# Retrieval of Relevant and Non-redundant Nodes*

Laura Langohr[†] Hannu Toivonen[†]

## Abstract

We discuss the problem of discovering interesting nodes in networks. We adapt a generic model to choosing relevant and non-redundant pieces of information in networks and probabilistic relations. In the model we assume that one or more query nodes have been given, and the problem is to identify other nodes that are relevant with respect to the query nodes but non-redundant with respect to each other. Also, negative query nodes can be specified. This is in contrast with mainstream graph mining, where one typically looks for frequent patterns, not for interesting individuals.

We consider two instances of the model: one where node proximity (and relevance) is measured by the shortest path, and one where the graph is probabilistic or uncertain and proximity reflects the probability that the nodes are connected. The generic model also has simple parameterization to allow for different behaviors with respect to query nodes.

We compare different similarity measures and empirically evaluate two algorithms on different applications: social networks and biomedical networks. The results indicate that the model and methods are useful in finding a relevant and non-redundant set of nodes.

**Keywords:** Relevance, Non-redundancy, Graphs, Knowledge Retrieval

## 1 Introduction

Information is often modeled as a network (or graph) of objects (or concepts): think of social networks, biological networks, traffic networks, or mind maps, for instance. We address the problem of discovering interesting nodes in networks. This is in contrast with mainstream graph mining and network analysis, where one typically looks for frequent patterns or community structures.

In the settings that we consider, one or more query nodes have been given, and the problem is to identify other nodes that are relevant with respect to the query nodes but non-redundant with respect to each other. Also, negative query nodes can be specified. Consider the following examples.

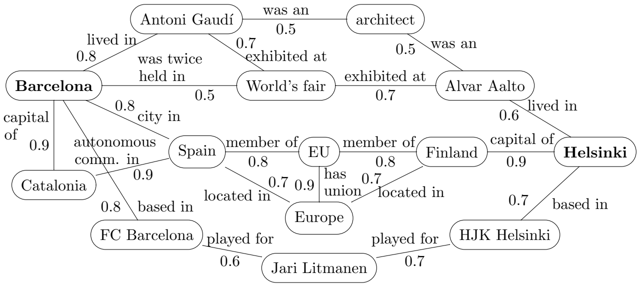


Figure 1: An example network with two positive query nodes *Barcelona* and *Helsinki* (in bold) and related concepts.

In *bioinformatics* a common problem is that high-throughput techniques associate several genes with a disease or trait. A lot of non-trivial biological knowledge can be represented as a network in which nodes represent biological entities (e.g., genes, proteins, or pathways) and edges represent relations between them (e.g., a gene codes a protein, or a protein is active in a pathway). Now, finding biological processes or pathways (nodes in the network) that are relevant both to the disease and the given genes (the query nodes) helps to understand how they are related and may help identify possible shared biological mechanisms.

Negative query nodes specify regions to be avoided, and are useful in preventing unwanted results in the output. In the biological example, the user might specify well-known positive results as well as known negative results as negative query nodes, and thus guide the mining process to more interesting results.

On the other hand, if the nodes returned as results are closely related to each other, the result is probably not as interesting as it could be. A more varied result of the same size is probably more useful, as it can represent several different hypothesis about the possible biological relationship. This is addressed by the requirement that the resulting nodes are mutually non-redundant.

Further on, if the relevant and non-redundant nodes are ranked, a scientist could start to study them from the top, and decide for herself when the relevance becomes too low or the cost of further studies too high.

In *semantic* or *word association networks*, nodes

[*] This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland and by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898. We would like to thank Petteri Hintsanen and Atte Hinkka for the DBLP implementation, and Lauri Eronen for his help with the biomedicine test data as well as for the Biomine and RWR implementations.

[†] Department of Computer Science and Helsinki Institute for Information Technology HIIT, University of Helsinki, Finland. {laura.langohr,hannu.toivonen}@cs.helsinki.fi

represent concepts and edges their semantic, associative, or co-occurrence relations. Consider, as a toy example, the network in Figure 1. A user who wants to know how *Barcelona* and *Helsinki* are related might already know that *Barcelona* is a city in *Spain*, that *Helsinki* is the capital of *Finland*, and that both *Spain* and *Finland* are in *Europe* and also are members of the European Union (*EU*). Other, perhaps less obvious relations, can be more interesting. For example, a user might have not known that *architect*s *Antoni Gaudí* (who lived in *Barcelona*) and *Alvar Aalto* (who lived in *Helsinki*) both have exhibited at a *world's fair* (also known as Expo). Another user again might not know that soccer player *Jari Litmanen* has played for *FC Barcelona* as well as for *HJK Helsinki*.

