Link Prediction for Annotation Graphs using Graph Summarization

Andreas Thor[1], Philip Anderson[1], Louiqa Raschid[1], Saket Navlakha[1], Barna Saha[1], Samir Khuller[1], Xiao-Ning Zhang[2]

[1] University of Maryland, USA
[2] St. Bonaventure University, USA
thor@umiacs.umd.edu, phand@umd.edu, louiqa@umiacs.umd.edu, saket@cs.umd.edu, barna@cs.umd.edu, samir@cs.umd.edu, xzhang@sbu.edu

Abstract. Annotation graph datasets are a natural representation of scientific knowledge. They are common in the life sciences where genes or proteins are annotated with controlled vocabulary terms (CV terms) from ontologies. The W3C Linking Open Data (LOD) initiative and semantic Web technologies are playing a leading role in making such datasets widely available. Scientists can mine these datasets to discover patterns of annotation. While ontology alignment and integration across datasets has been explored in the context of the semantic Web, there is no current approach to mine such patterns in annotation graph datasets. In this paper, we propose a novel approach for link prediction; it is a preliminary task when discovering more complex patterns. Our prediction is based on a complementary methodology of graph summarization (GS) and dense subgraphs (DSG). GS can exploit and summarize knowledge captured within the ontologies and in the annotation patterns. DSG uses the ontology structure, in particular the distance between CV terms, to filter the graph, and to find promising subgraphs. We develop a scoring function based on multiple heuristics to rank the predictions. We perform an extensive evaluation on Arabidopsis thaliana genes.

Keywords: Link prediction; Graph summarization; Dense subgraphs; Linking Open Data ontology alignment.

1 Introduction

Among the many "killer apps" that could be enabled by the Linking Open Data (LOD) initiative [2, 20] and semantic Web technologies, the ability for scientists to mine annotation graph datasets and to determine actionable patterns shows great promise. A majority of the links in LOD datasets are at the instance level as exemplified by the *owl:sameAs* relationship type. However, there has been a rapid emergence of biological and biomedical datasets that are typically annotated using controlled vacabulary (CV) terms from ontologies. For example, the US NIH clinical trial data `ClinicalTrial.gov` has been linked to (1) PubMed publications and Medical Subject Header (MeSH) terms; (2) drug names and

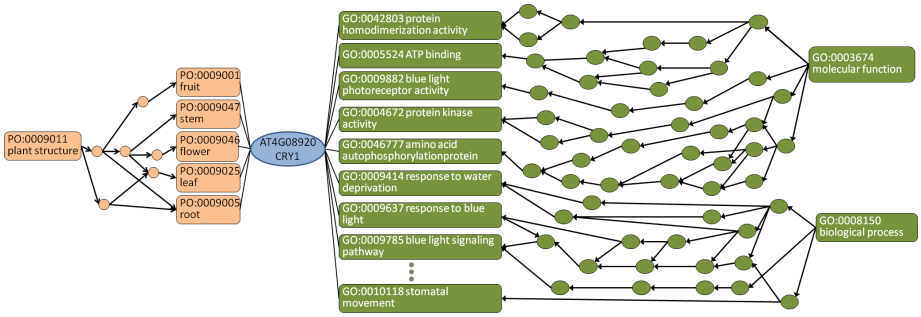


Fig. 1. GO and PO annotations for gene CRY1

drug terms from `RxNorm`; (3) disease names and terms from `Diseasome`; etc. This has lead to a rich annotation graph dataset [8]. Semantic Web research has laid the groundwork for research in link prediction and pattern discovery in the context of annotation graph datasets as discussed next.

1.1 Motivating Example

Arabidopsis thaliana is a model organism and TAIR `http://www.arabidopsis.org/` is the primary source of annotated data for Arabidopsis genes. Each gene in TAIR is marked up with CV terms from the Gene Ontology and from the Plant Ontology. The resulting tripartite annotation graph (TAG) is illustrated in Figure 1 where we visualize the annotations for gene `CRY1`; PO terms are on the left and GO terms are on the right of `CRY1`. The TAG has been enhanced to include relevant fragments of the GO and PO ontologies. As of October 2010 there were 18 GO and 36 PO annotations for `CRY1`. The figure illustrates partial annotations (due to lack of space). The annotations can be represented using an RDF class `gene_GO_PO_TAGtriplet` as follows:

t1: (TAGtripletID rdf:type gene_GO_PO_TAGtriplet)
t2: (TAGtripletID gene_ID name-of-gene)
t3: (TAGtripletID GO_ID uri-of-GO-CV-term)
t4: (TAGtripletID PO_ID uri-of-PO-CV-term)

A scientist is typically interested in a set of genes of interest within a biological context, e.g., flowering time genes or photomorphogenesis genes. Given the resulting large annotation graph dataset, the scientist would like to be presented with interesting patterns. For photomorphogenesis, a pattern may correspond to the following 4 TAG triplets for `CRY2` and `PHOT1`; note that we use a comma separated representation `(gene, GO CV term, PO CV term)`, instead of the RDF triples for ease of presentation and due to space constraints:

$TAGtriplet T_1$: (CRY2, GO_5773:vacuole, PO_13:cauline leaf)
$TAGtriplet T_2$: (CRY2, GO_5773:vacuole, PO_37:shoot apex)
$TAGtriplet T_3$: (PHOT1, GO_5773:vacuole, PO_13:cauline leaf)
$TAGtriplet T_4$: (PHOT1, GO_5773:vacuole, PO_37:shoot apex)

Subsequently, she will explore the literature to understand the evidence. PHOT1 and CRY2 belong to two different groups of blue light receptors, namely phototropins (PHOT1) and cryptochromes (CRY2). To date there has been *no evidence reported in the literature that confirm any interactions* between these 2 groups. A literature search identified 2 independent studies of interest [11, 24] that provide some background evidence. The set of 4 TAG triplets, in conjunction with the 2 studies, may lead her to design a set of bench experiments to validate the potential interaction in the vacuole between CRY2 and PHOT1.

