On Typed-Paths in Multi-Typed Information Networks

Tim Weninger Corey Pennycuff
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556
+1-574-631-6770
tweninger@nd.edu, cpennycu@nd.edu

1. INTRODUCTION

Information networks are known to model the interactions and relationships among countless real-world, physical and abstract objects. Most commonly these information networks are assumed to be *untyped* or *homogeneous*, wherein all nodes in the network are objects of the same entity-type (*e.g.*, person, Web page) and therefore the links are untyped relationships (*e.g.*, friendship, hyperlink). Research on untyped information networks reveal interesting applications in biological, social, chemical and other physical networks.

Previous work on this subject has found that this particular abstraction is good at representing important real-world phenomena. Information networks are easily stored and manipulated by computer programs allowing for straightforward, albeit computationally complex, analysis. In this paper, we define an information network formally as follows:

An information network is a directed or undirected simple graph $G = (V, E)$ with a vertex-type mapping function $\tau: V \to A$ and an edge-type mapping function $\phi: E \to R$, where each vertex $v \in V$ represents a real-world entity and belongs to a particular type $\tau(v) \in A$, each edge $e \in E$ belongs to a particular relation-type $\phi(e) \in R$, and if two edges belong to the same relation-type then the two links also share the same starting vertex-type and ending vertex-type.

Most graph definitions leave out the vertex and edge-type mapping function. However, this paper explicitly defines vertex-types and edge-types in the network. When the number of vertex-types $|A| > 1$ or the number of edge-types $|R| > 1$ then the network is called a **typed information network**[1], otherwise it is called an **untyped information network** or defaults to information network or common graph terminology, *i.e.*, simple graph, multigraph, etc.

1.1 Typed Information Networks

In this work we consider networks where $|A| > 1$ and $|R| > 1$, that is, we investigate typed information networks. With a typed information network it is necessary to determine the vertex and edge types in order to reason about the semantics and structure of the network. In many cases the *network schema* is easily determined or provided with the data.

A network schema denoted $T_G = (A, R)$, is the set of vertex-type mappings $\tau: V \to A$ and edge-type mappings $\phi(e) \in R$ for an information network $G = (V, E)$ [1].

The network schema provides type constraints which act to guide the user or application in reasoning over the network. In many cases, a given information network is provided without its network schema yet still possesses multiple types. In such cases the information network is said to possess a *hidden network* schema. As a preliminary proof of concept, this paper considers Wikipedia to be such an information network that contains a hidden network schema.

1.1.1 *Article Types in Wikipedia*

The category graph on Wikipedia provides a strong signal as to the “type” of a particular article. Like other category listings, the Wikipedia category graph is a loose hierarchy (actually, unfortunately, a cyclic graph) in which categories are collections of similar articles. As illustrated in Figure 1, categories can have children and parent categories which describe more fine-grained or coarse grained information respectively. The Wikipedia category graph is rooted at Category:Articles. However, there are four second level categories which provide greater insight into the overall organization of Wikipedia: Categories by Parameter, Fundamental Categories, Categories by Main Topic Classification, and Spoken Articles.

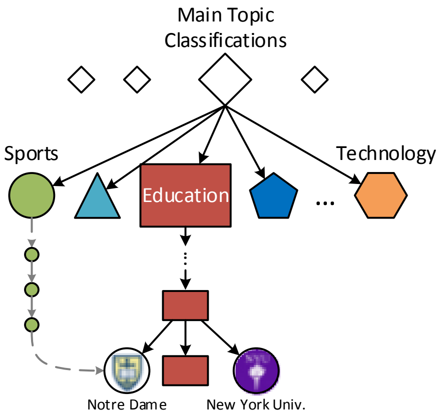

Figure 1: Illustration of how the Wikipedia category graph informs article types. Articles are often in multiple top-level categories. In this work we chose the category closest to the article.

[1] These networks are also known as heterogeneous information networks. However, this wording conflicts with heterogeneous graphs, which deals with degree-heterogenity.

In this work we choose to anchor the category hierarchy at Categories by Main Topic Classification (T) because it essentially classifies each article into one or more broad types from among 24 types: Agriculture, Arts, Belief, Business, etc. In this work each article is assigned a "type" according to its main topic classification: $A = T \therefore \tau(v) \in T$.

1.2 Frequent Typed-Paths

In graph mining literature, a path P in a graph G of length k from a vertex u to u' is a sequence $(v_0, v_1, \ldots, v_k)$ of vertices such that $v_0 = u$ and $v_k = u'$ and $(v_{i-1}, v_i) \in E$ for $i = 1,2, \ldots, k$. Most often graph mining algorithms and tools seek to find frequent paths or subgraphs in a database of graphs where the items in each graph match exactly. In this paper, we aim to find frequent paths in a Wikipedia graph, matching on the article's type $\tau(v)$. As a result, we aim to find frequent typed-paths in a typed information network. The results of this investigation could lead to new and interesting insights in to how humans collectively organize information.

