2011 11th IEEE International Conference on Data Mining

A Hypergraph-based Method for Discovering Semantically Associated Itemsets

Haishan Liu*, Paea LePendu†, Ruoming Jin‡ and Dejing Dou*
* *Department of Computer and Information Science*
University of Oregon, Eugene, OR, 97403, USA
Email: ahoyleo@cs.uoregon.edu & dou@cs.uoregon.edu
† *Stanford Center for Biomedical Informatics Research*
Stanford University, Stanford, CA, 94305, USA
Email: plependu@stanford.edu
‡ *Computer Science Department*
Kent State University, Kent, OH, 44242, USA
Email: jin@cs.kent.edu

***Abstract*—In this paper, we address an interesting data mining problem of finding semantically associated itemsets, i.e., items connected via indirect links. We propose a novel method for discovering semantically associated itemsets based on a hypergraph representation of the database. We describe two similarity measures to compute the strength of associations between items. Specifically, we introduce the average commute time similarity, $\mathbf{s}_{\mathrm{CT}}$, based on the random walk model on hypergraph, and the inner-product similarity, $\mathbf{s}_{\mathrm{L}+}$, based on the Moore-Penrose pseudoinverse of the hypergraph Laplacian matrix. Given semantically associated 2-itemsets generated by these measures, we design a hypergraph expansion method with two search strategies, namely, the clique and connected component search, to generate k-itemsets ($k > 2$). We show the proposed method is indeed capable of capturing semantically associated itemsets through experiments performed on three datasets ranging from low to high dimensionality. The semantically associated itemsets discovered in our experiment is promising to provide valuable insights on interrelationship between medical concepts and other domain specific concepts.**

***Keywords*-Semantically associated itemset, hypergraph, random walk**

I. Introduction

There has been a surge of interest in tackling the problem of mining linked collection of interrelated objects. A single transaction table can be characterized as a set of items linked by the co-occurrence relationship. The problem of frequent itemset mining can be then formalized as to identify sets of items that often occur together, or, in other words, items that heavily connected by co-occurrence links. An itemset is deemed frequent if its *support*, i.e., the percentage of transactions which contain that itemset, is above a threshold. The support of an itemset can be viewed as a measure of endorsement among items to each other in the set. However, it only takes into account the number of direct links between items while ignoring the number of indirect links (going through intermediaries) between items. For example, given a relation table of annotated medical publications in a "bag-of-word" representation where each row corresponds to one publication and each column a boolean variable indicating if some term is appeared. Using traditional frequent itemset generation algorithms, if the number of times the item pair *fish oil* and *Raynaud's syndrome* occurring together falls below some threshold, then ⟨*fish oil*, *Raynaud's syndrome*⟩ would not be picked up as a frequent itemset. However *fish oil* and *Raynaud's syndrome* may actually be meaningfully related and the latent association can be revealed through indirect links. For instance, if ⟨*blood changes*, *fish oil*⟩ and ⟨*blood changes*, *Raynaud's syndrome*⟩ are both frequent, the presence of *blood changes* provides a connection through which *fish oil* and *Raynaud's syndrome* can be related (this relationship is discovered in Swanson's land mark paper published in 1987 [1] before the age of Web-scale computation). We call the intermediary item such as *blood changes* in this case a *linking item*, and the latent association between items through the connection via one or more linking items the *semantically associated relationship*. We present in this paper an algorithm to find semantically associated itemsets.

An object set endowed with pairwise relationships can be naturally illustrated as a graph in which vertices represent objects, and any two vertices that have some kind of relationship are joined together by an edge. In the case of frequent itemset mining, a set of objects with the co-occurrence relationship can be represented as directed or undirected graphs. For illustrating this point of view, let us consider a relational table depicted in Figure 1(a). One can construct an undirected graph where the set of vertices is the set of relational attributes (column items) and an edge joins two vertices if the they co-occur in a tuple (as illustrated in Figure 1(b)). This graph is called *Gaifman graph* [2] of a relational structure. The undirected graph can be further enriched by assigning to each edge a weight equal to the support of the 2-itemset consisting of vertices incident to the edge. Cliques (complete subgraphs) in the Gaifman graph, or *Gaifman cliques* for short, are of particular interest because every tuple (ground atom) in data corresponds to a Gaifman clique. However, ambiguity arises as not all Gaifman cliques

1550-4786/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDM.2011.12

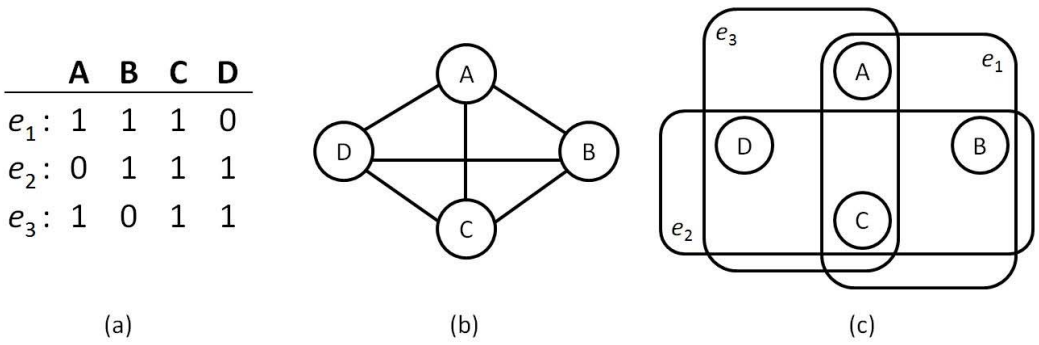

Figure 1. (a) an example transaction table; (b) the Gaifman graph representation of the table; (c) The hypergraph representation of the table

have matching tuple in the data. There exists cases where cliques are incidental in the sense that several relational ground atoms play together to induce a clique configuration in the Gaifman graph, but no ground atom covers the entire clique (e.g., the clique of $\{A, B, C, D\}$ in Figure 1(b) does not correspond to any tuple in the relational table).

