VoidWiz: Resolving Incompleteness Using Network Effects

Christina Christodoulakis [#1], Christos Faloutsos [*2], Renée J Miller [#3]

[#] *Department of Computer Science, University of Toronto*
Toronto, Canada
{[1]christina,[3]miller}@cs.toronto.edu

[*] *School of Computer Science, Carnegie Mellon University*
Pittsburgh, Pennsylvania, USA
[2] christos@cs.cmu.edu

***Abstract*—If Lisa visits Dr. Brown, and there is no record of the drug he prescribed her, can we find it? Data sources, much to analysts' dismay, are too often plagued with incompleteness, making business analytics over the data difficult. Data entries with incomplete values are ignored, making some analytic queries fail to accurately describe how an organization is performing. We introduce a principled way of performing value imputation on missing values, allowing a user to choose a correct value after viewing possible values and why they were inferred. We achieve this by turning our data into a graph network and performing link prediction on nodes of interest using the belief propagation algorithm.**

I. Introduction

Real data sources are plagued with incompleteness, hindering analysis, decision making, and insight. The missing data can be a result of user error, system failure, or data integration and exchange [1]. Many have addressed the value imputation problem, that can be described as a link prediction problem. We argue that a human must be kept in the loop when it comes to cleaning the data, and this means an intuitive interface is needed to view and understand the data, allowing a data analyst to control the imputation process and make his final decisions based on a thorough understanding of the provenance of the suggested values. We introduce *VoidWiz*[1], a system that reads relational data, translates it into a hypergraph and displays it for the data analyst. *VoidWiz* leverages an elegant machine learning algorithm, called belief propagation [2] to discover missing links in the graph and therefore suggest values for incomplete tuples. The system allows the user to make an informed decision on the value or link to be kept, keeping her in control of the data cleaning process. Our contributions are: principled value imputation using network effects, visualization of provenance, interaction with tabular and network data, and application on real world data.

II. System Overview

VoidWiz provides imputation and provenance, overview of datasets, and details on demand. We introduce our system with a toy example, and go on to show how it works with a larger real world dataset. We provide an overview of the mathematical foundations of the belief propagation algorithm used for link prediction, and we describe the implementation of the system for the purpose of reproducability.

A. Toy Example: Doctor-Patient Visits

We assume a simple relation, Fig. 1, that describes patient visits to doctors, and states a condition they are diagnosed with, and a drug prescribed as treatment. Rows with missing values are highlighted in red to draw user attention.

Tabular View

#	Patient Name	Doctor Name	Diagnosis	Treatment
1	Lisa	Dr. Brown	Emphysema	A
2	Alice	Dr. Smith	Acne	B
3	Lisa	Dr. Brown	Emphysema	
4	Tom	Dr. Smith	Rash	C
5	Alice	Dr. Brown	Asthma	D

Fig. 1. *VoidWiz* offers visualization of patient visits as a relational table, highlighting in red the tuple with a mising value.

VoidWiz translates the relational data into a hypergraph in Fig. 2, and displays it in an interactive interface. The visual aid of the hypergraph shows the user that certain nodes are interlinked heavily, while others are not. The user may pan the graph or zoom in to details she is interested in. When the user wants to find possible values for the missing data, *VoidWiz* runs belief propagation on the hypergraph, suggesting values and displaying the belief that those values would be used, Fig. 3. *VoidWiz* displays values (of the appropriate attribute type) ranked by highest belief. The user can select the value she wants, and visualize the updated hypergraph, with the new link displayed in yellow, Fig. 4. The relational table is also updated with the selected value, and marked with yellow.

B. Real World: Clinical Trials

We demonstrate the use of *VoidWiz* in a real dataset describing international clinical trials. The dataset is obtained from the Linked Clinical Trials (LinkedCT) database [3]

