learning
Department of Computing Science, University of Alberta
Edmonton, Alberta T6G 2E8, Canada
rgoebel@ualberta.ca

Extended Abstract

It seems that all current data management endeavours impinge on the capture and use of data — whether in science, engineering, industry or government — now consistently refer to "big data" if they refer at all to data. This use of the term "big data" is now common, but the breadth of possible interpretations is enormous. Included in this variety of interpretations are the identification of the attributes of *velocity, variety, volume, veracity* [Zikopoulos *et al.*, 2005][Thanos *et al.*, 2012], or the transition from OLTP-OLAP-RTAP in the evolution of data mining (e.g., [Heising, 2010]), or the technology-response definitions like MapReduce [Dean and Ghemawat, 2004] and HaDoop [Shafer *et al.*, 2010], or even vendor-driven interpretations like the application development model of IBM's InfoSphere stream mining [Rea and Mamidipaka, 2010].

The evolution of data analytic methods, from those labelled as "online transaction processing" (OLTP), to "online analytics processing" (OLAP), to so-called "real time analytics processing" (RTAP) are driven by two key observations: 1) that as the growth of data accelerates, static analysis of complete data sets becomes obsolete, and 2) that analysis of large data streams must consider analytic methods deployed in temporal data streams constrained by computational resources (see Figure 1)

From one viewpoint, it isn't rocket science to consider incremental dynamic characterization of data streams. Packet sniffers and Internet protocol (IP) data characterization systems have been doing this for years. But from a more general viewpoint, the need to develop more elaborate RTAP has encroached on new data landscapes, where the data streams are not just fast and voluminous, but simply impossible to store.

Our characterization of "big data" is based on examples of data accumulation methods where it seems, in principle, impossible to store the data stream; and so the analytics, whether OLAP or RTAP or any variant, is not just a strategy for extracting interesting signals, but is in fact a necessary component of data management infrastructure. Three cases where the volume of data anticipated are given in Figure 2, and arise from the areas of radio astronomy [SKA, 2012], genome sequencing[Polonsky *et al.*, 2007], and carbon flux sensors [Porter *et al.*, 2012].

In these examples, the consensus is that the volume and velocity of data is so large that there is no capture alternative.

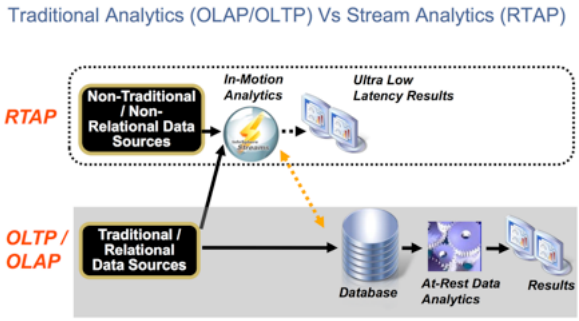

Figure 1: Evolution of data analytics (from [Heising, 2010], page 6)

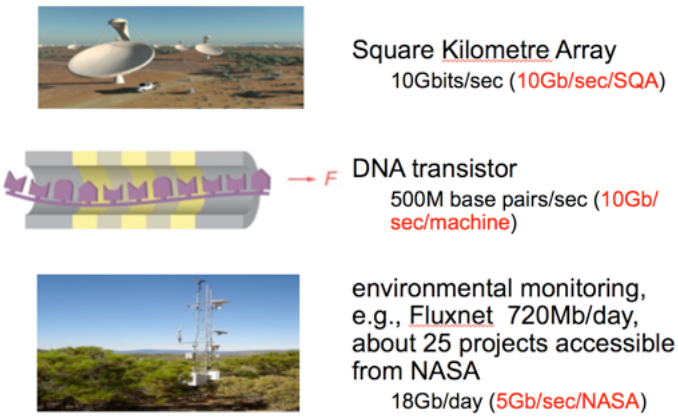

Figure 2: Examples of "Big Data"

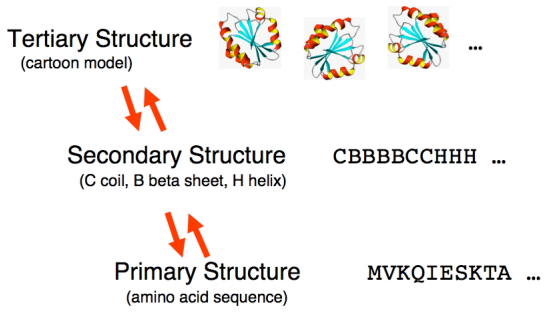

Figure 3: An abstraction of three levels of protein representation

But there remains the scientific challenge of confirming anticipated hypotheses within these data. The underlying analytics tasks are large scale scientific challenges, where the quest is not just for expected artifacts in the data, but for scientific insight for which all uninterpreted data may be crucial.

We this as background, the hypothesis sketched here is as follows. Given the view that big data means unstorable data, the challenge is to compress as much of a large data stream as possible, as abstraction labels, and then retain as much as feasible of the remaining data stream as the basis for deeper analysis and detection of new scientific concepts. The process of this semantic compression is contingent upon two foundations of artificial intelligence: 1) knowledge representation, especially for multi-scale scientific modelling, and 2) machine learning methods to construct classification methods to label or "chunk" data stream portions into identified components of such multi-scale models.

