Clustering Providers Across Disparate Healthcare Datasets using a Path-based Pseudo Similarity Measure

Sangkeun Lee, Mallikarjun Shankar
Computer Science and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
{lees4,shankarm}@ornl.gov

Byung-Hoon Park
Computer Science and Mathmatics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
parkbh@ornl.gov

ABSTRACT
Identifying correlations and relationships between entities within and across different data sets (or databases) is of great importance in many domains. The healthcare domain in particular requires bringing together diverse data sets to enable access and identity matching of beneficiaries and providers. In collaboration with the Centers for Medicare and Medicaid Services (CMS), we explored a graph-based integration of disparate data sets of medical service providers in Medicare and Medicaid programs for the purpose of matching identical providers across the programs. In this paper, we describe a path-based clustering algorithm that groups similar providers exploiting shared properties represented as linkages in the consolidated graph model. We also discuss preliminary results of the clustering algorithm.

Categories and Subject Descriptors
H.2.7 [**Database Management**]: Database Administration

General Terms
Algorithms, Design, Management

Keywords
Graph Database, Node Similarity, Healthcare, Data Integration, Matching, Clustering

1. INTRODUCTION
We live in an era of data and datastores proliferation. Both structured and unstructured data sets are growing not only in size but also in complexity at an unprecedented rate creating technical challenges in transforming them into actionable insights. One such challenge is identifying correlations and relationships between entities within and across different data sets (or databases). Entities in the healthcare domain such as beneficiaries (i.e., patients who receive medical service), providers (of medical services), and medical institutions have a host of identifying information embedded in a variety of different databases. Since the Patient Protection and Affordable Care Act (PPACA) relies on improving care delivery and quality of care in a targetted and high-quality manner, a subtle but critical need is to uniquely identify beneficiaries and providers based on partial and disparate information. When unique identifiers such as social security numbers and verified national provider identifiers (NPI) are given this task is easy. However, it is often the case that different datasources contain different identifying information, and some information is intentionally withheld or obfuscated to get around the system.

Data integration is an effort to combine data from different sources so that a unified view of the data would be provided. This unified view can enable analytics to pinpoint individuals in the system with high accuracy. Approaches that are inherently capable of disclosing associations and links between entities in a resulting consolidated data set can further enable analytics. The need to create links (i.e., edges) between entities (which we can think of as nodes) led us to choose a graph-based model to integrate and analyze the data for relationships. Unlike relational database models, which force schema alignment, the graph-based approach instead creates explicit linkages between entities defined within or across datasets [1]. We particularly selected a heterogeneous graph model [4], which allows multiple node and edge types, to integrate three provider datasets from Medicare and Medicaid programs so that providers within and across different programs are identified by their relationships represented as linkages over multiple features in the graph. The data sets included are NPPES[1] (Medicare), PECOS[2] (Medicare) and TMSIS[3] (Medicaid data from State of Texas).

Figure 1 shows a simplified view of the constructed graph structure. *Provider* nodes across or within the same data sources can be linked with other providers through feature nodes such as *Address Signature*, *Telephone Number*, *Fax Number*, *Identification Number*, and *Taxonomy* nodes. Note that redundant provider records within or across data sets are represented as separate provider nodes in the graph. Therefore, our challenge is to identify the same or similiar provider nodes by investigating their linkage structures.

In this paper, we discuss our efforts to identify similar or

[1]National Plan & Provider Enumeration System
[2]Provider Enrollment, Chain, and Ownership System
[3]Transformed Medicaid Statistical Information System