A natural way to remedy the ambiguity is to represent the relational data as a *hypergraph* [3]. A hypergraph is a generalization of traditional graph. An edge in the hypergraph, called hyperedge, can connect more than two vertices. In other words, every hyperedge is an arbitrary nonempty subset of vertices. It is obvious that a simple graph is a special kind of hypergraph with each hyperedge containing only two vertices. In this paper, we propose to employ hypergraphs to model relational structure for finding semantically associated itemsets. Specifically, we propose to construct a hyperedge for each tuple. The relational attributes constitute the universe of vertices in the hypergraph. In this representation, each hyperedge has an exact one-to-one correspondent tuple (see Figure 1(c), for example).

With the hypergraph model, finding semantically associated items amounts to developing a meaningful similarity measure between the items which takes into account the effect of linking items. Such similarity measure should satisfy the intuition that the more "short" connections between two given items (through linking items), the more similar those items are. To this end, we propose to employ the following quantities as the candidate similarity measure since both of them have the desired property. They are, namely, the *commute time distance* based similarity measure from the random walk model on hypergraph, and the inner product similarity based on the *pseudoinverse of the hypergraph Laplacian*. If the similarity of a pair of items measured by these quantities exceeds some threshold, then this pair of items can be deemed as a semantically associated 2-itemset. Given 2-itemsets, we propose a hypergraph expansion methods based on pruning of its primal graph together with two search strategies in the resulting graph to discover semantically associated k-itemsets ($k > 2$).

The rest of this paper is organized as follows. We introduce the basics of hypergraph and random walk model in Section II. We review related work in Section III. We present our method for discovering semantically associated itemsets based on hypergraphs in Section IV. We report experimental results in Section V and conclude the paper in Section VI.

II. Preliminaries and Background

A. Hypergraph

A hypergraph [3] is a generalization of a traditional graph where edges, called hyperedges, can connect any number of vertices. In other words, hyperedges can be viewed as non-empty subsets of the vertices. Formally, a hypergraph $G = (V, E)$, is a pair in which V is the vertex set and E is the hyperedge set where each $e \in E$ is a subset of V. A weighted hypergraph is a hypergraph that has a positive number $w(e)$ associated with each hyperedge e; called the weight of hyperedge e: Denote a weighted hypergraph by $G = (V, E, w)$. The degree of a vertex $v \in V$, $d(v)$, is defined as

$$d(v) = \sum_{v \in V, e \in E} w(e),$$

The degree of a hyperedge e, denoted as $\delta(e)$, is the number of vertices in e, i.e. $\delta(e) = |e|$. A hyperedge e is said to be incident with a vertex v when $v \in e$. The hypergraph incidence matrix $\mathbf{H} \in \mathbb{R}^{|V| \times |E|}$ is defined as

$$h(v, e) = \begin{cases} 1, & v \in e \\ 0, & otherwise \end{cases} \quad (1)$$

Throughout the rest of the paper, the diagonal matrix forms for $\delta(e)$, $w(e)$, $d(v)$ are denoted as $\mathbf{D}_e$, $\mathbf{W} \in \mathbb{R}^{|E|}$, and $\mathbf{D}_v \in \mathbb{Z}^{|V|}$, respectively.

B. Random Walk

1) Random Walk on Simple Graph: Given a graph and a starting point we select a neighbor of it at random and move to this neighbor then we select a neighbor of this

point at random and move to it etc. The random sequence of points selected this way is a random walk on the graph. In other words, a random walker can jump from vertex to vertex and each vertex therefore represents a state of the Markov chain. The average first-passage time $m(k|i)$ [4] is the average number of steps needed by a random walker for reaching state k for the first time, when starting from state i. The symmetrized quantity $n(i,j) = m(j|i) + m(i|j)$ called the average commute time [4], provides a distance measure between any pair of states. The fact that this quantity is indeed a distance on a graph was proved independently by Klein and Randic [5] and Gobel and Jagers [6].

The Laplacian matrix $\mathbf{L}$ of a graph is widely used for finding many properties of the graphs in spectral graph theory. Given node degree matrix $\mathbf{D}$ and graph adjacency matrix $\mathbf{A}$, the Laplacian matrix of the graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The normalized Laplacian is given by $\mathbf{L}_N = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{I}$ is the identity matrix. The average commute time $n(i,j)$ can be computed in closed form from the Moore-Penrose pseudoinverse [7] of $\mathbf{L}$, denoted by $\mathbf{L}^+$ as shown in Section IV.

2) Random Walk on Hypergraph: We can associate each hypergraph with a natural random walk which has the transition rule as described in [8]. Given the current position $u \in V$, first choose a hyperedge e over all hyperedges incident with u with the probability proportional to $w(e)$, and then choose a vertex $v \in e$ uniformly at random. Obviously, it generalizes the natural random walk defined on simple graphs. Let $\mathbf{P}$ denote the transition probability matrix of this hypergraph random walk. Then each entry of $\mathbf{P}$ is

$$p(u,v) = \sum_{e \in E} w(e) \frac{h(u,e)}{d(u)} \frac{h(v,e)}{\delta(e)} \ .$$

In matrix notation, $\mathbf{P} = \mathbf{D}_v^{-1}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T$.

Zhou et al. [8] define the following normalized hypergraph Laplacian $\mathcal{L}$ based on the random walk model:

$$\mathcal{L} = \mathbf{I} - \mathbf{\Theta}, \tag{2}$$

where

$$\mathbf{\Theta} = \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}} \ .$$

III. Related Work

A. Semantic Association and Connection Subgraph

Sheth et al. [9] proposed a formalism for *semantic association* between entities in an RDF graph. Specifically, the semantic association is defined based on semantic connectivity which indicates if there exists a sequence of interconnected links between two given entities. In our study of semantically associated itemsets in transaction data, the link between entities is essentially the 'co-occurrence' relationship. The semantic association according to Sheth et al's definition between transaction items i_0 and i_n can be established by identifying a link of the form $i_0, P_c, i_1, P_c, \ldots, i_{n-1}, P_c, i_n$, in which P_c denotes the co-occurrence property. The random walk model on hypergraph described in Section II-B2 formalizes this point of view. Our method for measuring the strength of semantic association is hence based on constructing a hypergraph representation and studying its property with both graph theoretical and spectral analysis techniques.