[1]Demo available at http://www.cs.utoronto.ca/∼christina/apps/VoidWiz/

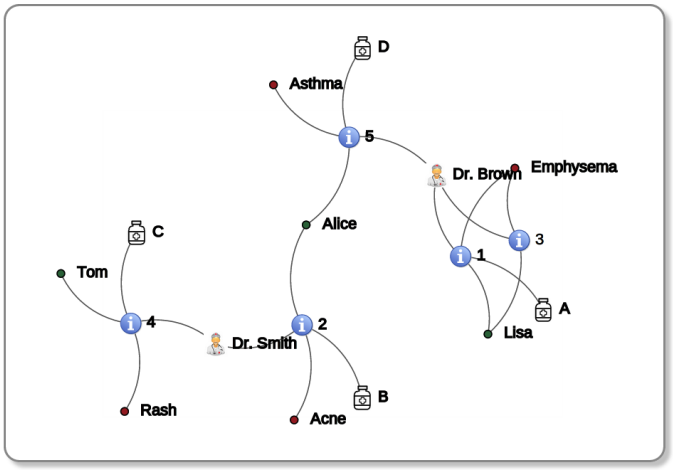

Fig. 2. *VoidWiz* visualizes the tabular data describing patient-doctor visits as a hypergraph.

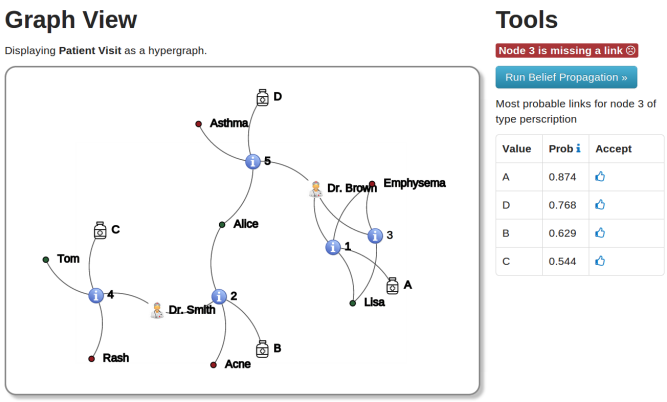

Fig. 3. Lisa's visit to Dr. Brown with visit id 3 is missing the prescribed drug. *VoidWiz* provides a list of nodes that are of the missing attribute type for this visit in order of belief.

which provides an open web data source of clinical trials data, originally published on ClinicalTrials.gov[2]. It is a large repository of clinical trials from various countries provided by the U.S. National Library of Medicine. The XML data was transformed into a set of relational tables. For simplicity we narrow the dataset down to four attributes, the clinical trial identifier, the source conducting the trial, the conditions under examination, and the drugs tested. *VoidWiz* displays this data in tabular format, Fig. 5, allowing the user to sort rows by attribute, and also as a hypergraph in Fig. 6, styling nodes by attribute type. As before, visualizing the data as a hypergraph allows the user to identify certain properties of the data, view heavily connected nodes, isolated nodes, etc. *VoidWiz* follows the Shneiderman mantra on information visualization [4], offering *'overview first, zoom and filter, then details-on-demand'*.

We allow for search over the hypergraph, providing the user with a search box that guides the user with autocomplete

[2]http://clinicaltrials.gov/

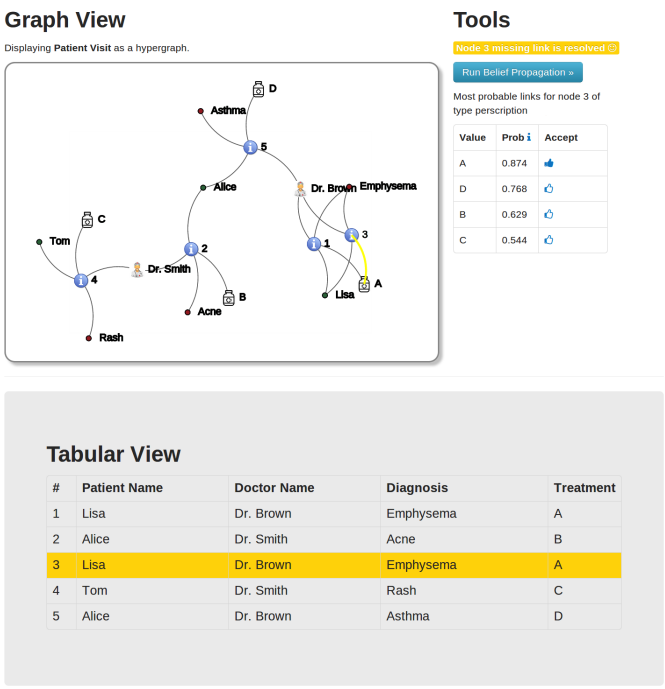

Fig. 4. *VoidWiz* displays the newly created link for the user on the hypergraph and in the table and awaits confirmation.