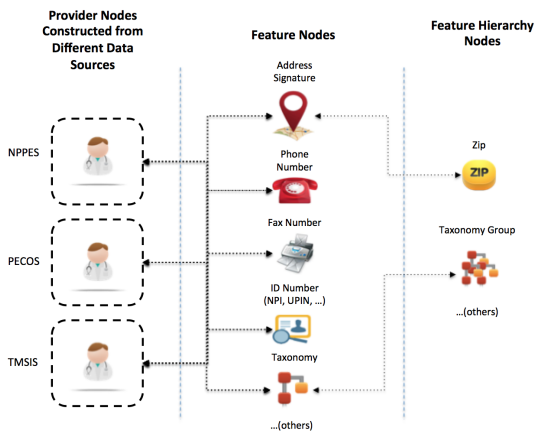

Figure 1: Simplified Overview of Constructed Graph Structure

the same providers and to create clusters which composed of highly related providers using the constructed graph database. First, we defined a pseudo-similarity measure on heterogeneous graph. Then, based on the measure, we developed top-k provider identity matching algorithm and provider clustering algorithm. The remainder of this paper is composed of four sections as follows. In Section 2, we present algorithms for provider identity matching and clustering. In Section 3, we share our analysis on the clustering results using our constructed graph and algorithms. In Section 4, we discuss use cases, future work, and then we conclude our work.

2. ALGORITHMS

Our clustering algorithm works in two steps. First, it identifies top-k similar nodes for every node. Next, it groups a set of nodes that share any of their top-k similar nodes validated by a heuristic rule into the same cluster. Thus, we need a proper similarity measure between nodes. While similarity or relevance measures are relatively well estabilished for the homogeneous graph [2, 3], measures on the heterogeneous graph are not well studied. There exist some measures [4–6], but all require users to choose specific *meta paths* to compute similiarities, where a *meta path* is a sequence of node and edge types in the graph [6]. Consequently, the performance is largely dependent on user's particular selection on meta paths.

In this work, we define a simple pseudo-similarity measure named *SP-Sim*, which does not depend on any single meta path, based on the following observations: (1) The shorter the paths between two nodes are, the more they are similar. (2) The more paths exist between two nodes the more they are similar. (3) The fewer paths between two nodes are shared by other nodes the more they are similar.

Let $n_1 \xrightarrow{e_1} n_2 \ldots \xrightarrow{e_m} n_{m+1}$ denote a path p between nodes n_1 and n_{m+1}, where e_i is the edge coming out of n_i. Also let $d(e_i)$ denote the number of edges that are coming out of n_i and of the type $e_i.type$[4], i.e. the same type as e_i. Then let us define $pathScore(p)$ as $\Pi_{i=1\ldots m} \frac{1}{d(e_i)}$ and $P\text{-}Sim(n_a, n_b, l)$ as $\sum_{p \in P_{(n_a,n_b,l)}} pathScore(p)$, where $P_{(n_a,n_b,l)}$ is a set of all paths between node n_a and node n_b, whose length are less than equal to l. According to the first observation described above, we define $SP\text{-}Sim(n_a, n_b)$ as $P\text{-}Sim(n_a, n_b, l)$ with l = $length$ of shortestPath(n_a,n_b). Note that the *P-Sim* and *SP-Sim* are pseudo similarity measures as similarity scores between the same nodes are not defined. Figure 2 shows several examples of *SP-Sim* computations. All paths depicted in the figure are shortest paths between two nodes, and the number denoted above each edge e_i represents $d(e_i)$. In the case of (a), the *SP-Sim* score is larger than that of the case (b) because an additional shortest path exists between the two nodes. Similarly, the score for the case (c) is larger than that of the case (d). However, note that the score for the case (b) is smaller than that of the case (d) even both cases have the same number of paths of the same length because $d(e_i)$ values are larger in the case (d). Note also that the *SP-Sim* score can be larger with the cases with longer shortest paths as found in the case (e) as opposed to the cases (a) and (b).