Faloutsos et al. [10] defined a *connection subgraph* as a small subgraph of a large graph (such as social networks graphs) that best captures the relationship between two nodes. Since finding all paths connecting two nodes is impractical and finding a single most "critical" path under some criteria is unfair, the connection subgraph addresses these problems by extracting a subset of all paths between two nodes that contains only the most significant ones to characterize their relationship under certain constraints. The paths in the connection subgraphs are essentially "important" semantic associations. Faloutsos et al. described an efficient algorithm to produce approximate, but high-quality connection subgraphs in real time on very large graphs. In comparison, our proposed method in the present paper does not explicitly keep track of paths (in hypergraphs). Instead, paths are implicitly utilized in the random walk model to measure the similarity between nodes and our algorithms to calculate the similarity are exact since the graphs that we deal with are much smaller (approx. 10k nodes) than Faloutsos et al's (approx 15m nodes).

B. Indirect Associations

Tan et al. [11] introduced the concept of *indirect association*: Consider a pair of items (a,b) with low support value. If there is an itemset Y (called the mediator set) such that the presence of a and b are highly dependent on items in Y, then (a,b) are said to be indirectly associated via Y. This definition of indirect association bears a similar motivation to the semantically associated itemset proposed in the present paper, as both problems aim at identifying itemsets that do not have sufficiently high support but are likely to provide useful insight into the data. The importance of indirect association has also been recognized by several other authors [12][13]. Tan et al. were the first to propose an algorithm to derive the indirect associations by iteratively finding mediator set for candidate itemsets. Later Wan et al. [14] proposed a more efficient algorithm, called HI-Mine, which improved the performance by avoiding the standard frequent itemset generation process.

The difference between the indirect and semantic association is that items in the indirect association are connected via a mediator set while items in the semantic association are connected via a path. It is worth noting that in our graph-based formalism for finding semantically associated itemsets, the paths connecting two items are not explicitly required to be identified, while in both Tan et al's and Wan

et al's algorithms the mediator set Y has to be identified along the process of discovering indirect associations.

C. Random Walk-based Similarity Analysis

Various quantities derived from random walk on graph has been used in a number of applications. Fouss et al. [15] comprehensively compared twelve scoring algorithms based on graph representation of the database to perform collaborative movie recommendation. Pan et al. [16] developed a similarity measure based on random walk steady state probability to discover correlation between multimedia objects containing data of various modalities. Yen et al. [17] introduced a new k-means clustering algorithm utilizing the random walk average commute time distance. Zhou et al. [18] presented a unified framework based on neighborhood random walk to integrate structural and attribute similarities for graph clustering.

Palmer et al. [19] exploited a rich duality between random walks on graphs and electrical circuits to develop an external similarity function called REP to measure the similarity between categorical attributes. In their work, they identified a subtle flaw of commute time distance where it degenerates on realistic data when the degree distribution follows a Zipf or power-low relationship. In other words, distances are skewed toward the high degree nodes. Our experiment results based on commute time similarity confirmed this finding (see Table IV for illustration of this phenomena and discussion in Section V). We also discover that the inner-product based similarity does not suffer from this flaw.

IV. Method

In this section, we present our method for discovering semantically associated itemsets based on hypergraph. Our method starts by generating 2-itemsets. A 2-itemset $\langle i, j\rangle$ is considered semantically associated if the hypergraph-based similarity measure $s(i, j)$ exceeds some threshold. In the following subsections, we propose two similarity measures s_{CT} and s_{L+} based on, respectively, the average commute time distance on hypergraph and the inner-product-based representation of the pseudoinverse of Hypergraph Laplacian. Given discovered semantically associated 2-itemsets, we propose a hypergraph expansion method along with two search strategies, namely, the clique and connected component search, in the resulting graph for finding semantically associated k-itemsets ($k > 2$).

A. Methods for Generating 2-itemsets

In the following we describe two similarity measures that define the strength of bond between a pair of semantically associated items.

1) Average Commute Time Similarity s_{CT}: As already mentioned, the commute-time distance $n(i, j)$ between two nodes i and j has the desirable property of decreasing when the number of paths connecting the two nodes increases and when the length of paths decreases. This is indeed an intuitively satisfying property of the effective resistance of the equivalent electrical network [20]. The usual shortest-path distance (also called geodesic distance) does not have this property: the shortest-path distance does not capture the fact that strongly connected nodes are closer than weakly connected nodes.

To compute commute-time distance between vertices in a hypergraph, we need to first define the combinatory hypergraph Laplacian $\mathbf{L}$. It follows from Zhou et al's formalism of normalized hypergraph Laplacian in Equation 2 that:

$$\mathbf{L} = \mathbf{D}^{1/2}\mathcal{L}\mathbf{D}^{1/2} = \mathbf{D}_v - \mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T \tag{3}$$

The average commute time $n(i, j)$ on simple graph can be computed in closed form from the Moore-Penrose pseudoinverse of $\mathbf{L}$ [7], denoted by $\mathbf{L}^+$ with elements $l_{ij}^+ = [\mathbf{L}^+]_{ij}$. It can be shown that $n(i, j)$ on hypergraph can be calculated in the same manner. The pseudoinverse $\mathbf{L}^+$ is given by the following equation:

$$\mathbf{L}^+ = (\mathbf{L} - \mathbf{e}\mathbf{e}^T/n)^{-1} + \mathbf{e}\mathbf{e}^T/n, \tag{4}$$

where $\mathbf{e}$ is a column vector made of 1s (i.e., $\mathbf{e} = [1, 1, \ldots, 1]^T$). The formula for the computation of $n(i, j)$ takes the form of the following equation:

$$n(i, j) = V_G(l_{ii}^+ + l_{jj}^+ - 2l_{ij}^+), \tag{5}$$

where $V_G = tr(\mathbf{D}_v)$ is the volume of the hypergraph. If we define $\mathbf{e}_i$ as the ith column of $\mathbf{I}$ (i.e., $\mathbf{e}_i = [\underset{1}{0}, \ldots, \underset{i-1}{0}, \underset{i}{1}, \underset{i+1}{0}, \ldots, \underset{n}{0}]^T$), Equation 5 can be transformed to:

$$n(i, j) = V_G(\mathbf{e}_i - \mathbf{e}_j)^T\mathbf{L}^+(\mathbf{e}_i - \mathbf{e}_j), \tag{6}$$

Since $n(i, j)$ is a distance, it is straightforward to convert it to a similarity measure $s_{CT}(i, j)$ by normalize it to unit range and subtract from 1.