1.2 Challenges and Contributions

While scientists are interested in complex patterns, in this paper, we examine a simpler task of link prediction. We predict edges between genes and GO CV terms or edges between genes and PO CV terms. We briefly summarize the challenges of link prediction for the annotation graph datasets. First, the TAG is a layered graph. Layered graphs impose restrictions on the link prediction process, e.g., the neighborhoods of two nodes in neighboring layers are disjoint and only edges between neighboring layers should be predicted. This restriction makes many popular prediction approaches ineffective as will be discussed.

The next challenge is the a heterogeneity of biological knowledge. As seen in the previous example, a set of gene_GO_PO_TAGtriplets forms a complex and interesting cross-ontology pattern. The GO ontology is focused on universal biological processes, e.g., DNA binding. It does not capture organism-specific processes, e.g., leaf development. The PO ontology is designed to capture such organism specific knowledge. Thus, a gene_GO_PO_TAGtriplet, or a complex pattern of multiple triplets, may be used to determine when a plant specific biological phenomenon has a relationship with a ubiquitous biological process.

A related challenge is identifying an area or subgraph of the dataset to make predictions or find patterns. Ontologies capture multiple relationship types between CV terms that can be exploited for prediction. GO supports multiple relationship types including *is_a*, *part_of* and *regulates*. From Figure 1, the GO CV term blue light photoreceptor activity is *part_of* blue light signaling pathway which *is_a* cellular response to blue light which *is_a* response to blue light. CRY1 is annotated with blue light photoreceptor activity and response to blue light. PO has relationship types *is_a*, *part_of* and *develops_from*. Our challenge is to restrict the patterns of gene_GO_PO_TAGtriplets so that they favor GO CV terms (or PO CV terms) that are closely related.

Our observation is that the edges of each relationship type are not uniformly distributed across the ontology structure. For GO, the edges of type *is_a* are dominant, and thus all the edges of any path in GO are more likely to be of

this type. The edges relevant to regulation are more densely placed in specific areas of the ontology; thus, an edge of this type also has a greater probability that an adjacent edge is of the same type. For PO, while neither *is_a* nor *part_of* dominate, the edge distribution of these types are similarly concentrated so that an edge of one type is more likely to have an adjacent edge of the same type.

Based on these observations, our first attempt at prediction will use the *topological shortest path distance on undirected graphs*, between 2 CV terms, as a proxy for relatedness. We note that this path length metric is affected by both human annotation patterns and the ontology structure representing biological knowledge, e.g., the depth of the tree along any branch. We will consider the impact of the GO (PO) relationship type(s) on the path based distance metric in future research. Our link prediction framework relies on 2 complementary approaches. Graph summarization (GS) is a minimum description length (MDL) encoding that represents a graph with a *signature* and *corrections*. Such a representation is intuitive for both explanation and visualization. Since annotation graph datasets may be large and sparse, high quality predictions must rely on finding good candidate regions or subgraphs. Dense subgraphs (DSG) is a methodology to find such regions that include clique-like structures, i.e., cliques with missing edges. Variations of the dense subgraph whose nodes satisfy some *distance restriction* is also useful to ensure possible relatedness of the CV terms. Our research makes the following contributions:

- We develop a prediction framework that can be used for both unsupervised or supervised learning. We focus on unsupervised learning in this paper. We perform an extensive evaluation on the annotation graph of TAIR.
- Our evaluation illustrates the benefit of the DSG and the *distance restriction* to identify a potential subgraph so as to increase prediction accuracy. We further show that high values of the scoring function, or predicted edges with high confidence, are correlated with increasing prediction accuracy.

Due to space limitations, our examples only involve TAGs; however, our prediction framework is not limited to TAGs. We have applied our framework to a layered graph of 5 layers; beyond 5 layers, we are unclear if the patterns and predictions will be meaningful. We are also studying the clinical trial dataset; this is a star graph with a clinical trial having links to PubMed publications, MeSH terms, (disease) conditions, interventions (drugs or treatments), etc.

1.3 Related Work

Semantic Web research has addressed information integration using ontologies and ontology alignment [9, 25]. There are also multiple projects and tools for annotation, e.g., Annotea/Amaya [10] and OntoAnnotate [20].

Graph data mining covers a broad range of methods dealing with the identification of (sub)structures and patterns in graphs. Popular techniques are, amongst others, graph clustering, community detection and finding cliques. Our work builds upon two complementary graph methods: graph summarization [23]

and dense subgraphs [27]. To the best of our knowledge, we are the first to consider the synergy of these two approaches.

Link prediction is a subtask of link mining [21]; prediction in bipartite and tripartite graphs is also of interest [15, 26]. Prediction methods can be supervised or unsupervised. Supervised link prediction methods (e.g., [1, 7, 26]) utilize training and test data for the generation and evaluation of a prediction model. Unsupervised link prediction in graphs is a well known problem [18]. There are two types of approaches: methods based on node neighborhoods and methods using the ensemble of all paths between two nodes. We discuss their disadvantages for tripartite graphs in Section 3. Many approaches for predicting annotations in the biological web are available [3, 5, 17]. The AraNet system [17] predicts GO functional annotations for Arabidopsis using a variety of biological knowledge; details are discussed with our evaluation in Section 4.

2 Problem Definition

A **tripartite annotation graph (TAG)** is an undirected layered tripartite graph $G = ((A, B, C), (X, Y))$ with three pairwise disjoint sets of nodes A, B, and C and two sets of edges $X \subseteq A \times B$ and $Y \subseteq C \times B$. Figure 2 shows an example of a TAG. For example, in the TAIR annotated graph, the node sets A, B, and C correspond to POs, genes, and GOs, respectively. The sets of edges then reflect gene annotations using POs (X) and GOs (Y).