Given *Barcelona* and *Helsinki* as query nodes, the goal is to identify a non-redundant set of nodes relevant to both cities. In the small example network of Figure 1, *EU* is highly relevant to both and therefore the first choice to be included in the result. *Europe* is highly relevant, too, but being closely related to *EU* it would be a redundant choice. Non-redundant but still relevant concepts include *world's fair* and *Jari Litmanen*.

Note that relevance with respect to query nodes is highest for nodes between the query nodes. A relevant node typically is a central node for the (indirect) relation between the query nodes. A non-redundant set of relevant nodes (such as *EU*, *world's fair*, and *Jari Litmanen*) then highlights distinct relations or contexts between the query nodes.

Relevance and redundancy have been addressed in several contexts before, in particular in information retrieval and web search. A recent paper proposes a general approach to find relevant and non-redundant objects based on object similarity or distance alone [16]. In the current paper, we extend that proposal in two significant ways: (1) to networks, for finding interesting and non-redundant nodes, and (2) to probabilistic relations between objects, and in combination with extension 1 to probabilistic or uncertain networks.

It is interesting to note how the components of this approach match the typical definition of data mining as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [10]. Our relevance measures utilize paths in networks, i.e., implicit relations between query nodes and the results nodes. So, while the results are explicit and simple — just a set of nodes — they represent implicit information. Negative query nodes are a tool to help the user find previously unknown information. That is, if the user specifies information about the query nodes that she already knows as negative query nodes, then our approach finds previously unknown information. While usefulness of results is difficult to measure outside the context of any particular application, it seems obvious that the simplicity of the results makes them easy to understand and that non-redundancy increases the likelihood of their usefulness as a whole.

This paper is structured as follows. Section 2 reviews the approach of [16] and briefly reviews other related work. We extend the approach to find relevant and non-redundant nodes in networks in Section 3. In the same section, we also propose alternative measures for relevance and non-redundancy with a probabilistic interpretation, and apply these to the network setting. In Section 4 we carry out experiments on article coauthorships networks and on biological networks. Section 5 concludes the paper.

## 2 Background

### 2.1 A model for relevance and non-redundancy.

A relatively simple yet powerful model for choosing relevant and non-redundant pieces of information has been proposed recently [16]. We here briefly review the model and algorithms of [16], to make this paper self-contained. The following sections then contain the original contributions of this paper.

The relevance model of [16] addresses settings where relevance of objects is measured with respect to a set of (positive) query objects and a set of negative query objects, and the resulting objects should also be mutually non-redundant.

The model assumes a similarity function $s : V \times V \to \mathbb{R}^+$ or distance function $d : V \times V \to \mathbb{R}^+$ on a set of objects $V$. It assumes that either one is given and identifies the other one simply with the inverse $s(u, v) = 1/d(u, v)$ for all $u, v \in V$, with the exception that $s(u, u) = \infty$.

**Relevance.** The *relevance* of an object $u \in V$ with respect to a positive query object $q \in V$ is defined directly as their proximity:

$$rel_P(u, q) = s(u, q) = 1/d(u, q). \tag{2.1}$$

Given a set $Q_P \subset V$ of (positive) query objects, an object is considered to be more relevant if it is relevant to all query objects. The relevance of object $u$ with respect to a set $Q_P$ of query objects is then defined as the inverse of the p-norm:

$$rel_P(u, Q_P) = (\sum_{q \in Q_P} d(u, q)^\alpha)^{-\frac{1}{\alpha}} \tag{2.2}$$

where $\alpha \geq 1$. Equation 2.1 is a special case of this definition when $Q_P = \{q\}$. The relevance is monotone decreasing in the distance to each query object (with the exception of $\alpha = \infty$ when it is a function of the largest distance alone).

**Irrelevance.** Negative query objects allow specification of subjective irrelevance (uninterestingness) of objects. The (negative) relevance of object $u$ with respect to a single negative query object $\bar{q}$ is measured with the given similarity or distance function, just like relevance to a single positive query object:

$$rel_N(u, \bar{q}) = s(u, \bar{q}) = 1/d(u, \bar{q}). \tag{2.3}$$

A negative query object's contribution is then $-rel_N(u, \bar{q})$.

A set $Q_N \subset V$ of negative query objects are treated more as a disjunction: the result is irrelevant if it is close to *any* negative query object. Thus, negative relevance of object $u$ with respect to a set $Q_N \subset V$ of negative query objects is defined as

$$rel_N(u, Q_N) = \sum_{\bar{q} \in Q_N} d(u, \bar{q})^{-\beta} = \sum_{\bar{q} \in Q_N} s(u, \bar{q})^{\beta}, \tag{2.4}$$

where $\beta > 1$. Their contribution to overall relevance is then $-rel_N(u, Q_N)$. That is, the overall relevance of a node is the relevance with respect to positive query nodes (Equation 2.2) minus the irrelevance with respect to negative query nodes (Equation 2.4). The irrelevance function is zero if there are no negative query objects, the effect of a negative query point infinitely far away is zero, and the function is monotonically decreasing in each distance.