2) Pseudoinverse-based Inner-Product Similarity s_{L+}: Equation 6 can be mapped into a new Euclidean space that preserves the commute time distance:

$$\begin{aligned} n(i, j) &= V_G(\mathbf{e}_i - \mathbf{e}_j)^T\mathbf{L}^+(\mathbf{e}_i - \mathbf{e}_j) \\ &= V_G(\mathbf{x}'_i - \mathbf{x}'_j)^T(\mathbf{x}'_i - \mathbf{x}'_j) \\ &= V_G\|\mathbf{x}'_i - \mathbf{x}'_j\|^2, \end{aligned} \tag{7}$$

where $\mathbf{x}'_i = \mathbf{\Lambda}^{1/2}\mathbf{U}^T\mathbf{e}_i$, $\mathbf{U}$ is an orthonormal matrix made of eigenvectors of $\mathbf{L}^+$ (ordered in decreasing order of corresponding eigenvalue λ_k) and $\mathbf{\Lambda} = \mathbf{Diag}(\lambda_k)$. In this way, the transformed node vectors $\mathbf{x}'_i$ are exactly separated in the new n-dimensional Euclidean space. From this definition, it follows that $\mathbf{L}^+$ is the matrix containing inner products of the transformed vectors $\mathbf{x}'_i$ as shown below:

$$\begin{aligned} \mathbf{x}_i'^T\mathbf{x}'_j &= (\mathbf{\Lambda}_i^{1/2}\mathbf{x}_i)^T\mathbf{\Lambda}_j^{1/2}\mathbf{x}_j = \mathbf{x}_i^T\mathbf{\Lambda}\mathbf{x}_j \\ &= \mathbf{e}_i^T\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{e}_j = \mathbf{e}_i^T\mathbf{L}^+\mathbf{e}_j = l_{ij}^+. \end{aligned} \tag{8}$$

Therefore, $\mathbf{L}^+$ can be considered as a similarity matrix for the nodes—that is

$$s_{L^+}(i,j) = l_{ij}^+ . \tag{9}$$

The inner-product-based similarity measures are well-studied for the vector-space model of information retrieval. It has been shown that when computing proximities between documents, inner-product-based measures outperform Euclidean distances [21].

B. Methods for Generating k-itemset ($k > 2$)

Now, we consider finding semantically associated k-itemset ($k > 2$) from given 2-itemsets. As is common in hypergraph theory, we can associate an induced graph $G(H)$ with every hypergraph H by expanding every hyperedge e in H to a clique in $G(H)$. Edges in the induced graph $G(H)$ can be called *subedges* to avoid unnecessary confusion. We can further construct a pruned graph $G'(H)$ from $G(H)$ by applying the following inclusion rule on each subedge: the similarity between the incident nodes of a subedge has to be greater than a user-specified threshold θ. In formal definition, given a hypergraph $H = (V, E)$, the pruned subgraph is $G'(H) = \{V, E'\}$ where

$$\begin{aligned} E' = \{(u,v) \in V^2 \,:\, & u \neq v \text{ and} \\ & u, v \in e \text{ for some } e \in E \text{ and} \\ & s(u,v) > \theta\}. \end{aligned}$$

Given $G'(H)$, finding semantically associated k-itemset ($k > 2$) can be formulated into two ways: finding cliques or connected components in $G'(H)$.

1) Cliques of $G'(H)$: Finding cliques in $G'(H)$ corresponds to searching and testing in the powerset of V. Given the fact that every subset of a clique is also a clique, this downward-closure property can make efficient clique discovery algorithm possible in a way similar to the Apriori algorithm for finding frequent itemsets — with a "bottom up" manner, the candidate generation step extends valid $k-1$ length itemsets one item at a time, and groups of candidates are tested against $G'(H)$ to determine if they form cliques. The algorithm terminates when no further successful extensions are found.

2) Connected Components of $G'(H)$: Complete subgraph (i.e., clique) is a very strong requirement that can limit the approach to restricted cases of semantically associated itemsets. One way to relax this requirement is to find connected components of $G'(H)$, which can be viewed as a closure under semantic association. The number of connected components equals the multiplicity of 0 as an eigenvalue of the Laplacian matrix of $G'(H)$. Although the set of connected components is not downward closed, there is efficient way to find all connected components of a graph in linear time using either breadth-first search or depth-first search. In either case, a search that begins at some particular vertex will find the entire connected component containing the vertex. When the search returns, loop through other vertices and start a new search whenever the loop reaches a vertex that has not already been included in a previously found connected component.

3) Ranking of Itemsets: Once the semantically associated 2-itemsets and k-itemsets are generated, they can be ranked by a quantity indicating the strength of association among items in the set. We tentatively compute this quantity by averaging the total pairwise similarities over the number of subedges of the itemset's corresponding clique or connected component in $G'(H)$.

C. Effective Computation

In high dimensional data sets, the computations of the Hypergraph Laplacian and the pseudoinverse become intractable. We discuss two approaches to mitigate this scalability problem.