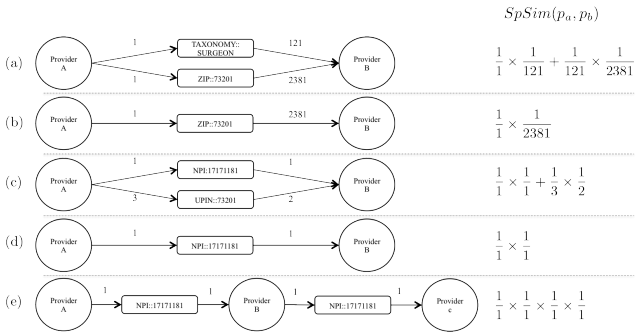

Figure 2: Examples of *SP-Sim* computations

Computing top-k similar provider nodes for every provider node inherently requires $|N| \times |N|$ comparisons, where N is the number of provider nodes. It becomes very costly when performing it with large number of providers. To reduce the computation time, we use a heuristic approach to approximte top-k provider nodes for each node by using the fact that distances between n_i and n_j tend to be shorter when *SP-Sim*(n_i, n_j) is larger. When an edge type t is given, we can get a set of nodes N_t= {n_1, n_2, ...} that are all start nodes of edges whose type is t. First, we define a function $d_T(t)$ which returns the average number of t type edges in N_t. Then, we define another function $d_P(\mathcal{P})$=$\Pi_{i=1\ldots m} d_T(e_i.type)$, where $\mathcal{P}$ is a meta path $n_1.type \xrightarrow{e_1.type} n_2.type \ldots \xrightarrow{e_m.type} n.type_{m+1}$. Using the defined function, we generate list of top-k similar provider nodes for a given provider node n_q as explained in Algorithm 1.

When a provider node n_q is given, the *topK* list includes all providers that have strong relationships with the given node. For example, provider nodes with the same NPIs, phone numbers, or the other properties such as billing addresses are included in the *topK* list. Thus, nodes representing the same provider as the query node tend to be included in the *topK* list. However, since highly ranked nodes are not always the same providers as the query provider node, we additionally

[4]A type of node n or edge e is denoted by $n.type$ and $e.type$ respectively

input : Graph G=(V,E), Provider query node n_q, Integer k
output: A list of provider nodes with their scores
topK=[(n_1,s_1),(n_2,s_2),...,(n_k,s_k)]

begin
  /*getTopK(G, n_q, k)*/
  $topK$ = []; len = 1;
  S_P is a set of all possible schema paths in the graph G;
  while |$topK$| ≠ k && S_P!=ϕ **do**
    $\mathcal{P}$ ← pick a length len meta path $\mathcal{P}$ that has the smallest $d_P(\mathcal{P})$ from S_P;
    Remove $\mathcal{P}$ from S_P;
    foreach *provider node n in a path instance p of* $\mathcal{P}$ **do**
      s = *SP-Sim*(n_q,n);
      if s>*minScore in topK* **then**
        Add (n_q,s) to $topK$;
      end
    end
    if *there is no length len meta path in* S_P **then**
      len++;
    end
  end
  return $topK$;
end

Algorithm 1: Finding Top-k Similar Providers

used pre-defined rules to confirm the same providers in the computed $topK$ list. Algorithm 2 is a path-based clustering algorithm based on top-k *SP-Sim*. Note that the function $merge\_clusters(L)$ returns a set of clusters by merging all clusters in L, which share one or more than one element.

input : Graph G=(V,E), Integer k
output: A set of clusters L

begin
  L_{pre}={};
  $clustered$={}
  foreach *provider node* $n \in V$ && $n \notin clustered$ **do**
    $topK$ = getTopK(G, n_p, k);
    C_n={};
    Add n to C_n;
    Add n to $clustered$;
    foreach *provider node* n_i *in topK* **do**
      if *confirmRule*(n, n_i)== *True* **then**
        Add n_i to C_n;
        Add n_i to $clustered$;
      end
    end
    Add C_n to L_{pre};
  end
  L = merge_clusters(L_{pre});
  return L;
end

Algorithm 2: Path-based Clustring

3. ANALYSIS ON GENERATED CLUSTERS

In this section, we discuss preliminary clustering results. The graph constructed from the three data sets has 2,969,198 nodes and 6,648,770 edges, from which 843,018 are provider nodes. *Neo4J* graph database are used to construct the graph. For getTop$k$ algorithm, we set the parmeter k to 5. For confirmRule(n_a, n_b), which tests whether nodes n_a and n_b should belong to the same cluster, we compare the names of the two providers. More specifically, if the two providers share the same first and last names, we conclude that the two belong to the same cluster. Note that the name of a provider is not included in our graph model, for names are often shared among non-related providers, thus confuse the similarity metric.