A simple example arises in the task of inducing the secondary and tertiary structure of proteins (see Figure 3). With even the simple multi-scale model at three levels of vocabulary abstraction (amino acids; beta-sheets, random coils, alpha-helices; 3D cartoon models), one can imagine that a stream of amino acids can be dynamically compressed to the vocabulary higher in the multi-scale model, so that the lower level data no longer needs to be retained.

Of course this hierarchical multi-level compression requires first a compression model in the form of a domain-specific model, as well as the machine learning processing to do the classification of the lower levels to the upper levels.

When such mappings are available, a data stream can be compressed to the known vocabulary labels. In the cases cited above, where it is likely that not all of the data can be immediately compressed by such labelling, then the lower level data stream must be retained for further analytical processing.

We sketch the requirements for these kinds of systems which combine the existing ideas of stream mining with the anticipated automatic construction of multi-scale models that can be deployed to help manage the unstorable volumes of data. From one viewpoint, the combination of using machine learning to both build and extend multi-scale models, and to compress large data streams using those models, is a kind of automation of the scientific method for interpreting data. The only perhaps novel aspect is that obtained in the context of unstorable data streams: we have no alternative to building systems in this way.

The fundamental challenges are to incrementally and dynamically use machine learning augmented by domain knowledge to build and maintain the multi-scale models, while simultaneously using machine learning directly to use the multi-scale models to classify the data stream, to reduce its size within any given resource constrains (e.g., like available storage) while retaining as much of the uninterpreted or unclassifiable data as possible, for exploitation in deeper scientific analysis.

References

[Dean and Ghemawat, 2004] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. http://research.google.com/archive/mapreduce.html, 2004.

[Heising, 2010] W. Heising. Stream computing: the evolution of data analysis. http://www.bwdb2ug.org/PDF/InfoSphere-Streams-usersgroup-12-7.pdf, 2010.

[Polonsky *et al.*, 2007] Stas Polonsky, Steve Rossnagel, and Gustavo Stolovitzky. Nanopore in metal-dielectric sandwich for dna position control. *Applied Physics Letters*, 91(15), 2007.

[Porter *et al.*, 2012] John H. Porter, Paul C. Hanson, and Chau-Chin Lin. Staying afloat in the sensor data deluge. *Cell*, 27(2):121–129, 2012.

[Rea and Mamidipaka, 2010] Roger Rea and Krishna Mamidipaka. Ibm infosphere streams: Enabling complex analytics with ultra-low latencies on data in motion. http://www.monash.com/uploads/IBM-InfoSphere-Streams-White-Paper.pdf, 2010.

[Shafer *et al.*, 2010] J. Shafer, S. Rixner, and A.L. Cox. The hadoop distributed filesystem: Balancing portability and performance. In *Performance Analysis of Systems Software (ISPASS), 2010 IEEE International Symposium on*, pages 122–133, 2010.

[SKA, 2012] SKA. Exploring the universe with the world's largest radio telescope. Square Kilometre Array Consortium, available at http://www.skatelescope.org/the-science/, 2012.

[Thanos *et al.*, 2012] Costantino Thanos, Stefan Manegold, and Martin L. Kersten. Big data - introduction to the special theme. *ERCIM News*, 2012(89), 2012.

[Zikopoulos *et al.*, 2005] P. Zikopoulos, D. Deross, K. Parasuraman, T. Deutsch, D. Corrigan, and J. Giles. *Harness the power of big data*. McGraw-Hill, 2005.

On Distributed Constraints and Big Data

Pedro Meseguer
IIIA - CSIC, Campus Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
pedro@iiia.csic.es

Constraint Programming (CP) [2] is an AI technique that has received substantial attention in the last 30-40 years obtaining substantial success [1]. Originally centralized, on the verge of XXI century the distributed version of the classical *constraint satisfaction problem* (CSP) was proposed and solved in the pioneer work of Makoto Yokoo and colleagues [4]. Briefly, a distributed CSP occurs when different parts of the problem are distributed among different agents (with computing capabilities) and cannot be joined into a single one for several reasons. Each agent knows a part of the problem, but no agent knows the whole problem. A solution is found by message passing among the agents. After Yokoo's work, different approaches have been proposed to solve this kind of problems considering both satisfaction and optimization versions.

Big Data is an umbrella term to denote computing activities that deal with massive amounts of data. Some sources of big data are large scale e-commerce, social networks, science (for instance, consider experiments in astronomy or biology), and Internet.

In some cases CP applications are faced with BD. Distribution has been considered as a possible technique to deal with the issues that BD poses to CP [3]. In this note, we want to go deeper in this relationship, to identify trends in distributed constraint satisfaction/optimization of special interest for handling BD instances.