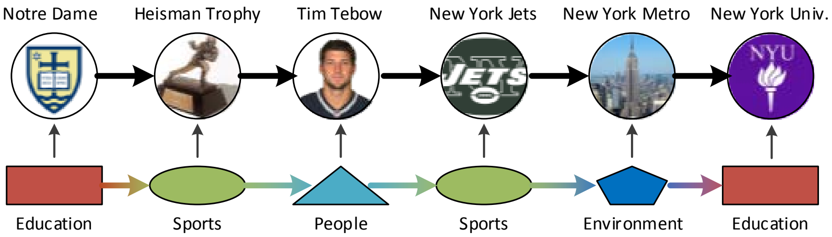

Figure 2: Candidate article path and its corresponding type-path between Notre Dame and NYU Wiki-pages.

This paper presents the method, results and some preliminary analysis on the mining of frequent typed paths from the typed-enhanced Wikipedia network.

2. MINING FREQUENT TYPED-PATHS

The goal of this preliminary work is to find frequent typed-paths in the Wikipedia dataset. For example, the illustration in Figure 2 shows a path from the Notre Dame Wiki-page to the NYU Wiki-page. In this study, a path is represented by its top-level category-types indicated below the Wiki-page in Figure 2. Thus, the candidate typed-path illustrated in Figure 2 is Ed → S → P → S → En → Ed. This particular path is considered to be a *frequent* typed-path if the number of times it appears is greater than the user-supplied minimum support threshold Θ.

Frequent paths can be informative by themselves, but more interesting paths can be found when we limit the results to only *closed* and/or *maximal*, frequent typed-paths [2].

Traditional frequent pattern mining approaches can be adapted and applied to find frequent paths in graphs. Unfortunately, as we will show, these approaches are impossibly slow on even modest data sets.

2.1 Frequent Path Mining

Traditional market-basket algorithms such as Apriori, and FP-Growth examine non-sequential datasets. These traditional algorithms cannot be adopted because graph-paths are ordered sequences of items. Sequential pattern mining algorithms, such as gSpan and PrefixSpan do mine ordered sequences, such as graph paths, but they must iterate through a database of sequences. For our purposes this would require an enumeration of all possible paths. This enumeration would be an exponential on the size of the graph – which is prohibitively large for any meaningful computation.

Some attempts were made to grow frequent patterns from within the typed-graph, but these attempts were impossibly slow even for extremely high Θ.

2.2 Approximate Frequent Path Mining

We approximate frequent path mining by randomly walking through the graph. Our approach adapts the PageRank/random surfer algorithm and records the paths that the algorithm "walks" as candidate paths. The approximation algorithm is shown in Algorithm 1. The procedure works by recording steps that the random walker takes a candidate paths. One path ends and another begins when the "random jump" occurs in line 7. The procedure repeats for n iterations. Upon completion the Path counter variable P contains the set of all paths and their number of occurrences. Paths which do not meet the minimum support criteria Θ are removed.

Finding closed and maximal patterns is not shown in Algorithm 1, the mining of closed and maximal patterns is a straightforward process, and its description is outside the scope of this paper.

```
ApproxPathMining($G$=Graph, $v$=Vertex-type,
$p$=Path, $\alpha$=Jumping Probability, $P$=Path Counter,
n=Number of Random Walks to perform
1  for(1 to $n$)
2      $v \leftarrow$ rand($G$); $p$.add($v$);
3      while(true)
4          $v \leftarrow$ rand(outlinks($v$));
5          $p$.add($v$);
6          $P$.put($p, P$.get($p$) + 1);
7          if (rand(0 ... 1) < $\alpha$) break;
```

Algorithm 1: Approximate Typed Path Mining with PageRank walks.

3. EVALUATION

In this paper, we perform experiments on the Wikipedia dataset where each Wiki-article is assigned a type according to its most-direct top-level category from the main topic classification category set[2]. We performed a brief runtime and output analysis on the ApproxPathMining algorithm with varied values for α and n. Unfortunately, it is impossible to determine the accuracy of the approximation scheme because the Wikipedia dataset is too large to retrieve exact results; nevertheless, we show the most frequent, maximal typed-paths as an indication of the nature of the results.

3.1 Dataset

The Wikipedia dataset used in this paper comes from the Wikipedia database snapshot dated July 8, 2013. It contains 10,276,554 Wiki-articles, 364,767,694 article-to-article directed links, 1,018,609 categories, and 24 top level categories.