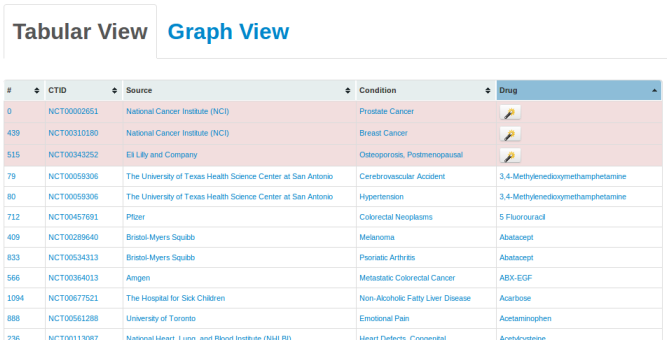

Fig. 5. *VoidWiz* displays an *'overview'* of the data first. Above we see clinical trials in Canada in tabular format, sorted on attribute drug.

suggestions as she types. The node the user has been searching for is zoomed into, enlarged and highlighted in yellow, as shown in Fig. 7. This is done to ensure visibility when the user zooms out to view the entire graph. To allow for further exploration, on node selection user can specify a measure of filtering the graph to neighboring nodes of maximum k-hops away of selected node.

During the execution of link prediction algorithms, it can be small portions or large portions of the graph that lead to the final listing of suggested links. Data analysts need to to trust the values they are using do repair incompleteness in the data. They can gain this trust by having an intuition of how the algorithm arrives at the suggested links. The system must provide an intuitive visualization to communicate the provenance of suggested values to a user without confusing her

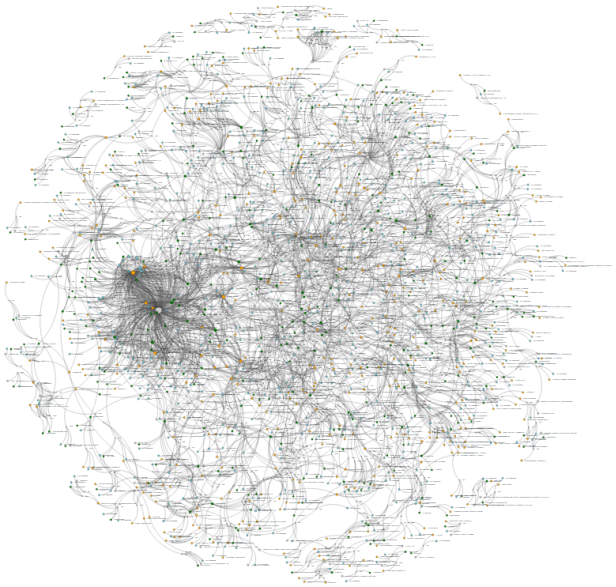

Fig. 6. *VoidWiz* displays an *'overview'* of the clinical trials data as a hypergraph, allowing user to spot interesting properties such as densely or sparcely interlinked data. User is able to pan, filter, and zoom into the data.

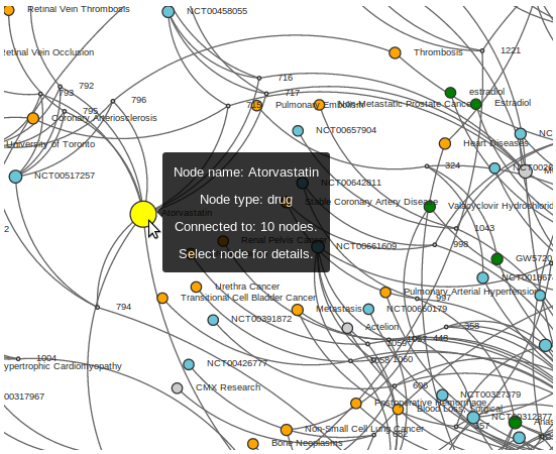

Fig. 7. *VoidWiz* allows for *'details on demand'* with tooltips that appear when the user hovers over a node of interest. In this case user searched for the drug Atorvastatin , *VoidWiz* zooms in and highlights the node, and on hover displays neighboring node information. The user can select the node to filter to only immediate context.

with a massive table of evidence, Fig 8, 9. This can be done by highlighting portions of the graph, or indicating parts of the graph structure that lead to the link prediction algorithm's suggestions. Prior to running the value imputation process, the user can specify if the algorithm should take into account all the data, including imputed and user specified values, or a subset of that data. In the future, this could allow for training the system to deal with dynamic streams of data [5]. Currently, the possible suggested values are limited to the existing values in the dataset. We aim to enhance the hypergraph by adding to it ontologies of the domain under examination. This will ensure that the pool of suggested values comes from a well defined set of values for that domain.