To compute Hypergraph Laplacian $\mathbf{L}$ in Equation 3 requires multiplication of hypergraph incidence matrices $\mathbf{H}$ and its transpose $\mathbf{H}^T$. Since $\mathbf{H}$ grows in proportion to the size of underlying transaction data (each node corresponds to a column and each hyperedge corresponds to a row), it eventually becomes unable to fit in memory when the size exceeds a certain amount. In this case the computation can still be carried out using a block partitioned matrix product by performing operations only on the submatrices of tractable sizes. Owing to the fact that, in most cases, $|V|$ is much smaller than $|E|$, $\mathbf{H}$ can then be partitioned into s vertical stripes and the square matrix $\mathbf{D}_e$ into s diagonal blocks. The multiplication in Equation 3 can be calculated by $\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^T = \sum_{\gamma=1}^{s} \mathbf{H}_\gamma \mathbf{D}_{e\gamma}^{-1} \mathbf{H}_\gamma^T$. Note that $\mathbf{H}$ is sparse in many applications. This property can be exploited to gain high performance and due to its importance much effort has been devoted to the study resulting a number of libraries and routines from which we can leverage.

As the number of nodes grows, to compute pseudoinverse in closed form using Equation 4 also becomes intractable. A procedure based on Cholesky factorization to compute $\mathbf{L}^+$ for large sparse matrices [22] is proved useful. It allows to compute $\mathbf{L}^+$ in a column-by-column manner. In particular, the procedure involves the following steps for computing the ith column of $\mathbf{L}^+$:

1) Compute the projection $\mathbf{y}_i$ of base vector $\mathbf{e}_i$ on the column space of $\mathbf{L}$.
2) Find a solution l_i^{*+} of the linear system $\mathbf{L}\mathbf{l} = \mathbf{y}_i$.
3) Project l_i^{*+} on the row space of $\mathbf{L}$ to get l_i^+.

Since $\mathbf{L}$ is symmetric, its row space is the same as column space. The projection in step 1 and 2 can be represented by the matrix $(\mathbf{I} - \mathbf{e}\mathbf{e}^T/n)$. The equation in step 2 can be solved by first solving a reduced linear system: $\hat{\mathbf{L}}\hat{\mathbf{l}} = \hat{\mathbf{y}}_i$, where $\hat{\mathbf{L}}$, $\hat{\mathbf{l}}$, and $\hat{\mathbf{y}}$ are obtained respectively by removing the last row from $\mathbf{l}$, $\mathbf{y}$, and last row and column from $\mathbf{L}$. We observe that $\hat{\mathbf{L}}$ is full rank and positive definite

and hence is able to be decomposed using the Cholesky factorization, $\hat{\mathbf{L}} = \mathbf{R}\mathbf{R}^T$. Since $\mathbf{R}$ is lower-triangular, one solution of $\hat{\mathbf{L}}\hat{\mathbf{l}} = \mathbf{R}\mathbf{R}^T\hat{\mathbf{l}} = \hat{\mathbf{y}}_i$ can be efficiently obtained by two back-substitutions. After solving the reduced linear system, the solution to the original equation in step 2 is therefore $(\mathbf{l}_i^{*+}) = [\hat{\mathbf{l}}_i^{*+}, 0]^T$. With the help of this technique, we are able to analyze datasets of a million rows and 10 thousand columns.

V. Experimental Evaluation

Because we are interested in understanding the differences between the s_{CT} and s_{L+} similarity measures for generating semantically associated itemsets, we conducted a series of experiments to highlight their differences. First, to illustrate the power of hypergraphs in finding associations via linking items, we synthesized a dataset for the *fish oil* example. Next, to illustrate the differences between the two methods, we evaluated both methods against a commonly used *shopping cart* dataset. Finally, encouraged by these results, we applied these methods to actual *electronic health records* to highlight their scalability and applicability to the medical domain.

A. Fish Oil

1) Dataset: As mentioned in Section I, *fish oil* and *Raynaud's syndrome* have been shown by Swanson [1] to be linked together indirectly via various *blood changes*. He found these associations from examining biomedical texts. As a proof of concept, we replicated this situation by synthesizing a table of 50 rows, which is about the same scale as in Swanson's experiment. Each row represents a set of terms generated to represent biomedical text. Each set of terms was specifically generated so that *fish oil* and *Raynaud's syndrome* never appear together. The column headers include *fish oil, blood changes*, *Raynaud's syndrome*. Six other random variables acted as noise. We then applied the s_{CT}, s_{L+} to the dataset. Specifically, we set a threshold for first generating top-15 2-itemsets using either similarity measure. Based on the generated 2-itemsets we used clique search to generate $(k > 2)$-itemsets.

2) Results: The hypergraph approach finds significant links between *fish oil* and *Raynaud's syndrome*, as demonstrated particularly well by the s_{CT} method as shown in Table I. Even the triplet was discovered by the clique search technique. Most notably, because their co-occurrence is zero, the association would never be discovered by traditional frequent itemset techniques such as the Apriori algorithm [23].

The s_{L+} method also picks-up the association, but it was fairly weak: the association is ranked 23rd among all 2-itemsets (column 3 in Table I lists the ranking of the s_{CT} results given by the s_{L+}). However, as our next evaluations suggest, the s_{L+} demonstrates other favorable qualities.

$\mathbf{s_{CT}}$	$\mathbf{s_{L+}}$ rank	**Freq**	**Itemset**
0.83	2	25	⟨ *blood_change, fish_oil* ⟩
0.83	1	25	⟨ *blood_change, Raynaud_synd* ⟩
0.79	**–**	**0**	⟨ ***blood_change, fish_oil, Raynaud_synd*** ⟩
0.76	–	10	⟨ *blood_change, fish_oil, f* ⟩
0.76	7	16	⟨ *blood_change, f* ⟩
0.76	6	16	⟨ *blood_change, d* ⟩
0.76	3	16	⟨ *blood_change, b* ⟩
0.75	9	15	⟨ *blood_change, a* ⟩
0.75	4	15	⟨ *blood_change, e* ⟩
0.73	10	14	⟨ *blood_change, c* ⟩
0.72	**23**	**0**	⟨ ***fish_oil, Raynaud_synd*** ⟩
0.70	10	10	⟨ *fish_oil, f* ⟩
0.70	–	10	⟨ *fish_oil, d* ⟩
0.70	9	9	⟨ *fish_oil, b* ⟩
0.68	20	6	⟨ *Raynaud_synd, f* ⟩

Table I
Top semantically associated itemsets generated by s_{CT} from the Synthetic Fish Oil dataset.