Table 1: Statistics of Clusters

# of Provider Nodes	843,018
# of Clusters	422,751
Average Size of Clusters	2.196
Clusters with size≤ 5	392,397 (92.820%)
Clusters with size≥ 2	170,856 (40.416%)
Max Cluster Size	141
Min Cluster Size	1

Table 2: Data Sources of Providers in Clusters

	S_1 (#)	S_2 (#)	S_3 (#)	AVG (%)
TNP	2,596	2,561	2,563	12.227
TP	240	255	216	1.159
TN	2,350	2,429	2,458	11.243
NP	820	825	854	3.887
TOTAL	6,006	6,070	6,091	28.520
N	6,445	6,475	6,360	30.539
P	946	978	985	4.526
T	7,740	7,614	7,701	36.420
TOTAL	15,131	15,067	15,046	71.480

Table 1 shows the statistics of the generated clusters. The number of clusters generated is slightly less than the half the number of all providers (40.4%, 422,751/843,018). The average cluster size is 2.196, which indicates roughly more than 40% of providers potentially have identical or similar providers. 92.820% (392,297) clusters are of size 5 or less. The largest cluster size is 141.

As mentioned earlier, data sources integrated into the graph are three datasets, NPPES (Medicare), PECOS (Medicare) and TMSIS (Medicaid). A cluster may contain provider nodes from a single or multiple sources. To analyze distributions of different compositions, we sampled three 5% (21,137) clusters and analysed the proportions. For this We defined seven cluster types - TNP, TP, N, TN, P, T, NP, where T, N, and P stands for TMSIS, NPPES, and PECOS respectively. With these letters, a homogeneous or heterogeneous cluster types are expressed. For example, the cluster type TP includes providers from TMSIS and PECOS, whereas P includes providers from PECOS only. Table 2 shows detailed the result. About 28.52% of clusters are found to include providers from multiple sources, thus heterogeneous. These heterogeneous clusters can be used to correlate providers across different data sources. Among heterogeneous clusters cluster type TNP was most prevelant, and the next was TN.

4. CASE STUDY

In this section, we illustrate how the generated clusters can be used to enrich information of providers that are brought in from an external source. List of Excluded Individuals/Entities (LEIE) [5] is a complete database containing all exclusions of providers participated in Medicare in effect. Since it could unveil identical or similar providers in other program (Medicaid, for example), to match providers listed in LEIE against the clusters is of strategical importance. For this reason, We

[5]http://oig.hhs.gov/exclusions/exclusions_list.asp

selected the Texas portion of LEIE data dated June, 2013, which includes 3,679 excluded providers from Medicare, and analyzed their matches in the graph and the corresponding clusters.

The analysis was conducted in the following two steps. First we search LEIE providers against the provider graph using whichever features available with them. Features (or types of information) available for a provider in LEIE includes name, address, and UPIN or NPI ID. However, in many cases, not all of the information are available with LEIE data. Once the provider nodes in the graph are found to match a LEIE provider, the clusters to which the nodes belong are investigated to identify possibly the same providers.

186 LEIE providers are found to match 630 provider nodes in the graph. Among 630 provider nodes, 104 nodes (16.5%) were from NPPES, 7 (1.1%) from PECOS, and 519 (82.3%) were from TMSIS. The number of nodes that were identified from the generated clusters was 282 (44.76%, 282/630). Detailed use case analysis shows that the clusters can help acquire additional information missing in the original LEIE database, which we list below.