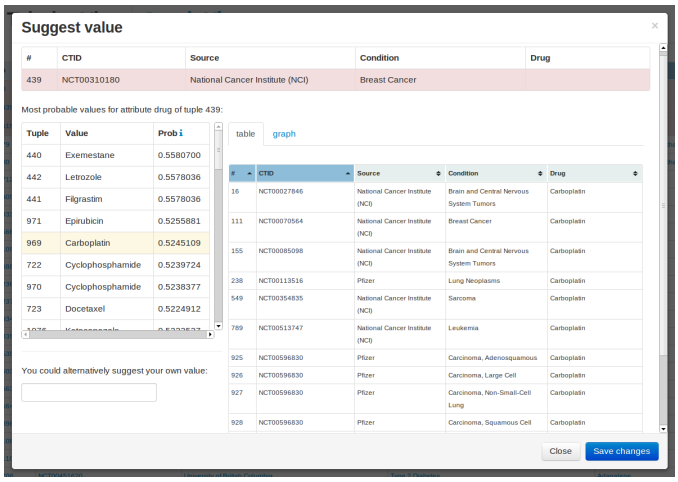

Fig. 8. The value suggestion wizard displays the incomplete tuple of interest, and the results of the belief propagation algorithm with suggested values, and the context of that value in tabular and hypergraph format.

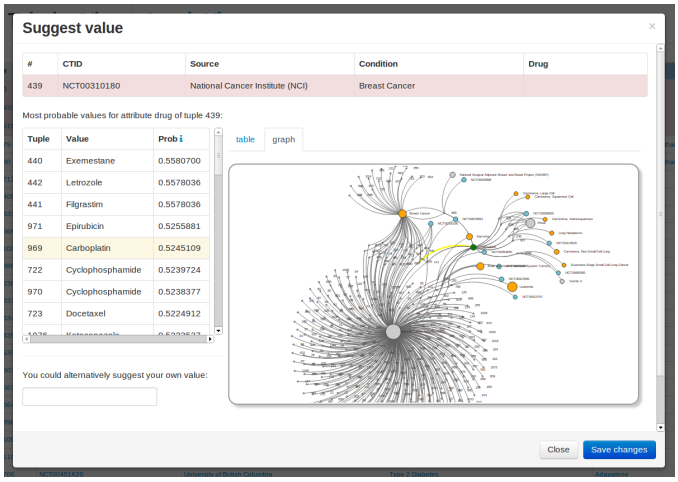

Fig. 9. The value suggestion wizard displaying context of suggested values as a hypergraph.

C. Math Foundations

We use the standard belief propagation equations from Yedidia et al [2]:

$$p(\{x\}) = \frac{1}{Z} \prod_{(i,j)} \psi_{ij}(x_i, x_j) \prod_{(i)} \phi(x_i) \tag{1}$$

$$b_i(\{x_i\}) = k\phi(x_i) \prod_{j \in N(i)} m_{ji}(x_i) \tag{2}$$

$$m_{ij}(\{x_j\}) \leftarrow \sum_{x_i} \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i) \tag{3}$$

Variable $m_{ij}(x_j)$ is used to describe messages between hidden nodes i and j about the believed state node j should be in. Belief at node i is described as $b_i(x_i)$, and the joint probability distribution for unknown variables x_i is $p(\{x\})$. In addition, similar to [6], *VoidWiz* will use connection subgraphs for the newly proposed links, to provide provenance and convey to the user the influence of each node of the network in predicting the link.