We study the **link prediction** problem for TAGs. Given a TAG G at time t_1 and a future time t_2, we assume that edges will be added during the transition from the original graph G_1 to the new graph G_2, i.e., $G_1 = ((A, B, C), (X, Y))$ and $G_2 = ((A, B, C), (X \cup X_{new}, Y \cup Y_{new}))$. The goal of link prediction is to infer the set of new edges based on the original graph G_1 only. Ideally the **predicted edges** $P_X(G)$ and $P_Y(G)$ are the added edges, i.e., $P_X(G) = X_{new}$ and $P_Y(G) = Y_{new}$.

For a given TAG $G = ((A, B, C), (X, Y))$ we refer to X and Y as the set of observed edges. We call all other possible edges, i.e., $((A \times B) - X) \cup ((C \times B) - Y)$ **potential edges**. Predicted edges $P_X(G)$ and $P_Y(G)$ and new edges are subsets of the corresponding potential edges.

Note that we consider only edge additions and we do not consider node additions for the transition from G_1 to G_2. In biological terms, we plan to use prior annotations to existing PO and GO nodes in G_1 to predict new edges in G_2. We are not attempting to predict new annotations to new PO or GO nodes that do not occur in G_1.

3 Approach

Unsupervised link prediction in graphs is a well known problem, e.g., see [18] for a survey on link prediction approaches in social networks. Basically there are two types of approaches. Neighborhood-based approaches consider the sets of node neighbors $N(a)$ and $N(b)$ for a potential edge (a, b) and determine a prediction

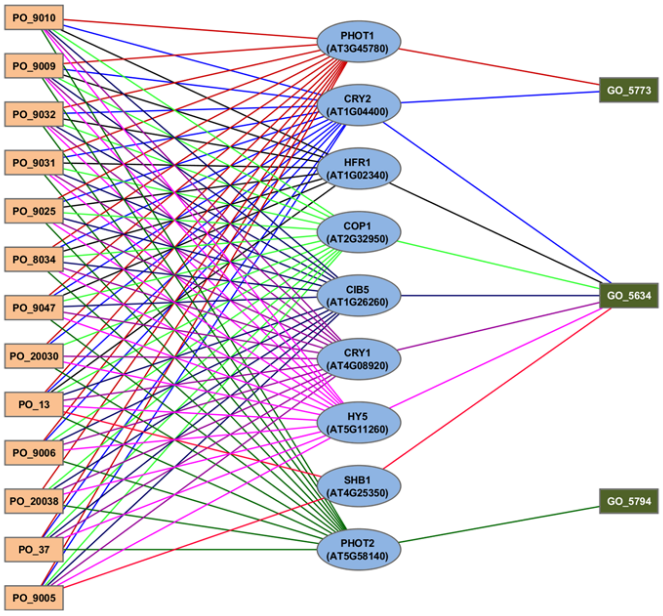


Fig. 2. Example of a TAG $G = ((A, B, C), (X, Y))$ with $|A| = 13$ PO nodes, $|B| = 9$ genes, and $|C| = 3$ GO nodes. The nodes are connected by $|X| = 98$ edges of type $A \times B$ and $|Y| = 10$ edges of type $C \times B$.

probability based on the (relative) overlap of these two sets. Methods based on the ensemble of all paths aggregate all paths from a through b to a combined prediction score. Shorter paths usually have a higher impact than longer paths and the more paths exist the higher the score will be.

Unfortunately, these types of approaches are not suited to TAGs. Neighborhood-based approaches will even fail for TAGs because the sets $N(a)$ and $N(b)$ are disjoint. Given a tripartite graph $G = ((A, B, C), (X, Y))$ and a potential edge (a, b) with $a \in A$ and $b \in B$, the node neighbors of a are in B ($N(a) \subseteq B$) and b's neighbors are in A ($N(b) \subseteq A$) and therefore $N(a) \cap N(b) = \emptyset$. On the other hand, path-based approaches are in general applicable for tripartite graphs but will produce similar prediction scores for many potential edges due to the structure of a tripartite graphs for two reasons. First, the minimal path length for a potential edge (a, b) equals 3 because there are only two possible path types ($a \to b' \to c' \to b$) or ($a \to b' \to a' \to b$). Second, most potential edges will have multiple paths with length 3 because it is very likely in the annotated biological web that any two genes b and b' have (at least) one GO (a') or PO (c') in common. Furthermore path-based approaches are not able to benefit from the rich ontology knowledge because they do not distinguish paths between the three layers (GO/genes/PO) and paths within the ontologies (PO, GO).

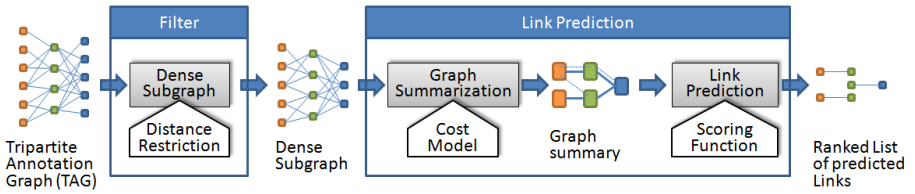


Fig. 3. The proposed link prediction framework combines graph summarization with link prediction functions. The original TAG can be subject to an optional filter step to identify dense subgraphs.

We therefore propose a different approach that employs graph summarization that transforms a graph into an equivalent compact graph representation using super nodes (groups of nodes) and super edges (edges between super nodes). The summary reflects the basic pattern (structure) of the graph and is accompanied by a list of corrections, i.e., deletions and additions, that express differences between the graph and its simplified pattern. The idea of our link prediction approach is that adding predicted edges reinforces the underlying graph pattern, i.e., predicted edges are the missing building blocks for existing patterns.