- **Finding identical providers across different data sources by using clusters:** For example, entry 1944 in the LEIE has its identifier UPIN:'B2XXXX' [6]. At first, only one node (2334924 /NPPES)[7] is initially found by matching the UPIN. Then, additional nodes (2466171 /TMSIS), (2466172 /TMSIS) can be identified from the cluster we generated. In the LEIE, there is no NPI information, but we can get from node 2466172 whose NPI is 'NPI:198XXXXXXX'
- **Finding additional NPI(s) of a provider in LEIE that we cannot directly achieve from the LEIE dataset:** For example, entry 1 in the LEIE has its NPI:'198XXXXXXX'. Two nodes (2335473 /NPPES), (2434345 /TMSIS) are initially found by NPI matching. An additional node (2277585 /NPPES) can be identified from their cluster. From the node (2277585), we can get an NPI:'172XXXXXXX' that was not initially included in the LEIE.
- **Finding MEDICAID_ID, LICENSE_ID, SSN of a provider in LEIE:** Initially, LEIE does not include MEDICAID_ID, LICENSE_ID, and SSN. For example, entry 41 has its identifier NPI:'147XXXXXXX'. One node (2221112 /NPPES) is found. Five additional nodes (2660875, 2660876, 26608 77, 2660878, 2660879) can be identified from the its cluster. From the node 2660877, LICENSE_ID (15XXX TX), MEDICAID_ID (126XXXXXX), SSN (4673XXXX) can be identified.
- **Finding the same provider nodes with different names or addresses:** For example, we find nodes with the same names/zipcodes (or city names). One node (2600996/TMSIS) is initially found. Then, additional nodes (2182297/NPPES), (2600997/TMSIS) can be identified from its cluster. Although Provider node 2182297 and 2600996 have different names (partially matched), but they share the same NPIs, LICENSE_IDs, they are clustered into the same cluster.

[6]All identification numbers or sensitive information are anonymized to protect personal information

[7]We denote by (*node_id* / [NPPES, PECOS, TMSIS]) a provider node whose node ID is *node_id* and original data source is either NPPES, PECOS, or TMSIS

5. CONCLUSIONS

We proposed a linked-based clustering algorithm which is based on a psuedo-similarity measure defined on heterogeneous graphs. Also, we showed how we can exploit the proposed algorithm for identifying nodes representing the same providers from different health care programs. To evaluate the usefulness of our approach, we used algorithm to indetify provider nodes included in the LEIE. The results suggest that new and rich information can be obtained improving traditional attribute matching; other applications include:

Data Cleansing: By identifying redundant nodes, we can re-construct a cleaned graph without duplicated nodes representing the same entities.
Advanced Provider Search System: Practitioners may need to find all information about a provider such as identification codes, addresses, telephone numbers, and so on. Based on our proposed top-k similarity search, we can build a keyword-based search system which does not require writing any SQL queries.
Advanced Surveillance System: In many cases, excluded providers tend to have stronger relationships between them. Our algorithms can be used to generate lists of highly related providers with the providers that are excluded from healthcare programs due to their illegal activities. Such lists would prevent future illegal activities or detect them earlier.

6. ACKNOWLEDGMENTS

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The analysis was performed under the auspicies and sponsorship of an inter-agency agreement between the Department of Health and Human Services and the Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

7. REFERENCES

[1] R. Angles and C. Gutierrez. Survey of graph database models. *ACM Computing Surveys (CSUR)*, 40(1):1, 2008.
[2] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.
[3] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279. ACM, 2003.
[4] S. Lee, S. Park, M. Kahng, and S.-g. Lee. Pathrank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. *Expert Systems with Applications*, 40(2):684–697, 2013.
[5] C. Shi, X. Kong, Y. Huang, P. S. Yu, and B. Wu. Hetesim: A general framework for relevance measure in heterogeneous networks. *arXiv preprint arXiv:1309.7393*, 2013.
[6] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB'11*, 2011.