D. Implementation and Reproducability

The *VoidWiz* system is implemented in Python, Java and JavaScript. It uses the D3 (Data-Driven Documents) JavaScript library [7] for visualizing the hypergraphs. The graph data is created by parsing the relational tables and creating a csv file with <source, target> id's, where each unique value-attribute pair is assigned a unique id. We implement belief propagation as described in [2]. We provide the interactive interface as a web application accessible by any modern browser. We are open sourcing our code[3], with instructions for reproducing the demo.

III. Demonstration plan

In our demonstration, we will invite the audience to interact with *VoidWiz* and try out its capabilities on both the toy dataset and the clinical trials dataset. We will prompt the audience to become data analysts and look for occurrences in the data where there are missing values. Users will use the system to find suggested values and select the most intuitive answer. Users may also perform exploratory analysis, and see for example what conditions a drug could be tested for or what conditions a source ought to perform trials on. Users may easily search for a condition of interest and inspect all affiliated sources, trials, and drugs at a glance.

IV. Related work

As mentioned, there has been a lot of work on the link prediction problem in the past, with multiple applications [8]. Similar to belief propagation, suggested techniques include random walks [9], random walks with restart [10], [11], personalized or topic sensitive page rank [12], [13]. In addition, representing data bases as hypergraphs has been successfully used previously for ranking query results [14] and [15]. To the best of our knowledge, there are no systems for intuitively visualizing the suggested links or the provenance of those links for that matter. We expect our visualization techniques to generalize to previous approaches of link prediction. For the purpose of the demo, we focus on belief propagation. We use connection subgraphs [6] and further filter them to display provenance of predicted links.

V. Conclusion

The contributions of *VoidWiz* are as follows:

- **Principled value imputation**, using network effects
- **Visualization of provenance** of suggested values
- **Visualization and interaction** with large tabular and network data
- Application on **real word data**

More specifically, we introduce a system and intuitive interface for resolving incompleteness in relational data by transforming it into a hypergraph. We use belief propagation with the Yedidia equations (see Eq. 1, 2, 3), with which we provide the user with a ranking of probable values for the missing data. We allow the user to choose a value from the list of most probable values, and visualize his choice both in the relational table and in the graph. Such a system must provide the user with attribution of the suggested value, giving users confidence in selected values. Lastly, *VoidWiz* provides an interactive interface for visualizing both the relational data and the hypergraph, and the values suggested by the belief propagation algorithm.

[3]http://www.cs.utoronto.ca/∼christina/code/VoidWiz/

References

[1] P. C. Arocena, B. Glavic, and R. J. Miller, "Value invention in data exchange," in *SIGMOD*, 2013, pp. 157–168.

[2] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[3] S. Yeganeh, O. Hassanzadeh, and R. J. Miller, "Linking semistructred data on the web," in *Workshop on the Web and Databases*, 2011.

[4] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996, pp. 336–343.

[5] M. Volkovs, C. Fei, J. Szlichta, and R. J. Miller, "Continuous data cleaning," in *ICDE*, 2014.

[6] C. Faloutsos, K. S. McCurley, and A. Tomkins, "Fast discovery of connection subgraphs," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 118–127.

[7] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011. [Online]. Available: http://vis.stanford.edu/papers/d3

[8] L. Getoor and C. P. Diehl, "Link mining: a survey," *SIGKDD Explor. Newsl.*, vol. 7, no. 2, pp. 3–12, Dec. 2005. [Online]. Available: http://doi.acm.org/10.1145/1117454.1117456

[9] Z. Yin, M. Gupta, T. Weninger, and J. Han, "A unified framework for link recommendation using random walks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*. IEEE, 2010, pp. 152–159.

[10] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[11] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *Proceedings of the fourth ACM international conference on Web search and data mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011, pp. 635–644. [Online]. Available: http://doi.acm.org/10.1145/1935826.1935914

[12] T. H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 15, no. 4, pp. 784–796, 2003.

[13] K. Avrachenkov, N. Litvak, D. Nemirovsky, E. Smirnova, and M. Sokol, "Quick detection of top-k personalized pagerank lists," in *Algorithms and Models for the Web Graph*. Springer, 2011, pp. 50–61.

[14] F. Geerts, H. Mannila, and E. Terzi, "Relational link-based ranking," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 552–563.

[15] A. Balmin, V. Hristidis, and Y. Papakonstantinou, "Objectrank: Authority-based keyword search in databases," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 564–575.