B. Shopping Cart

1) Dataset: To better understand how the s_{CT} method compares against the s_{L+} method, we tested them on a business shopping cart dataset. This dataset contains purchase information on 100 grocery items (represented by boolean column headers) for 2,127 shopping orders (corresponding to tuples). We applied s_{L+} and s_{CT} and set a threshold to include top-100 2-itemsets, based on which we subsequently used clique search to generate $(k > 2)$ itemsets. The top-10 2-itemset results and $(k > 2)$-itemsets corresponding to maximum cliques generated by s_{CT} and s_+ are reported in Table II and III respectively.

2) Results: Unlike the experiment on the fish oil dataset, We do not have specific hypothesis to validate in this test. After examining the results from both measures, we can only conclude they make intuitive sense. However, we observe that the difference between the s_{CT} and s_{L+} becomes more significant in this experiment. The s_{CT} tends to include itemsets with high support and the effect of indirect links is less pronounced. On the other hand, s_{L+} promotes items with support values towards the lower end. We also observe one drawback of the s_{CT} that the result is centered around items with large frequencies (i.e., many direct links to other nodes) and hence in a sense limiting the information (most itemsets are about *cheese*, *soup* and *cookie*). By contrast, the s_{L+} produces more diversified itemsets. This phenomenon is illustrated in Table IV by comparing the rankings under s_{CT}, s_{L+} and the ranking under support using Kendall-τ score. The degeneration of s_{CT} towards support is more pronounced in larger datasets as will be seen in the next experiment.

Finally we tested our methods on the dataset of electronic health records of real patients. This dataset is different from the above two datasets not only in scale but also in practical importance as described in the following.

	s_{CT}	Freq	Itemset
2-itemsets	0.74	39	⟨ *Cheese, Soup* ⟩
	0.73	32	⟨ *Cheese, Dried Fruit* ⟩
	0.72	36	⟨ *Dried, Fruit Soup* ⟩
	0.72	38	⟨ *Cookies, Soup* ⟩
	0.71	24	⟨ *Cheese, Cookies* ⟩
	0.70	30	⟨ *Cookies, Dried Fruit* ⟩
	0.68	31	⟨ *Cheese, Preserves* ⟩
	0.67	24	⟨ *Cheese, Wine* ⟩
	0.67	21	⟨ *Preserves, Soup* ⟩
	0.67	28	⟨ *Soup, Wine* ⟩
(k>2)-itemsets	0.64	0	⟨ *Canned Vegetables, Cheese, Cookies, Dried Fruit, Frozen Vegetables, Nuts, Preserves, Soup, Wine* ⟩

Table II
TOP SEMANTICALLY ASSOCIATED ITEMSETS GENERATED BY s_{CT} FROM THE SHOPPING CART DATASET.

	s_{L+}	Freq	Itemset
2-itemsets	10.17	3	⟨ *Sardines, Conditioner* ⟩
	8.17	6	⟨ *Toothbrushes, Nasal Sprays* ⟩
	6.70	6	⟨ *Yogurt, Anchovies* ⟩
	6.25	5	⟨ *Sports Magazines, Cottage Cheese* ⟩
	5.82	5	⟨ *Tofu, Sour Cream* ⟩
	5.79	3	⟨ *Toothbrushes, Acetominifen* ⟩
	4.77	4	⟨ *Sauces, Nasal Sprays* ⟩
	4.46	3	⟨ *Sports Magazines, Gum* ⟩
	4.43	4	⟨ *Sunglasses, Paper Dishes* ⟩
	4.05	5	⟨ *Tofu, Canned Fruit* ⟩
(k>2)-itemsets	4.51	2	⟨ *Canned Fruit, Sour Cream, Tofu* ⟩
	2.01	1	⟨ *Batteries, Cereal, Cooking Oil* ⟩
	1.75	5	⟨ *Canned Vegetables, Nuts, Waffles* ⟩

Table III
TOP SEMANTICALLY ASSOCIATED ITEMSETS GENERATED BY s_{L+} FROM THE SHOPPING CART DATASET.

C. Electronic Health Records

1) Dataset: In our third evaluation, we analyzed the electronic health records of real patients. Applying methods like the ones we have described to this kind of data is particularly relevant because of recent legislation aimed at increasing the meaningful use of electronic health records. Discovering meaningful semantically associated itemsets among the set of drugs and diseases identified in patients' clinical notes is a critical step toward identifying combinations of drug classes and co-morbidities, or risk-factors and co-morbidities that are common in patients with a certain outcome (for example, those suffering from myocardial infarction), toward building predictive risk models, as well as toward providing probable hypotheses about the possible causes of that outcome.

We obtained the set of drugs and diseases for each patient's clinical note by using a new tool, the *Annotator Workflow*, developed at the National Center for Biomedical Ontology (NCBO). The patient notes are from Stanford Hospital's Clinical Data Warehouse (STRIDE). These records archive over 17-years worth of patient data comprising of

	Support	
	Shopping cart	Electronic health
s_{CT}	0.58	0.82
s_{L+}	0.32	0.06

Table IV
THE KENDALL-τ SCORE BETWEEN RANKINGS OF ITEMSETS GENERATED BY s_{CT}, s_{L+} AND SUPPORT IN THE TWO EXPERIMENTS.

1.6 million patients, 15 million encounters, 25 million coded ICD9 diagnoses, and a combination of pathology, radiology, and transcription reports totaling over 9 million clinical notes (i.e., unstructured text).

From this set of 1.6 million patients, we extracted a cohort of patients that suffered from kidney failure. Out of those records, we applied our algorithms to all previous records in the patient's timeline, looking at just the set of drugs. Therefore, at a very simplistic level, the experiment result shows that semantically associated itemsets in this context could possibly represent sets of drugs that could lead toward kidney failure when used in combination.