**Relevance vs. irrelevance.** Given positive and negative query objects (sets $Q_P$ and $Q_N$, respectively), the total relevance of object $u$ is defined as

$$REL(u, Q_P, Q_N) = rel_p(u, Q_P) - rel_N(u, Q_N). \tag{2.5}$$

It favors objects that are centered between the positive query objects and that are not close to any negative one. Given a single positive and single negative query point, it simply measures which one is closer.

**Non-redundancy.** The most relevant objects, as defined above, can be close neighbors. To retrieve mutually non-redundant or complementary objects, redundancy is defined in a similar way like negative relevance.

The *redundancy* of a set $R \subseteq V$ of objects is defined by

$$red(R) = \sum_{\substack{u,v \in R \\ u \neq v}} d(u, v)^{-\beta} = \sum_{\substack{u,v \in R \\ u \neq v}} s(u, v)^{\beta}, \tag{2.6}$$

where $\beta \geq 1$. Redundancy will also contribute negatively to the overall relevance of a set, that is, by $-red(R)$.

**A relevant and non-redundant set of objects.** The overall goal is to find a diverse set of relevant objects according to the user's query. The overall relevance and non-redundancy of a set of (retrieved) objects $R \subseteq V$ is defined as

$$\begin{aligned} REL(R, Q_P, Q_N) \quad = \quad & \textstyle\sum_{u \in R} rel_P(u, Q_P) \\ & - \textstyle\sum_{u \in R} rel_N(u, Q_N) \\ & - red(R). \end{aligned} \tag{2.7}$$

We can now present the problem formally. In addition to the positive and negative query objects, assume the user also specifies the number $k$ of objects in the output, and possibly (for practical convenience) a set $V' \subseteq V$ of objects among which to select the output.

**Problem definition.** Given a set $V$ of objects, a distance function $d(u, v)$ or a proximity function $s(u, v)$ for objects $u, v \in V$, a set $Q_P \subset V$ of positive query objects, a set $Q_N \subset V$ of negative query objects, a target set $V' \subseteq V$, and an integer $k$, the *problem of retrieving a relevant and non-redundant set of objects* is to identify a set $R \subseteq V'$ of size $|R| = k$ that maximizes $REL(R, Q_P, Q_N)$.

The overall relevance $REL(\cdot)$ (Equation 2.7) is submodular [16]. For a submodular function, a greedy algorithm is guaranteed to find a set which achieves at least $1/k$ of the optimal score [19]. We next review two algorithms given in [16], a greedy and an iterative one.

**Greedy algorithm.** The greedy algorithm produces a ranked list of objects in an incremental, greedy fashion with respect to the overall relevance $REL(u, Q_P, Q_N)$. In each iteration, it finds the currently most relevant object and outputs it.

**Greedy algorithm**

```
1. Repeat until a sufficient number of objects
   has been retrieved:
   1.1 Find the most relevant object $r$
       w.r.t. $Q_P$ and $Q_N$.
   1.2 Output $r$ and add it to $Q_N$.
```

The greedy algorithm is made simple by the observation that – thanks to the uniformity of negative query nodes and non-redundancy – non-redundancy can be achieved by adding nodes to the set of negative query nodes as they are output [16].

**Iterative algorithm.** The iterative algorithm below produces a non-redundant set of $k$ relevant objects, where $k$ is given as a parameter. The algorithm takes $k$ initial objects as input (or chooses them by random), and then iteratively improves the solution. In each iteration, the algorithm takes one of the $k$ objects and replaces it by the optimal one, given the $k-1$ other current objects. When no improvements can be achieved, the algorithm stops.

**Iterative algorithm**

```
1. Get an initial solution $R$ of $k$ objects
   (e.g. random)
2. Repeat while $R$ changes:
   2.1 Find the optimal swap of any object $r$ in $R$
       to any object not in $R$.
   2.2 If the swap improves the result, implement it.
```

Two variants of the iterative algorithm were proposed: (1) Run the greedy algorithm first (for $k$ iterations at least) and then use the top $k$ objects from it as the initial solution to the iterative algorithm. (2) Give $k$ random objects as the initial solution. Optionally run the iterative algorithm several times with different random seeds and choose the result that maximizes $REL(R, Q_P, Q_N)$.

**2.2 Other related work.** Identifying a set of relevant objects (typically documents) is a classical problem in information retrieval (IR). In typical settings, the selection is primarily based on the information contents of objects. Our problem differs from the main body of IR literature in the following aspects. (1) In our work the objects (nodes) are not assumed to have attributes or other content. (2) Relevance is based solely on node proximity in a network. (3) Queries are specified as nodes themselves, not by keywords.