Figure 4 illustrates a possible summarization of the graph shown in Figure 2. The utilization of a graph summary has several advantages. First, the summary gives a better understanding of the overall structure of the underlying graph and may itself be used for visualization. Second, the corrections, foremost deletions, are intuitive indicators for edge prediction. Third, the summary captures semantic knowledge not only about individual nodes and their connections but also about groups of related nodes.

Figure 3 illustrates the overall scheme of our approach. The input is a TAG G and the output is a ranked list of predicted edges. Our approach consists of three consecutive steps. The first step is optional and deals with the identification of dense subgraphs, i.e., highly connected subgraphs of G like (almost) cliques. The goal is to identify interesting regions of the graph by extracting a relevant subgraph. Next, graph summarization transforms the graph into an equivalent compact graph representation using super nodes (groups of nodes) and super edges (edges between super nodes). The summarized graph is then input to the last step. A prediction function computes prediction scores for potential edges and returns a ranked list. Our approach is not limited to TAGs. A K-partite layered graph can be first converted to a more general (bi-partite) graph before creating a DSG and applying graph summarization.

3.1 Dense Subgraphs

Given an initial tripartite graph, a challenge is to find interesting regions of the graph, i.e., candidate subgraphs, that can lead to accurate predictions. We commence with the premise that an area of the graph that is rich or dense with annotation is an interesting region to identify candidate subgraphs. For example,

for a set of genes, if each is annotated with a set of GO terms and/or a set of PO terms, then the set of genes and GO terms, or the set of genes and PO terms, form a clique. We thus exploit cliques, or dense subgraphs (DSG) representing cliques with missing edges.

Density is a measure of the connectedness of a subgraph; it is the ratio of the number of induced edges to the number of vertices in the subgraph. Even though there are an exponential number of subgraphs, a subgraph of maximum density can be found in polynomial time [16, 6, 4]. In contrast, the maximum clique problem to find the subgraph of largest size having all possible edges is NP-hard; it is even NP hard to obtain any non-trivial approximation. Finding densest subgraphs with additional size constraints is NP hard [13]; yet, they are more amenable to approximation than the maximum clique problem.

Recall that our annotation graph is a tripartite graph $G = ((A, B, C), (X, Y))$. We employ our approach in [27] and thus first transform the tripartite graph G in the form of a bipartite graph $G' = (A, C, E)$ between the two sets A and C of outer nodes in G. The bipartite graph is a weighted graph where each edge $e = (a, c) \in E$ is labeled with the number of nodes $b \in B$ that have links to a and c in the tripartite graph, i.e., $(a, b) \in X$ and $(c, b) \in Y$. We then compute a densest bipartite subgraph $G'_{dense} = (A', C', E)$ by choosing subsets $A' \subset A$ and $C' \subset C$ to maximize the density of the subgraph, which is defined as $\frac{w'(E)}{|A|+|C|}$. Here $w'(E)$ denotes the weight of the edges in the subgraph induced by E. Finally, we build the dense tripartite graph G_{dense} out of the computed dense bipartite graph G'_{dense} by adding all intermediate nodes $b \in B$ that are connected to at least one $a \in A'$ or $c \in C'$.

An interesting variation on the DSG includes a **distance restriction** according to the ontology of nodes. In the annotated biological web (see Figure1) nodes from PO and GO are hierarchically arranged to reflect their relationships (e.g., is-a or part-of). Assume we are given a distance metric d_A (d_C) that specifies distances between pairs of nodes in set A (C). We are also given distance thresholds $\tau_A(\tau_C)$. The goal is to compute a densest subgraph G'_S that ensures that for all node pairs of A (C) are within a given distance. For any pair of vertices $a_1, a_2 \in A_S$ we have $d_A(a_1, a_2) \leq \tau_A$, and the same condition holds for pairs of vertices in C_S, namely that for all $c_1, c_2 \in C_S$ we have $d_C(c_1, c_2) \leq \tau_C$. We will evaluate the influence of a distance restriction in Section 4.

The distance restricted DSG algorithm calls a routine with complexity $O(n^3 \cdot log(n))$, where n is the number of nodes in a valid distance-restricted subgraph; it is called once for each pair of nodes in A, and for each pair in C. We have also implemented a linear time greedy 2-approximation to DSG that greatly outperforms our previous running time results reported in [27]; this solution was previously reported in [4, 14].

3.2 Graph Summarization

We start with the intuition that a summary of a tripartite graph is also a graph. The summary must however include a compact representation that can be easily

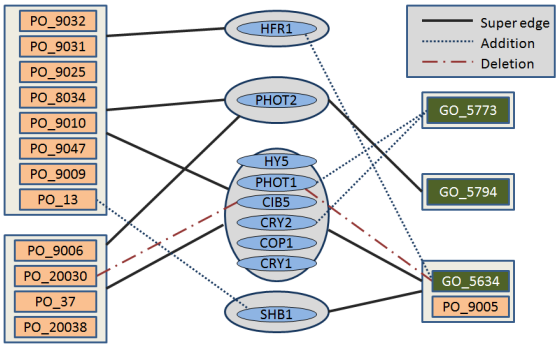


Fig. 4. Possible summary of the graph in Figure 2. The summary has 9 supernodes, 8 superedges, two deletions (PO_20030, CIB5) and (PHOT1, GO_5643) and 6 additions.

visualized and that can be used for making predictions. While there are many methods to summarize graphs, we focus on the graph summarization (GS) approach of [22, 23]. Their graph summary is an aggregate graph comprised of a signature and corrections. It is the first application of minimum description length (MDL) principles to graph summarization and has the added benefit of providing intuitive course-level summaries that are well suited for visualization and link prediction.