2) Results: The cohort dataset described above contains 467791 rows (corresponding to patients' clinical notes) and 10167 columns (corresponding to annotated terms appeared in the notes). With the help of the techniques described in Section IV-C, we are able to compute L^+ in a tractable amount of time (Equation 3 and 4 are calculated within 4 hours on a Quad-Core AMD Opteron(tm) Processor with 8 gigabyte memory), based on which we can efficiently derive the s_{L+} itemsets. However, the calculation of s_{CT} on this scale is intractable because an exact computation of all pairwise s_{CT} requires to fill in a $|V|\times|V|$ similarity table. In order to ameliorate the computational cost, we exploit domain knowledge to identify 582 terms of particular interest and then apply both s_{CT} and s_{L+} on the reduced dataset. The results are shown in Table V and VI respectively, where we list top-10 2-itemsets and all (k >2)-itemsets corresponding to the maximum clique.

It is clear that, continuing the trend shown in the shopping cart analysis, the s_{CT} result becomes increasingly concordant with the support-based method. For illustrating this point of view, we calculate the Kendall-τ score between the ranking of itemsets generated by s_{CT}, s_{L+}, and support as shown in Table IV. We observe from the table that as the s_{CT} converges to support, the s_{L+} becomes even more distinct from it. The result is that the itemsets discovered by s_{CT} contain mostly general terms that are repeatedly found in the patients' notes. Although the association is reasonable but hardly interesting. On the contrary, the s_{L+} result is not affected by the dimension of data as well as the presence of items with massive support. It identifies itemsets of relatively low support but more closely bonded by indirect links.

To demonstrate the scalability of the method based on the s_{L+}, we also conducted the same analysis on the data of the

	s_{CT}	Freq	Itemset
2-itemsets	0.80	39204	⟨ *Calcium Chloride, Amiloride* ⟩
	0.77	29325	⟨ *Calcium Chloride, Aspirin* ⟩
	0.76	28644	⟨ *Calcium Chloride, Probenecid* ⟩
	0.73	24805	⟨ *Calcium Chloride, Furosemide* ⟩
	0.72	34271	⟨ *Calcium Chloride, Calcium* ⟩
	0.71	21481	⟨ *Calcium Chloride, Disulfiram* ⟩
	0.70	16814	⟨ *Calcium Chloride, Amphetamine* ⟩
	0.66	19850	⟨ *Calcium Chloride, Prednisone* ⟩
	0.65	12231	⟨ *Aspirin, Amiloride* ⟩
	0.65	12106	⟨ *Probenecid, Amiloride* ⟩
(k>2)-itemsets	0.56	0	⟨ *Calcium Chloride, Disulfiram, Amphetamine, Acetaminophen, Calcium, Aspirin, Probenecid, Amiloride, Prednisone, Furosemide* ⟩

Table V
TOP SEMANTICALLY ASSOCIATED ITEMSETS GENERATED BY s_{CT} FROM THE KIDNEY FAILURE COHORT OF THE ELECTRONIC HEALTH DATASET.

	s_{L+}	Freq	Itemset
2-itemsets	0.820	354	⟨ *sevoflurane, remifentanil* ⟩
	0.691	978	⟨ *frovatriptan, almotriptan* ⟩
	0.633	693	⟨ *Etomidate, Rocuronium* ⟩
	0.496	234	⟨ *Atazanavir, Pyrimethamine* ⟩
	0.420	3004	⟨ *ciclesonide, Fluorometholone* ⟩
	0.377	231	⟨ *naratriptan, Mefenamic Acid* ⟩
	0.373	1792	⟨ *ciclesonide, Vincristine* ⟩
	0.332	92	⟨ *Rocuronium, sevoflurane* ⟩
	0.325	1368	⟨ *tazarotene, halobetasol propionate* ⟩
	0.322	506	⟨ *Buprenorphine, alosetron* ⟩
(k>2)-itemsets	0.131	701	⟨ *Ketorolac, Flurbiprofen, Ketorolac, Etodolac, Sulindac, Piroxicam, Ketoprofen* ⟩

Table VI
TOP SEMANTICALLY ASSOCIATED ITEMSETS GENERATED BY s_{L+} FROM THE KIDNEY FAILURE COHORT OF THE ELECTRONIC HEALTH DATASET.

whole cohort after 2010. The data consisted 1 million rows and 10 thousand columns. We were able to produce the s_{L+} based 2-itemsets in 6 hours. The top results are shown in Table VII.

The discovered s_{L+} itemsets provide much valuable insights on the possible interrelationship between drugs. Some of them has been studied in the literature. For example, *sevoflurane/remifentanil* can be used for anaesthesia; *frovatriptan* and *almotriptan* are both oral treatment of migraine headache; *Etomidate* and *Rocuronium* can be used for rapid sequence intubation; etc. This area of research is still very new and there are no good gold standards to compare our results against. However, for single-item drugs that lead to kidney failure, SIDER[1] database lists drugs and their side-effects. Most notably, multi-itemsets are difficult to identify, but our methods have found not only *Ketoprofen* but it has also group other drugs like it (see the ($k > 2$)-itemset shown in Table VI, all of the items are anti-inflammatories). Our

[1]http://sideeffects.embl.de/se/C0035078/all

s_{L+}	Itemset
0.0301	⟨ *White faced hornet venom, Yellow hornet venom* ⟩
0.0195	⟨ *Trichloroacetic Acid, Trichloroacetate* ⟩
0.0108	⟨ *Cloxacillin Sodium, benzathine cloxacillin* ⟩
0.0101	⟨ *Methacycline, Methacycline hydrochloride* ⟩
0.01	⟨ *Entamoebiasis, Hepatic, Liver Abscess, Amebic* ⟩
0.0086	⟨ *butenafine, Butenafine hydrochloride* ⟩
0.0085	⟨ *Acetone, Cantharidin* ⟩
0.0085	⟨ *ethyl cellulose, Cantharidin* ⟩
0.0085	⟨ *ethyl cellulose, Acetone* ⟩
0.0085	⟨ *Poloxamer 407, Eucalyptol* ⟩

Table VII
TOP SEMANTICALLY ASSOCIATED ITEMSETS GENERATED BY s_{L+} FROM THE WHOLE ELECTRONIC HEALTH DATASET AFTER 2010. THE DATASET CONTAINS 1 MILLION ROWS AND 10K COLUMNS.

results are a matter of on-going evaluation with medical experts.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a method for discovering semantically associated itemsets. It is based on a hypergraph representation of the database where each column corresponds to a hypergraph node and each row corresponds to a hyperedge. We described two similarity measures to compute the strength of association between items which are used to generate semantically associated 2-itemsets. Specifically, we introduced the average commute time similarity, s_{CT}, based on the random walk model on hypergraph, and the inner-product similarity, s_{L+} , based on the Moore-Penrose pseudoinverse of the hypergraph Laplacian matrix. Given generated 2-itemsets, we proposed a hypergraph expansion method with two search strategies, namely, the clique and connected component search, to generate k-itemsets ($k > 2$).