Negative query terms and redundancy are well-known in IR (see, e.g., [11, 15, 30]). Incremental retrieval methods are widely used (see, e.g., [3, 5, 24]). The problem of finding non-redundant or representative objects has been addressed in numerous other contexts, too (see, e.g., [17, 21, 31]). However, all these method address non-redundancy only indirectly, and relevance to query terms not at all. See [16] for a review of more related work in general.

We have special interests on finding relevant nodes in networks. Relevance and proximity can be modeled in numerous ways, e.g., using the length of the shortest path, network flow, effective conductance [14], and random walk models (see, e.g., [9, 20, 26]). A variant of random walk also considers positive and negative query nodes [29]. In this approach the original graph structure is refined to take negative query nodes into account, whereas in our approach we do not change the graph structure. There also is a variant that diversifies the ranking by reinforcing the probability to stay at a node by the number of visits to the node [18]. Again, non-redundancy is addressed only indirectly.

Recently, an interesting variant of random walk with restart was proposed to directly address non-redundancy [28]. Relevance is measured by the personalized PageRank, and non-redundancy (called diversity) by a personalized adjacency matrix, which is biased towards the query vector and weighted by the personalized PageRank vector. A greedy, iterative algorithm finds such a ranking in $O(|E| + |V|^2)$ time, or $O(|E| + |V|k)$ if only the top $k$ nodes are ranked. These relevance and non-redundancy measures could be alternatively used instead of the ones we use here. However, negative query nodes and irrelevance are not considered.

Probabilistic relevance and proximity in networks has been considered, for example, to find a relevant subnetwork [12, 8]. In semantic networks, a related approach is spreading activation between nodes [2, 1]. As such, all these methods address non-redundancy only indirectly. They could potentially be used as proximity (relevance) functions in our model, but their computational complexity would likely be a problem.

The model proposed in [16] and its adaption to networks proposed here differ from the previous work by providing relatively simple but very flexible and generic measures for finding relevant but non-redundant nodes. Our model only relies on a distance or proximity function between nodes, and does not assume any other contents or properties for the individual nodes. Potential targets thus range from mining or retrieving atomic concepts to complex structures which can be represented as nodes and their relationships as edges in a network.

## 3 Relevant and non-redundant nodes in networks

We are interested in data represented as a network, and in finding relevant and non-redundant nodes. We next adapt the model of [16] to networks. In order to identify relevant and non-redundant nodes, we first formalize the concepts of relevance with respect to given query nodes, irrelevance with respect to given (negative) query nodes, and mutual non-redundancy between nodes. We then show how they can also be applied to probabilistic relations in networks.

**3.1 Standard networks.** Let us start by discussing some desirable properties of relevance and non-redundancy in networks, using the network of Figure 1 as an example.

Obviously, proximity to a query node indicates relevance with respect to it. In the case of multiple query nodes, the most relevant results are somewhere between the query nodes. For instance, if *Barcelona* and *Helsinki* are the query nodes, then *EU*, *Europe*, and *World's fair* are relevant results since they are relatively well connected to both query terms.

On the other hand, if there are several negative query nodes, each one's neighborhood is to be avoided. Given *Barcelona* and *Helsinki* as negative query nodes, *Catalonia* and *Finland* are quite irrelevant, but *EU* is not.

These desiderata indicate that some distance measure, such as the shortest path length between nodes, is a suitable distance function. A bit more formally, assume we have an undirected, weighted network where edge weights are positive and where $w(u, v)$ can be interpreted as a distance between nodes $u$ and $v$. For example, edge weights can be boolean ($w(u, v) = 1$ if an edge between $u$ and $v$ exists, and $w(u, v) = 0$ else) or they can represent some domain specific distance. The distance between two nodes can be defined as the length $len(sp(u, v))$ of the shortest path $sp(u, v)$ between them:

$$d(u,v) = \begin{cases} 0 & \text{if } u = v \\ len(sp(u,v)) & \text{if } u \neq v \text{ and they} \\ & \text{are connected} \\ \infty & \text{else.} \end{cases} \tag{3.8}$$

Clearly, function $d(\cdot)$ is a metric: it is symmetric, satisfies triangle inequality and is positive definite. Using it, we can directly apply the relevance, irrelevance, and non-redundancy functions and algorithms of Section 2.1 and [16] to retrieve relevant and non-redundant nodes.

In addition, one can also consider more complex proximity measures, such as maximum flow or random walk based models, that are not limited to just the best path. Random walk with restart (RWR) provides a relevance score between two nodes in a weighted graph [27]. The standard random walk starts from a node $u$ and then iteratively moves from a node to a neighboring one. The probability of choosing any particular edge to follow (transition probability) is proportional to the edge weight [22]. In a random walk with restart, the random walker will at each step return to the original node $u$ with some probability. The relevance score of nodes $u, v$ can then be defined as the steady-state probability that the random walker is at node $v$.