A **graph summary** of a graph $G = ((A, B, C), (X, Y))$ consists of a graph **signature** $\Sigma(G)$ and a set of **corrections** $\Delta(G)$. The graph signature is defined as follows: $\Sigma(G) = ((S_{AC}, S_B), S_{XY})$. The sets S_{AC} and S_B are disjoint partitionings of $A \cup C$ and B, respectively, that cover all elements of these sets. Each element of S_{AC} or S_B is a **super node** and consists of one or more nodes of the original graph. Elements of S_{XY} are called **super edges** and they represent edges between super nodes, i.e., $S_{XY} \subseteq S_{AC} \times S_B$. The second part of a summary is the sets of edge additions and deletions $\Delta(G) = (S_{add}, S_{del})$. All edge additions are edges of the original graph G, i.e., $S_{add} \subseteq X \cup Y$. Deletions are edges between nodes of G that do not have an edge in the original graph, i.e., $S_{Del} \subseteq ((A \cup C) \times B) - (X \cup Y)$. Figure 4 depicts a possible summarization of the graph shown in Figure 2.

The summarization algorithms makes sure that $G \equiv (\Sigma(G), \Delta(G))$, i.e., the original graph G can be reconstructed based on the graph summary and the edge corrections $\Delta(G)$. The nodes A, B, and C are "flattened" sets of S_{AC} and S_B, respectively. A super edge between two super nodes $s_{AC} \in S_{AC}$ and $s_B \in S_B$ represents the set of all edges between any node of s_{AC} and any node of s_B. The original edges can therefore be reconstructed by computing the Cartesian product of all super edges with consideration of edge corrections $\Delta(G)$. For example, X is therefore $X = \{(a, b) | a \in A \wedge b \in B \wedge a \in s_{AC} \wedge b \in s_B \wedge (s_{AC}, s_B) \in (S_{XY} \cup S_{add} - S_{del})\}$.

Graph summarization is based on a two-part minimum description length encoding. We use a greedy agglomerative clustering heuristic. At first, each node

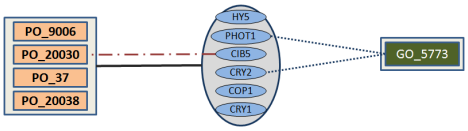


Fig. 5. Detail of Figure 4 for computing prediction scores for potential edges (PO_20030, CIB5) and (GO_5773, CIB5).

belongs to its own supernode. Then, in each step, the pair of supernodes are merged that result in the greatest reduction in representation cost. When the cost of merging any pair becomes negative, the algorithm naturally terminates. There are no parameters or thresholds to set. The complexity of the original GS problem is currently unknown. However, if nodes are allowed to belong to more than one super node (i.e., overlapping supernodes), the problem reduces to finding the maximum clique in a graph, which is NP-hard.

The possible summaries of a graph will depend on the **cost model** used for an MDL encoding. In general, the cost model is a triple (α, β, γ) that assigns weights to the number of superedges, deletions, and additions, respectively. Graph summarization looks for a graph summary with a minimal cost of $C(G) = \alpha \cdot |S_{XY}| + \beta \cdot |S_{add}| + \gamma \cdot |S_{del}|$. A simple cost model that gives equal weight to supernodes, superedges and corrections was used in [23] and was used to produce Figure 4.

GS has time complexity $O(d_{av}^3 \cdot (d_{av} + log(n) + log(d_{av})))$, where d_{av} is the average degree of the nodes [23]. The average degree in our datasets is low so average running time is low.

3.3 Prediction function

A **prediction function** is a function $p : e \mapsto s \in [0, 1]$ that maps each potential edge e of a TAG to a real value between 0 and 1. This value s is called **prediction score**. The function p can be used for ranking all possible edges according to their probability. Consider the graph summary $\Sigma(G)$; let s_{AC} and s_B be the corresponding super nodes of e. Note that this does not imply the existence of an super edge between s_{AC} and s_B. The prediction score for an edge $e \in ((A \cup C) \times B) - (X \cup Y)$ is defined as $p(e) = s(e) \cdot c(e)$ and combines a so-called **supernode factor** $s(e)$ and a **correction factor** $c(e)$. The supernode factor is defined as follows:

$$s(e) = \begin{cases} 1 - \frac{|s_{AC} \times s_B \cap S_{del}|}{|s_{AC}| \cdot |s_B|} & \text{if } e \in S_{del} \\ \\ \frac{|s_{AC} \times s_B \cap S_{add}|}{|s_{AC}| \cdot |s_B|} & \text{otherwise} \end{cases}$$

For $e \in S_{del}$ the graph summary contains a super edge between s_{AC} and s_B. The supernode factor determines the fraction of missing edges between the two super nodes. The larger the super nodes and the smaller the number of deletions

are, the higher is the supernode factor. On the other hand, $e \notin S_{del}$ implies that there is no super edge between s_{AC} and s_B. The supernode factor then reflects the fraction of additions in all possible edges between these two supernodes. The larger the super nodes and the smaller the number of additions are, the lower is the supernode factor. The correction factor for an edge $e = (a, b)$ is as follows:

$$c(e) = \frac{1}{1 + |S_{corr}(a)|} \cdot \frac{1}{1 + |S_{corr}(b)|}$$

Here $S_{corr}(a)$ and $S_{corr}(b)$ describe the set of corrections involving a and b, respectively, i.e., $S_{corr}(a) = \{b' | b' \neq b \wedge (a, b') \in S_{del} \cup S_{add}\}$ and $S_{corr}(b) = \{a' | a' \neq a \wedge (a', b) \in S_{del} \cup S_{add}\}$. The correction factor accounts for the number of corrections that are relevant to a given edge. The higher the number of corrections, the smaller the correction factor, and thus, the prediction score.