Exact vs. Approximate Solving. First thing that comes to mind is *exact* against *approximated* resolution: can BD instances be solved approximately, or the exact solution is required? Is strict global consistency needed? Equivalently, is the strict global optimum required, or a good quality suboptimum could be enough? Without trying to be too conclusive here, we guess that there is room for approximate solutions. Several reasons support this idea. The most obvious one is that exact solution may require an exponential time in the size of the data (worst case), and if the amount of data is huge, this is simply not feasible. [1] Another reason is that it could be preferable to assure some degree of consistency –although this consistency were not completely global– than no consistency at all. Finally, BD instances are real-world problems and in these cases, finding the exact solution (global optimum) is not as important as in academic problems, typically more theoretically-oriented.

[1] If the data considered satisfy some properties (for instance, a small induced width) exact resolution might still be feasible.

Agent Granularity. A simplistic view of the task done by agents in distributed constraint satisfaction/optimization is that it is composed of two main elements:

1. Intraagent consistency: agent selects values for its variables trying to maximize its internal consistency.
2. Interagent consistency: agent selects values for its variables trying to maximize the consistency with other agents.

These two elements are not independent and some criteria exist to solve potential conflicts. Typically, intraagent consistency is subordinated to interagent consistency, which is a reasonable option since computation inside an agent is cheaper than computation involving several agents, including message passing, etc. Most existing work on distributed constraint satisfaction/optimization is focused on the second point (interagent consistency), ignoring –to a large extent– issues related with intraagent consistency (because centralized approaches can be applied inside each agent). It is worth noting that the second point is more costly than simply enforcing consistency among the variables of the different agents under a centralized view, that is, enforcing interagent consistency has a extra overhead due to distribution that is not present in centralized approaches.

A common assumption in distributed constraint satisfaction/optimization is that each agent holds a variable. We claim that this assumption is no longer valid for BD instances. The obvious reason is size: one cannot afford "one agent, one variable" when the number of variables is huge, because the number of agents would also be huge, causing very serious implementation problems (number of computers, overhead, communication issues, etc). In addition, there is another reason to remove that assumption. When advocating distribution for BD in CP, we are expecting to obtain benefits with respect to a centralized approach. This is because we expect that a substantial amount of work could be done concurrently on different computers, and that work concurrently done remain valid for the final solution. If an agent is limited to a variable, enforcing intraagent consistency originates practically no work once a consistent domain has been set for that variable, and all the attention is focused on enforcing interagent constraints. But if comparing this with a cen-

tralized approach, the work done to enforce interagent constraints is slower (done by message passing), more expensive (involves communication), and it is highly changeable (because of backtracking). Enforcing intraagent consistency concurrently may offer good opportunities to gain practical efficiency, and it does not seem to be a good idea to eliminate it (as implied by that assumption).

Synchronous vs Asynchronous Algorithms. Another dimension in distributed constraint satisfaction/optimization is about which type of distributed algorithms are more suitable. According to the way agents take their decisions, distributed algorithms are divided in two main classes: *synchronous* and *asynchronous* (intermediate options also exist). Typically, a synchronous algorithm occurs when agents take their decisions following some kind of external signal, or waiting for some external event to occur. Alternatively, agents following an asynchronous algorithm do not wait for particular events to take their decisions: a agent takes its own decisions and the other agents are ready to react to any decision taken by that agent (this occurs concurrently for all agents in the system). Broadly speaking, synchronous algorithms synchronize their actions following some kind of "external clock" (and in many cases, they can be seen as distributed versions of existing centralized algorithms), while asynchronous algorithms are closer to a kind of "chaotic iteration", where agents take decisions and try to adapt themselves to the decisions of others. Typically, under synchronous algorithms agents take decisions based on an updated view of the other agents, while less updated views occur under asynchronous ones. This has impact in the quality of the decisions that agents take, and the actions for undoing that agents have to perform when their view is correctly updated. Synchronous algorithms seem to be less concurrent (= offer less opportunities for concurrency) than asynchronous ones.

There is some debate in the distributed constraint satisfaction/optimization community about which type of algorithm is more adequate for which problems. Considering BD problems, its seems that the most crucial point is concurrency, the more work to be done in parallel the better for efficiency. Asynchronous algorithms are the ones that offer a higher degree of concurrency. In this sense, they seem to be more appropriate than synchronous ones to solve BD problems. Anyway, experimental work is needed to substantiate this claim.

Conclusion. As executive summary, this note considers the use of distributed constraint satisfaction/optimization techniques for dealing with BD problems inside CP. In that case, a promising direction seems to be looking for approximate solutions, using asynchronous algorithms with agents handling several (many) variables. We believe that these options reinforce mutually, looking for a feasible processing of BD instances that involve constraints. More work, specially of experimental nature, is needed to confirm the discussed points.

References

[Puget, 1998] J.-F. Puget. Constraint programming: A great AI success. *Proc. 13th ECAI*, pages 698–675, 1998.

[Rossi *et al.*, 2006] F. Rossi, P. Van Beek, and T. Walsh. *Handbook of Constraint Programming*. Elsevier, 2006.

[Yap, 2012] R. H. C. Yap. Constraint programming in the age of Big Data?, 2012. Position paper at CP 2012 conference on the Future of CP.

[Yokoo *et al.*, 1998] M. Yokoo, E. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Trans. Know. and Data Engin.*, pages 673–685, 1998.