**Algorithms.** Fast solutions exist for computing RWR [27]. However, computing the distance function of Equation 3.8 requires finding best paths between nodes. All-pairs best paths can be computed in time $O(|V|(|E| + |V|) \log |V|)$ and this may be prohibitive on large networks.

We propose the following simple optimization to reduce run time. It does not change the worst case complexity, but can give a practical advantage.

Recall that the overall relevance and non-redundancy of a node $u$ depends on its distance to all positive query nodes, to all negative query nodes, and to all other nodes in the output. However, a reasonably good approximation can be obtained without computing all of these. As argued above, relevance with respect to positive query nodes should usually depend on all of them. However, irrelevancy with respect to negative query nodes is usually mostly dependent on the nearest negative query node.

We thus propose to use the maximum inverse distance or maximum similarity

$$rel_N(u, Q_N) = \max_{\bar{q} \in Q_N} d(u, \bar{q})^{-\beta} = \max_{\bar{q} \in Q_N} s(u, \bar{q})^{\beta} \tag{3.9}$$

instead of the sum of similarities (or inverse distances). Clearly, $rel_N(\cdot)$ of Equation 3.9 is a lower bound of the respective value of Equation 2.4, and it is the highest lower bound we can obtain using just one negative query node.

**3.2 Probabilistic relations in networks.** Consider now a situation where the similarity of two objects is measured by a probability, such as the probability that the objects are related or linked. We are particularly interested in probabilistic networks describing such uncertain relations. Such settings also arise in probabilistic or uncertain databases [4, 7].

Assume that we are given probabilities $p(u, v)$ for all pairs of objects $u$ and $v$. The probability is then a natural similarity and relevance measure, but limited to the range $[0, 1]$.

A natural relevance measure with respect to a set of positive query objects then is the probability that a given objects $u$ is related to all of the query objects:

$$prob_P(u, Q_P) = \prod_{q \in Q_P} p(u, q). \tag{3.10}$$

It can be shown that this is essentially a special case of the relevance function $rel_P(u, Q_P)$ of Equation 2.2. Use function $d(u, v) = -\log(p(u, v))$ to map probabilities to distances, and set $\alpha = 1$. Then

$$\begin{aligned} rel_P(u, Q_P) &= \sum_{q \in Q_P} \log(p(u, q)) \\ &= \log(prob_P(u, Q_P)). \end{aligned} \tag{3.11}$$

In uncertain networks where edges describe probabilistic relations between nodes, we can define $p(u, v)$ as the probability of the best path between $u$ and $v$. This is a simple but relatively efficient lower bound approximation of the probability that $u$ and $v$ are connected, a measure known as network reliability [6].

To be exact, Equation 3.10 then is approximate also for another reason: it does not take into account possible overlap in the best paths. The probabilities of any shared edges will be counted several times. This could be circumvented by considering the union of all edges, but we anticipate this additional complexity is not significant in practice.

Note that most probable paths can be reduced to shortest paths: compute the sum $\sum d(\cdot)$ of edge lengths when they are defined by $d(u,v) = -\log(p(u,v))$.

Finally, in uncertain networks, we can also use the optimized negative relevance of Equation 3.9, which translates to

$$rel_N(u, Q_N) = (-\log \max_{\bar{q} \in Q_N} p(u, \bar{q}))^{-1} \tag{3.12}$$

when $\beta = 1$.

## 4 Experiments

We will first illustrate the concepts and methods in a social network setting, analysing relations between computer scientists in a co-authorship network, looking for researchers related to *C. Faloutsos* and *J. Han.* We then study the performance of the methods on biological network data.

**4.1 Co-authorship relations.** In the first set of experiments we used a co-authorship network extracted from DBLP[1] (Digital Bibliography & Library Project) of Oct 6th, 2010.

**Test Setting.** We extracted a network of 20 authors and 45 co-authorships connecting *Christos Faloutsos* and *Jiawei Han* and used four different pairwise similarity measures:

- LEN-SP: the reciprocal of the length of the shortest path (Equation 3.8) on boolean edge weights,
- LEN-SP-RWR: Random walk with restart with transition probabilities proportional to LEN-SP,
- CUM: a similarity measure proportional to a cumulative distribution function [23] in the range $[0, 1]$, where the proximity of any two authors, especially when not co-authors, is defined using the best path between them, taking the product of pairwise similarities along the path as the final proximity, and
- CUM-RWR: Random walk with restart with transition probabilities proportional to CUM.

**Results.** The top eight relevant and non-redundant authors obtained with the greedy algorithm are shown in the left and middle column of Table 1. While the left column shows the relevant and non-redundant authors obtained with the relevance and non-redundance measure of Section 2.1, the middle shows those obtained with the probabilistic relevance and non-redundance measure of Section 3.2.