We showed the proposed method is indeed capable of capturing semantically associated itemsets through experiments performed on three datasets ranging from low to high dimensionality. We observed that the s_{CT} tends to generate concordant itemsets with those generated by support-based method as the dimensionality grows, while s_{L+} performs well in consistently picking up items connected via indirect links. The semantically associated itemsets discovered by s_{L+} on the patients' clinical note dataset provide valuable insights on the possible interrelationship between drugs. The draw back of this method is that the s_{CT} measure does not scale well for large datasets. We have to resort to "a priori" pruning of the data in the experiment. We are going to investigate iterative formulas and approximation algorithms to improve the scalability.

Using hypergraph-based representation to model data has an important benefit of enabling systematic combination of top-down and bottom-up insight discovery methods. Domain knowledge encoded in ontologies has a natural graphical representation. We will study methods to put the data graph

and knowledge graph together in a principled way to achieve a synergy between data mining and domain knowledge.

VII. Acknowledgment

Haishan Liu and Dejing Dou's work in this paper is partially supported by the NIH/NIBIB with Grant No. R01EB007684. Ruoming Jin's work in this paper is partially supported by National Science Foundation under CAREER Award IIS-0953950. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the supporting institutions.

References

[1] D. R. Swanson, "Two medical literatures that are logically but not bibliographically connected," *Journal of the American Society for Information Science*, no. 4, pp. 228–233, Jan.

[2] I. Hodkinson and M. Otto, "Finite conformal hypergraph covers and Gaifman cliques in finite structures," *Bull. Symbolic Logic*, vol. 9, pp. 387–405, 2002.

[3] C. Berge, "Hypergraphs." *Bull. Symbolic Logic*, 1989.

[4] L. Lovasz, "Random Walks on Graphs: A Survey," in *Combinatorics*. Budapest: Janos Bolyai Math. Soc., 1993, pp. 353–397.

[5] D. Klein and M. Randic, "Resistance Distance," *J. Math. Chemistry*, vol. 12, pp. 81–95, 1993.

[6] F. Gobel and A. Jagers, "Random Walks on Graphs," *Stochastic Processes and Their Applications*, vol. 2, pp. 311–336, 1974.

[7] S. Barnett, Ed., *Matrices: Methods and Applications*. Oxford Univ. Press, 1992.

[8] D. Zhou, J. Huang, and B. Scholkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing Systems (NIPS) 19*. MIT Press, 2006, p. 2006.

[9] A. Sheth, B. Aleman-Meza, F. S. Arpinar, A. Sheth, C. Ramakrishnan, C. Bertram, Y. Warke, K. Anyanwu, B. Alemanmeza, I. B. Arpinar, , K. Kochut, C. Halaschek, C. Ramakrishnan, Y. Warke, D. Avant, F. S. Arpinar, K. Anyanwu, and K. Kochut, "Semantic association identification and knowledge discovery for national security applications," *Journal of Database Management*, vol. 16, pp. 33–53, 2005.

[10] 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*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 118–127. [Online]. Available: http://doi.acm.org/10.1145/1014052.1014068

[11] P.-N. Tan, V. Kumar, and J. Srivastava, "Indirect Association: Mining Higher Order Dependencies in Data," in *IN PRINCIPLES OF DATA MINING AND KNOWLEDGE DISCOVERY*, 2000, pp. 632–637.

[12] I. D. Melamed, "Automatic Construction Of Clean Broad-Coverage Translation Lexicons," in *In Proceedings of the 2nd Conference of the Association for Machine Translation in the Americas*, pp. 125–134.

[13] G. Das, H. Mannila, and P. Ronkainen, "Similarity of Attributes by External Probes," in *In Knowledge Discovery and Data Mining*. AAAI Press, 1997, pp. 23–29.

[14] Q. Wan and A. An, "Efficient mining of indirect associations using hi-mine," in *In Proceedings of 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2003*, 2003.

[15] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, p. 2007, 2006.

[16] J.-Y. Pan, H. Yang, C. Faloutsos, and P. Duygulu, "Cross-modal correlation mining using graph algorithms," *Knowledge Discovery and Data Mining: Challenges and Realities with Real World Data.*

[17] L. Yen, L. Vanvyve, D. Wouters, F. Fouss, F. Verleysen, and M. Saerens, "Clustering using a random-walk based distance measure," in *Proceedings of ESANN'2005*, 2005. [Online]. Available: http://citeseer.ist.psu.edu/yen05clustering.html

[18] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proc. VLDB Endow.*, vol. 2, pp. 718–729, August 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1687627.1687709

[19] C. R. Palmer and C. Faloutsos, "Electricity based external similarity of categorical attributes," in *In PAKDD 2003*. Springer, 2003, pp. 486–500.

[20] P. Doyle and J. Snell, "Random Walks and Electric Networks," *The Math. Assoc. of Am.*, 1984.

[21] R. Baeza-Yates and B. Ribeiro-Neto, Eds., *Modern Information Retrieval*. Addison-Wesley, 1999.

[22] I. Herstein and D. Winter, Eds., *Matrix Theory and Linear Algebra*. Maxwell Macmillan International Editions, 1988.

[23] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499. [Online]. Available: http://portal.acm.org/citation.cfm?id=645920.672836