Figure 5 shows the relevant part of the example summarization of Figure 4 for potential edges (PO_20030, CIB5) and (GO_5773, CIB5). The deletion (PO_20030, CIB5) is the only deletion between the two supernodes and the size of the supernodes are 4 and 6, respectively. For (GO_5773, CIB5) there are two additions between the corresponding supernodes of size 1 and 6, respectively. The supernode factors are therefore calculated as follows: $s(\text{PO\_20030, CIB5}) = 1 - \frac{1}{4 \cdot 6} = \frac{23}{24}$ and $s(\text{GO\_5773, CIB5}) = \frac{2}{1 \cdot 6} = \frac{1}{3}$. The correction factors for the two example edges are: $c(\text{PO\_20030, CIB5}) = \frac{1}{1+0} \cdot \frac{1}{1+0} = 1$ and $c(\text{GO\_5773, CIB5}) = \frac{1}{1+2} \cdot \frac{1}{1+1} = \frac{1}{6}$. Finally, the overall prediction scores are: $p(\text{PO\_20030, CIB5}) = \frac{23}{24} \cdot 1 \approx 0.96$ and $p(\text{GO\_5773, CIB5}) = \frac{1}{3} \cdot \frac{1}{6} \approx 0.06$. In other words, the edge (PO_20030, CIB5) seems to be a good prediction whereas edge (GO_5773, CIB5) does not.

4 Experimental Evaluation

4.1 Dataset Preparation

The Arabidopsis Information Resource (TAIR) consists of *Arabidopsis thaliana* genes and their annotations with terms in the Gene Ontology (GO) and Plant Ontology (PO). The entire TAIR dataset includes 34,515 genes, with 201,185 annotations to 4,005 GO terms and 529,722 annotations to 370 PO terms circa October 2010. We created three subsets labeled ds1, ds2 and ds3, respectively. Each dataset was constructed by choosing 10 functionally related genes associated with photomorphogenesis, flowering time and photosynthesis, respectively, and expanding the graph to include all GO and PO terms. The statistics of these 3 dataset are shown in Table 6. Recall that we use the *shortest path distance* between a pair of CV terms as a proxy for relatedness. To test the distance restriction we create subgraphs ds1-DSG, etc. The impact of the distance restriction will be discussed in a later section.

4.2 Evaluation Methodology

We use a simple leave-K-out strategy to evaluate our link prediction approach. Given a dataset, we remove 1 (up to K) edges that are selected at random from the set of all edges. We then predict 1 (up to K) edges.

Subset	ds1	ds2	ds3	ds1+ DSG	ds2+ DSG	ds3+ DSG
Genes	10	10	10	7	10	10
GO Terms	68	44	28	14	4	8
PO Terms	44	53	48	22	11	24
Total Nodes	122	108	86	43	25	42
Annotations	395	355	426	159	123	246
Density	3.24	3.29	4.95	3.70	4.92	5.86

Fig. 6. Statistics of the 3 datasets along with their dense subgraphs.

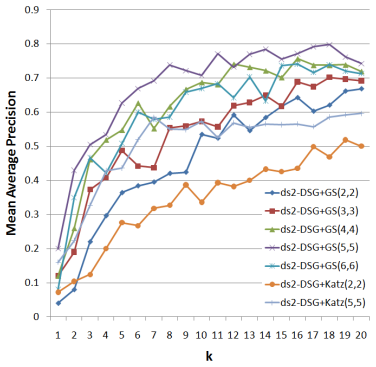


Fig. 7. MAP of predicting k annotations in ds2 dense subgraphs. Distance restrictions are (GO Distance, PO Distance).

We report on precision. We consider precision at the Top 1 or P@1 when we predict 1 edge and mean average precision (MAP) when we predict K edges [19].

To further study the quality of our prediction, we report on the scores produced by our scoring function. For those predictions in which we have the *highest confidence*, i.e., those predictions are consistently above a threshold of the scoring function, we report on the true positives (TP) and false positives (FP). A TP is a correct prediction while a FP is an incorrect prediction.

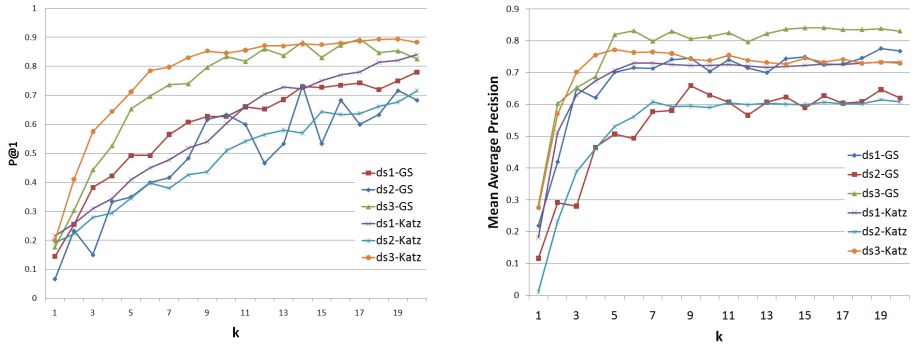
As a baseline, we compute the Katz metric between any 2 pair of nodes [12]. The Katz metric is a path based measure equal to $\sum_{l=1}^{\infty} \beta^{\ell} \cdot paths_{\ell}(x, y)$, where $paths_{\ell}(x, y)$ is the number of paths between nodes x and y of length ℓ. For our experiments, we used $\beta = .005$. All potential edges were ranked and sorted by the value of the Katz metric, creating a ranked list of predictions. This is labeled `dsi-Katz` or `dsi-DSG-Katz` where the prefix `dsi` identifies the dataset.

Three following variations of our prediction approach were considered:

- `dsi-GS`: The prefix represents the dataset and the suffix indicates that there was no DSG created and we only used graph summarization.
- `dsi:DSG+GS`: We created a DSG with no distance restrictions.
- `dsi:DSG+GS(dP,dG)`: We created a DSG with a distance restriction of `dP` for PO and a distance restriction of `dG` for GO.