When using LEN-SP or CUM as similarity, and either relevance and non-redundancy measures, the authors obtained are all prominent researchers that are relatively closely related to Faloutsos and Han by direct or indirect co-authorship relations. In both cases, the first four of the chosen authors have never published together according to DBLP, so they are likely to represent different communities or areas relevant to Faloutsos and Han. The spread of the results is also illustrated by the fact that many of the first eight authors come from different countries.

In contrast, if redundancy is ignored and the computation is based only on relevance, a redundant set of authors is obtained, regardless of which similarity measure is used (Table 1, right column). The eight most relevant authors are highly connected to each other in the co-authorships network, and come from either the US or Canada, with a few exception. A redundant set of authors is also obtained when LEN-SP-RWR or CUM-RWR are used as similarity measure.

*Comparison of algorithms.* Let us next compare the algorithmic variants when a fixed number $k$ of nodes should be given as result. We compare four different approaches: (1) finding relevant and non-redundant nodes with the greedy algorithm and taking the top $k$ nodes, (2) finding them initially with the greedy algorithm and improving the results with the iterative algorithm, (3) picking $k$ nodes randomly initially and improving the results with the iterative algorithm, and (4) simply picking $k$ nodes randomly.

Figure 2 (a) shows the $k$th node's effect on the overall relevance (in the original probability domain). The overall relevance decreases with increasing $k$ as it is more difficult to find relevant and non-redundant nodes. However, the individual factors fluctuate as the algorithm chooses between different trade-offs of relevance and non-redundancy. The irrelevance is constant at 1.0 as no negative query node was chosen.

Figure 2 (b) shows a comparison of the algorithmic variants and random ranking. The three algorithmic variants, using the greedy and iterative algorithms, are practically indistinguishable while the random results are systematically inferior. This indicates that the result of the greedy algorithm is, in addition to being a ranking of the nodes, also a good choice for any given $k$. Another observation is that the iterative algorithm performed equally well with random initialization as it does with initial ranking obtained by the greedy algorithm.

**4.2 Biomedicine.** We used data published by Köhler et al. [13], who defined 110 disease-gene families based on the OMIM database. The families contain

[1] http://dblp.uni-trier.de/

Table 1: Eight most relevant and non-redundant authors (left and middle columns) or just relevant authors (right column) for $Q_P = \{C.\ Faloutsos,\ J.\ Han\}$.

| | Relevance and non-redundancy | | Probabilistic | | Relevance only | |
|---|---|---|---|---|---|---|
| | with $\alpha = 4$, $\beta = 2$ | | relevance and non-redundancy | | with $\alpha = 4$ | |
| LEN-SP | P.S. Yu | IL, USA | P.S. Yu | IL, USA | P.S. Yu | IL, USA |
| | R.T. Ng | Canada | R.T. Ng | Canada | R.T. Ng | Canada |
| | H.J. Zhang | China | B. Chin Ooi | Singapore | C. Liu | WA, USA |
| | B. Chin Ooi | Singapore | Y. Tao | Hong Kong | W. Fan | NY, USA |
| | Y. Tao | Hong Kong | X. He | China | T.K. Sellis | Greece |
| | C. Liu | WA, USA | W. Fan | NY, USA | L.V.S. Lakshmanan | Canada |
| | T.K. Sellis | Greece | J. Gao | IL, USA | J. Gao | IL, USA |
| | W. Fan | NY, USA | J. Yang | OH, USA | J. Yang | OH, USA |
| LEN-SP-RWR | P.S. Yu | IL, USA | P.S. Yu | IL, USA | P.S. Yu | IL, USA |
| | R.T. Ng | Canada | R.T. Ng | Canada | R.T. Ng | Canada |
| | H.V. Jagadish | MI, USA | H.V. Jagadish | MI, USA | H.V. Jagadish | MI, USA |
| | C. Liu | WA, USA | J. Pei | Canada | J. Pei | Canada |
| | J. Pei | Canada | C. Liu | WA, USA | C. Liu | WA, USA |
| | H. Tong | NY, USA | L.V.S. Lakshmanan | Canada | L.V.S. Lakshmanan | Canada |
| | L.V.S. Lakshmanan | Canada | H. Tong | NY, USA | H. Tong | NY, USA |
| | J. Gao | IL, USA | J. Gao | IL, USA | J. Gao | IL, USA |
| CUM | P.S. Yu | IL, USA | P.S. Yu | IL, USA | P.S. Yu | IL, USA |
| | R.T. Srivastava | NJ, USA | R.T. Ng | Canada | R.T. Ng | Canada |
| | H.J. Zhang | China | H.J. Zhang | China | S. Papadimitriou | NY, USA |
| | Y. Tao | Hong Kong | Y. Tao | Hong Kong | L.V.S. Lakshmanan | Canada |
| | C. Liu | WA, USA | C. Liu | WA, USA | H.V. Jagadish | MI, USA |
| | B. Chin Ooi | Singapore | J. Gao | IL, USA | X. Yan | CA, USA |
| | T.K. Sellis | Greece | B. Chin Ooi | Singapore | J. Yang | OH, USA |
| | J. Gao | IL, USA | T.K. Sellis | Greece | W. Fan | NY, USA |
| CUM-RWR | P.S. Yu | IL, USA | P.S. Yu | IL, USA | P.S. Yu | IL, USA |
| | R.T. Ng | Canada | R.T. Ng | Canada | R.T. Ng | Canada |
| | H.V. Jagadish | MI, USA | H.V. Jagadish | MI, USA | H.V. Jagadish | MI, USA |
| | H. Tong | NY, USA | L.V.S. Lakshmanan | Canada | H. Tong | NY, USA |
| | Y. Tao | Hong Kong | J. Pei | Canada | J. Pei | Canada |
| | C. Liu | WA, USA | H. Tong | NY, USA | L.V.S. Lakshmanan | Canada |
| | T.K. Sellis | Greece | C. Liu | WA, USA | S. Papadimitriou | NY, USA |
| | S. Papadimitriou | NY, USA | X. Yan | CA, USA | C. Liu | WA, USA |