3.2 Runtime Analysis

For runtime analysis we used a 1,000,000 edge subset of the Wikipedia dataset. Figure 3 shows the execution time for 10,000 iterations and the number of frequent typed-paths and the number of maximal typed-paths. The results show that the approximation scheme is fast, even for a million edges. Furthermore, the results in subfigures b and c show, as expected, a decrease in the number of frequent type-paths as the minimum support threshold Θ increases. Interestingly, the number of typed-paths found decreases as the jumping probability α increases. This is because the high values of α cause frequent jumps giving a higher likelihood to short typed-paths. There are a limited number of short typed-paths (*i.e.*, there are only 24 possible length-1 paths corresponding to 24 top-level categories).

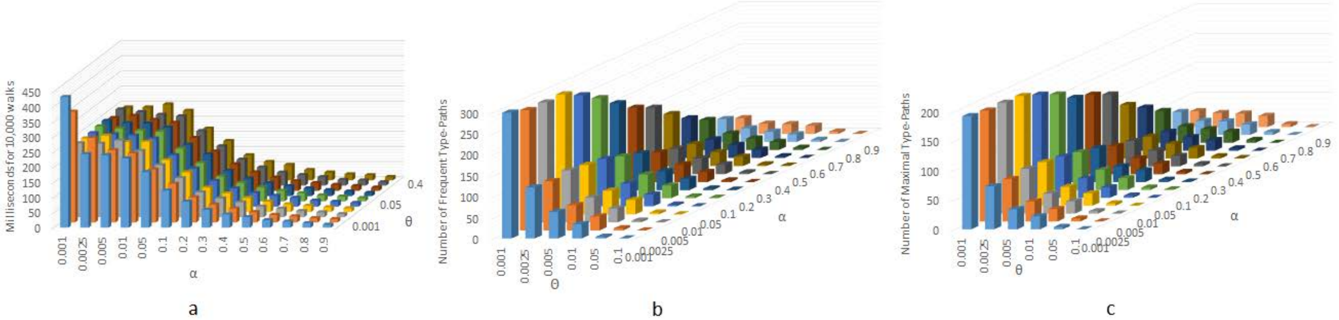

Figure 3: a. Execution time in milliseconds, **b.** Number of frequent typed-paths found, **c.** Number of maximal typed-paths found, for varying restart probabilities $\boldsymbol{\alpha}$ and minimum support thresholds $\boldsymbol{\Theta}$.

3.3 Frequent Typed-Paths Results

We ran the approximation algorithm with n=100,000, α=.05, Θ=.1%. The algorithm completed in 43.7 seconds and returned 179 frequent typed-paths, of which 118 were maximal typed-paths. The most frequent maximal typed-path was `Chronology→Culture` which was walked 511 times. The longest typed-path was a chain of 8 `Geography` pages. The full list of frequent paths is available at http://cse.nd.edu/~tweninge. Due to page constraints, Figure 3 shows the graph of the aggregation of all paths. This is an indication of the nature of the typed-paths retrieved.

These results seem intuitive. People and Culture are closely related, as is Chronology with People, Culture and Politics. Agriculture is linked with Geography; Humanities with People and so on.

Future work is needed to expand these results to include different top-level categories, different granularities and human paths through the typed-network.

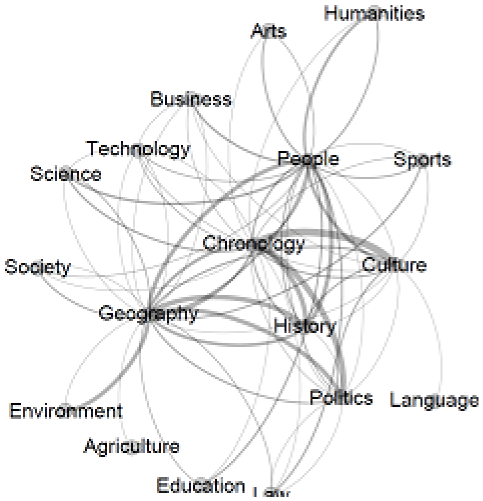

Figure 4: This graph demonstrates the types of connections made when all maximal typed-paths are aggregated. Edge thickness indicates frequency.

4. REFERENCES

[1] Sun, Y., and Han J. 2012. Mining Heterogeneous Information Networks: Principles and Methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*. (Jul 23, 2012). Morgan and Claypool.

[2] Han, J., Kamber, M., and Pei, J. Data Mining: Concepts and Techniques. *The Morgan Kaufmann Series in Data Management Systems.* (Jul 6, 2011). Morgan and Claypool.

[2] http://en.wikipedia.org/wiki/Category:Main_topic_classifications