We note that the DSG with no distance restriction results in the densest subgraph. Imposing a distance restriction may result in a less dense subgraph, but possibly one with greater biological meaning. The cost model for graph summarization is another experimental parameter, but one that we did not vary in our experiments. Equal weights were given to supernodes, superedges and corrections throughout all of our summarizations.

AraNet [17] created an extensive functional gene network for Arabidopsis exploiting pairwise gene-gene linkages from 24 diverse datasets representing $>$ 50 million observations. They report on prediction accuracy of GO *biological*



(a) Precision of predicting 1 annotation (b) MAP of predicting k annotations

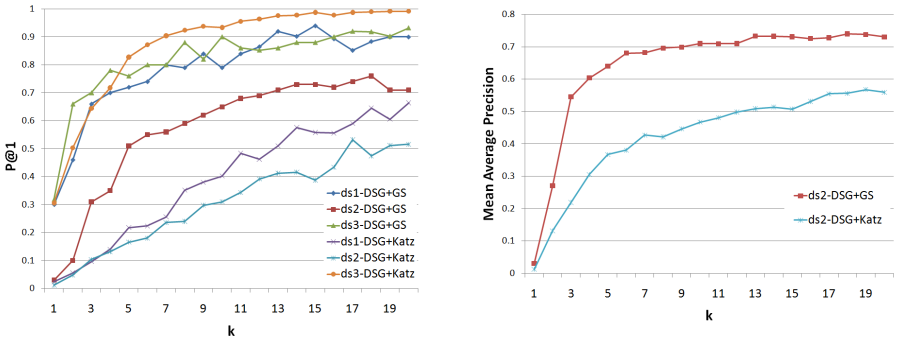
Fig. 8. Evaluation using different approaches across all datasets.

process CV terms for over 27,000 genes. Their prediction method computes a score for each gene and association using its neighborhood and naive Bayes estimation; this is similar in spirit to the Katz metric. Their results demonstrate that for over 55% of gene annotation, their predictions (cumulative likelihood ratio) were more significant compared to random prediction. A direct comparison of our approach with AraNet was not possible since AraNet exploits significant knowledge beyond GO and PO annotations. We note that the mean average precision (MAP) for our method and Katz reflect that the prediction accuracies of all three methods appear to occur in a similar range; this is notable since Katz and our method exploit only PO and GO annotation data.

4.3 Summary of Results

Baseline Analysis Given a ranked list of predictions, precision at one (P@1) provides a useful metric for evaluating the performance of the different approaches. To establish a baseline, Figure 8(a) reports on P@1 for the 3 datasets for `dsi-GS` and `dsi-Katz`. The P@1 values are low for lower K values and increase with higher K. This is expected since larger K provides a larger ground truth and improves prediction accuracy. Both methods perform best on `ds3` and show the worst prediction accuracy on `ds2`. A visual examination of the datasets and the graph summary intuitively illustrates the difference in performance across the 3 datasets. For example, `ds3` is the most dense dataset.

To complete the baseline analysis, we consider the Top K predictions as we leave out K. Figure 8(b) reports on the mean average precision (MAP) of the different approaches as a function of K. As expected MAP for Top K is higher than the values for P@1 since we are making K predictions (and not 1 prediction as before). We note that `ds3-GS` outperforms `ds3-Katz`. Both methods show the least prediction accuracy for `ds2`.



(a) Precision at 1 after removing k edges from a subgraph

(b) MAP of k annotations after removing k edges from a subgraph

Fig. 9. Comparison of our graph summarization approach with the Katz metric.

Impact of Varying the Distance Restriction The average distance between a pair of GO CV terms in ds2 is 5.57. Of the 946 pairs, 402 are within distance 5 of each other; this is the distance restriction used in our previous experiments. 160 pairs are not connected at all, i.e., they are in different parts of the ontology.

Figure 7 reports on MAP for `dsi-DSG+GS` and `dsi-DSG+Katz` for varying PO and GO restrictions on dataset `ds2`. Method `dsi-DSG+GS` dominates `dsi-DSG+Katz` *over all distance restrictions*. This is a very strong validation of the prediction accuracy of our approach. Accuracy initially increases with increasing (PO,GO) distance. The best accuracy was obtained with (5,5) after which accuracy decreases, e.g., for a (6,6) distance restriction. Figure 9(a) reports on P@1 for `dsi-DSG+GS` and `dsi-DSG+Katz` for the 3 datasets with distance (5,5). Method `dsi-DSG+GS` dominates `dsi-DSG+Katz` for `ds1` and `ds2`. Surprisingly `ds3-DSG+Katz` outperforms `ds3-DSG+GS` for `ds3`. An examination of the predictions indicates that `ds3-DSG+GS` makes an incorrect prediction which has a high prediction score and is therefore ranked high. Since Figure 9(a) reports on P@1, this has a significant penalty on the accuracy of `ds3-DSG+GS`.

Figure 9(b) reports on the mean average precision (MAP) for `ds2` with distance restriction (5,5). Again, `ds3-DSG+GS` outperforms `ds3-DSG+Katz`, further confirming the strength of our approach.