three to 41 genes each; each family is related to one disease.

**Test Setting.** We randomly picked several test queries as follows. For each query, three different gene families were randomly chosen, and from each family a gene was randomly picked. Two genes were then used as positive query nodes ($Q_P$) and one as a negative one ($Q_N$).

For each query, we used Biomine [25] to obtain a biomedical network $G$ connecting the positive query nodes and to compute probabilistic proximities $p(\cdot)$. The sizes of the networks range between 61 to 130 nodes. Experiments with larger networks produce similar results.

**Results.** Let us first look at the results from a single network of 82 nodes, with two positive and one negative query node. All the other 80 nodes are to be ranked by the greedy algorithm. Figure 3 (a) shows the $k$th node's effect on the overall relevance (in the original probability domain, again). While non-redundancy starts at 1 for $k = 1$ and is high at the beginning, it eventually drops. The overall relevance decreases with increasing $k$ as it is more difficult to find relevant and non-redundant nodes. However, the individual factors fluctuate as the algorithms chooses between different trade-offs of relevance and non-redundancy.

Figure 3 (b) illustrates the relevance and non-redundancy of the the whole set $R_k$ of top $k$ nodes for 10 different networks and queries. The panels show that the quality decreases in quite a similar manner for all these different cases, but there are differences of orders of magnitude.

## 5 Conclusion

This paper is a step towards a generic approach to problems where a non-redundant set of relevant nodes should be found, given positive and negative query nodes and a distance or proximity measure. We adapted definitions of relevance, irrelevance and non-redundancy to networks and probabilistic relations of nodes. We performed experiments on a co-authorship network as well as on biomedical networks. We performed experiments with different similarity measures, based on which the length of the shortest path and a similarity measure proportional to a cumulative distribution seem to produce a good set of nodes.

We experimentally analyzed two algorithms: one that greedily ranks a given set of nodes, and another one for finding an optimal set of nodes when the size of the set is fixed. Based on the results, both algorithms seem to produce a good set of nodes. Interestingly, the greedy algorithm that produces a ranking worked well for any top $k$ nodes.

This work is preliminary in several aspects and at least the following aspects should be addressed in future work. (1) More efficient definitions or approximations of node proximity are needed for better scalability to large networks. (2) More expressive node similarities based, e.g., network reliability could prove more powerful, but are also computationally more demanding. (3) The current methods and experiments are a proof of concept and show great promise, but more experimentation is needed to understand the practical behaviour of the methods and parameters.

## References

[1] S. Asur and S. Parthasarathy. A viewpoint-based approach for interaction graph analysis. In *KDD '09*, pages 79–88, 2009.

[2] M. R. Berthold, U. Brandes, T. Kötter, M. Mader, U. Nagel, and K. Thiel. Pure spreading activation is pointless. In *CIKM '09*, pages 1915–1918, 2009.

[3] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, pages 335–336, 1998.

[4] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *SIGMOD '03*, pages 551–562, 2003.

[5] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR '08*, pages 659–666, 2008.

[6] C. J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

[7] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB Journal*, 16(4):523–544, 2007.

[8] L. De Raedt, K. Kersting, A. Kimmig, K. Revoredo, and H. Toivonen. Compressing probabilistic Prolog programs. *Machine Learning*, 70(2–3):151–168, 2008.

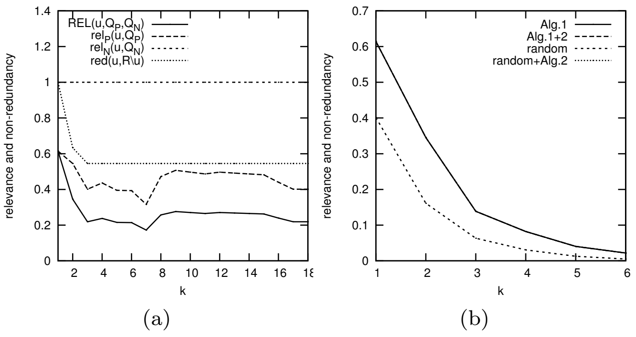
[9] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01*, pages 57–66, 2001.

[10] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. Knowledge discovery in databases: An overview. *AI Magazine*, pages 213–228, 1992.

[11] F. C. Gey, A. Chen, J. He, L. Xu, and J. Meggs. Term importance. Boolean conjunct training, negative terms, and foreign language retrieval: Probabilists algorithms at TREC-5. In *TREC-5*, 1996.

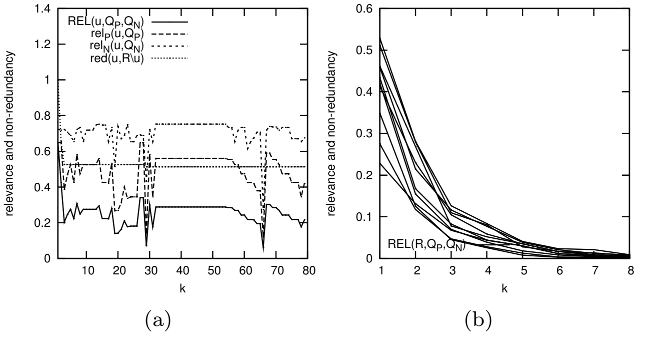
[12] P. Hintsanen and H. Toivonen. Finding reliable subgraphs from large probabilistic graphs. *Data Mining and Knowledge Discovery*, 17(1):3–23, 2008.

[13] S. Köhler, S. Bauer, D. Horn, and P. N. Robinson. Walking the interactome for proritization of candidate disease genes. *American Journal of Human Genetics*, 82(4):949–958, 2008.



(a) (b)

Figure 2: (a) Overall relevance (solid), relevance (dashed), irrelevance (short dashed), and nonredundancy (dotted line) of the $k$th node by the greedy algorithm. (b) Overall relevance of set $R_k$ of top $k$ nodes obtained by the three variants of the algorithms (the lines are indistinguishable) and random ranking for $k = 1, \ldots, 6$.



(a) (b)

Figure 3: (a): Overall relevance (solid), relevance (dashed), irrelevance (short dashed), and nonredundancy (dotted line) of the $k$th node by the greedy algorithm. (b): Overall relevance of the top $k$ nodes for 10 different networks by the greedy algorithm.

[14] Y. Koren, S. North, and C. Volinsky. Measuring and extracting proximity in networks. In *KDD '06*, pages 245–255, 2006.

[15] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Core algorithms in the CLEVER system. *ACM Transactions on Internet Technology*, 6(2):131–152, 2006.

[16] L. Langohr and H. Toivonen. A model for mining relevant and non-redundant information. In *ACM Symposium On Applied Computing*, 2012. In press.

[17] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *KDD '09*, pages 467–476, 2009.

[18] Q. Mei, J. Guo, and D. Radev. DivRank: the interplay of prestige and diversity in information networks. In *KDD '10*, pages 1009–1018, 2010.

[19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions — I. *Mathematical Programming*, 14:265–294, 1978.

[20] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1999.

[21] F. Pan, W. Wang, A. K. H. Tung, and J. Yang. Finding representative set from massive data. In *ICDM '05*, pages 338–345, 2005.

[22] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD '04*, pages 653–658, 2004.

[23] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. k-nearest neighbors in uncertain graphs. In *VLDB '10*, 2010.

[24] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *ICML '08*, pages 784–791, 2008.

[25] P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi, and H. Toivonen. Link discovery in graphs derived from biological databases. In *DILS '06*, pages 35–49, 2006.

[26] H. Tong and C. Faloutsos. Center-piece subgraphs: Problem definition and fast solutions. In *KDD '06*, pages 404–413, 2006.

[27] H. Tong, C. Faloutsos, and J.-Y. Pan. Random walk with restart: Fast solutions and applications. *Knowledge and Information Systems: An International Journal (KAIS)*, 14:327–346, 2008.

[28] H. Tong, J. He, Z. Wen, R. Konuru, and C.-Y. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *KDD '11*, pages 1028–1036, 2011.

[29] H. Tong, H. Qu, and H. Jamjoom. Measuring proximity on graphs with side information. In *ICDM '08*, pages 598–607, 2008.

[30] X. Wang, H. Fang, and C. X. Zhai. A study of methods for negative relevance feedback. In *SIGIR '08*, pages 219–226, 2008.

[31] D. Yan, L. Huang, and M. I. Jordan. Fast approximate spectral clustering. In *KDD '09*, pages 907–916, 2009.