Confidence in Predictions Our final experiment is to validate that high confidence predictions result in more accurate predictions. High confidence predictions are those that receive a high prediction score. Table 1 reports on the percentage of true positive (TP) and false positives (FP) for `ds2-Katz`, `ds2-GS` and `ds2-DSG+GS`, bucketized by the range of prediction score. Note that for Katz, we normalize the score from 0.0 to 1.0 prior to bucketization. The values on the left represent the high confidence (high score) prediction buckets and the confidence (score) decreases as we move to the right. As expected, the % of TP values

Score	<1.0	<.90	<.80	<.70	<.60	<.50	<.40	<.30	<.20	<.10
TP `ds2-Katz`	50	51	77	18	11	4	1	3	6	0
FP `ds2-Katz`	50	49	23	82	89	96	99	97	94	100
TP `ds2-GS`	50	42	35	37	38	40	18	3	1	1
FP `ds2-GS`	50	58	65	63	62	60	82	97	99	99
TP `ds2-DSG+GS`	80	89	61	61	56	51	18	18	18	25
FP `ds2-DSG+GS`	20	11	39	39	44	49	82	82	82	75

Table 1. Percentage of true positives (TP) and false positives (FP) as a function of the prediction score for predictions made on ds2 with DSG+GS(5,5), GS, and Katz

is greater than the % of FP values for high confidence buckets. The reverse is true for low confidence buckets. This holds for all the methods. Further, the % TP values for `ds2-DSG+GS` for the 2 left most buckets, 80% and 89%, dominates the % TP values of `ds2-GS` (50% and 42%) and `ds2-Katz` (50% and 51%). The % TP values for `ds2-DSG+GS` is overall higher than the other two methods except for one exception (score between 0.7 to 0.8). These results confirm that `ds2-DSG+GS` had both higher confidence scores and higher prediction accuracy, compared to `ds2-GS` and `ds2-Katz`. This held across all 3 datasets and further validates our prediction approach.

5 Conclusions and Future Work

We presented a novel approach for link prediction in the layered annotation graph datasets that employs graph summarization for link prediction. Furthermore, the complementary method of identifying dense subgraphs helps find interesting regions for high quality predictions. To the best of our knowledge, we are the first to consider the synergy of these two approaches. Future work includes learning GS cost models using supervised learning.

References

1. N. Benchettara, R. Kanawati, and C. Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In *Proc. ASONAM*, pages 326–330, 2010.
2. C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
3. P. Bogdanov and A. K. Singh. Molecular Function Prediction Using Neighborhood Features. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 7(2):208–217, 2010.
4. M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*, pages 84–95, 2000.
5. H. N. Chua, W.-K. Sung, and L. Wong. An efficient strategy for extensive integration of diverse biological data for protein function prediction. *Bioinformatics*, 23(24):3364–3373, 2007.
6. A. V. Goldberg. Finding a maximum density subgraph. Technical Report UCB/CSD-84-171, EECS Department, University of California, Berkeley, 1984.

7. M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link Prediction Using Supervised Learning. In *Proc. on Link Analysis, Counterterrorism and Security*, 2006.
8. O. Hassanzadeh and et al. Linkedct: A linked data space for clinical trials. In *Proc. WWW 2009 Workshop on Linked Data on the Web (LDOW2009)*, 2009.
9. P. Jain, P. Yeh, K. Verma, R. Vasquez, M. Damova, P. Hitzler, and A. Sheth. Contextual ontology alignment of lod with an upper ontology: A case study with proton. In *Proc. ESWC*, pages 80–92, 2011.
10. J. Kahan and M. Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *Proc. of the WWW*, pages 623–632, 2001.
11. B. Kang, N. Grancher, V. Koyffmann, D. Lardemer, S. Burney, and M. Ahmad. Multiple interactions between cryptochrome and phototropin blue-light signalling pathways in arabidopsis thaliana. *Planta*, 227(5):1091–1099, 2008.
12. L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–40, 1953.
13. S. Khuller and B. Saha. On Finding Dense Subgraphs. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 597–608, 2009.
14. G. Kortsarz and D. Peleg. Generating sparse 2-spanners. *J. Algorithms*, 17(2):222–236, 1994.
15. J. Kunegis, E. D. Luca, and S. Albayrak. The link prediction problem in bipartite networks. In *Proc. IPMU*, pages 380–389, 2010.
16. E. Lawler. *Combinatorial optimization - networks and matroids*. Holt, Rinehart and Winston, New York, 1976.
17. I. Lee, B. Ambaru, P. Thakkar, E. Marcotte, and S. Rhee. Rational association of genes with traits using a genome-scale gene network for arabidopsis thaliana. In *Nature Biotechnology*, number 28, pages 149–156, 2010.
18. D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(7):1019–1031, 2007.
19. C. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
20. S. Mir, S. Staab, and I. Rojas. An unsupervised approach for acquiring ontologies and rdf data from online life science databases. In *Proc. of ESWC (2)*, pages 319–333, 2010.
21. G. M. Namata, H. Sharara, and L. Getoor. A Survey of Link Mining Tasks for Analyzing Noisy an Incomplete Networks. In J. H. Philip S. S. Yu and C. Faloutsos, editors, *Link Mining: Models, Algorithms, and Applications*. Springer, 2010.
22. S. Navlakha and C. Kingsford. Exploring biological network dynamics with ensembles of graph partitions. In *Proc. 15th Intl. Pacific Symposium on Biocomputing (PSB)*, volume 15, pages 166–177, 2010.
23. S. Navlakha, R. Rastogi, and N. Shrivastava. Graph summarization with bounded error. In *Proc. of Conference on Management of Data (SIGMOD)*, 2008.
24. M. Ohgishi, K. Saji, K. Okada, and T. Sakai. Functional analysis of each blue light receptor, cry1, cry2, phot1, and phot2, by using combinatorial multiple mutants in arabidopsis. *Proc. of the National Academy of Sciences*, 1010(8):2223–2228, 2004.
25. R. Parundekar, C. Knoblock, and J. Ambite. Linking and building ontologies of linked data. In *Proc. of the ISWC*, pages 598–614, 2010.
26. M. Pujari and R. Kanawati. A supervised machine learning link prediction approach for tag recommendation. In *Proc. of HCI*, 2011.
27. B. Saha, A. Hoch, S. Khuller, L. Raschid, and X.-N. Zhang. Dense subgraphs with restrictions and applications to gene annotation graphs. In *Conference on Research on Computational Molecular Biology (RECOMB)*, 2010.