Ranking in Heterogeneous Networks with Geo-Location Information

Abhinav Mishra
Stony Brook University
Department of Computer Science
abhmishra@cs.stonybrook.edu

Leman Akoglu
Carnegie Mellon University
H. John Heinz III College
lakoglu@cs.cmu.edu

Abstract

Entity ranking by importance or authority through relational information is an important problem in network science. A large body of existing work addresses the problem for homogeneous networks. With the emergence of richer networks, containing various types of entities and meta-data (e.g., attributes) in which edges carry rich semantic information, it becomes essential to build models that can leverage all available data in a meaningful way. In this work, we consider the ranking problem in heterogeneous information networks (HIN) with side information. Specifically, we introduce a new model called HINSIDE that has two key properties: (i) it explicitly represents the interactions (i.e., authority transfer rates or ATR) between *different types* of nodes, and (ii) it carefully incorporates the geo-location information of the entities to account for the *distance* and the *competition* between them. Besides an intuitive local formula, our model has a matrix form for which we derive a closed-form solution. Thanks to its closed form, HINSIDE lends itself to be used within various learning-to-rank objectives, for the estimation of its parameters (the ATR) provided training data. We formulate two kinds of objective functions for parameter learning with efficient estimation procedures. We validate the effectiveness of our proposed model and the learning procedures on samples from two real-world graphs, where we show the advantages of HINSIDE over popular existing models, including Pagerank and degree centrality.

1 Introduction

Given a network of entities with directed edges (e.g., a network of physicians with patient-referral relations), how can we quantify the "importance" or "authority" of the individual entities (e.g., to find the best cardiologists)? How about if the network is heterogeneous, consisting of various types of entities (e.g., physicians with different expertises)? What if we also have access to side information about the entities, such as their location (e.g., physical address)? How would all these different pieces of information (i.e., the network, entity types, locations) factor into the ranking of the entities by authority?

Ranking is an important problem in network analysis, and has been studied widely. Perhaps the most popular ranking methods include Pagerank [2] and HITS [8], which date back to nearly two decades. Roughly speaking, they model the importance of a node recursively, as a function of the importance of its neighbors. (See Appendix A) Those and many other centrality measures [17] mainly address the problem for homogeneous networks.

The ranking problem for heterogeneous networks differs, mainly for the different types of nodes in the network influence the importance of their neighbors differently. In the example of a network in which physicians refer patients to one another, a cardiologist referring to another cardiologist is intuitively a stronger indicator of the authority of the latter than say, a dietician referring to the same cardiologist. As such, what is called the "authority transfer rates" (ATR) between different types of entities should be carefully accounted for for a meaningful ranking. Among ranking models for heterogeneous networks, ObjectRank assumes the ATR to be known [1], while PopRank employs a slow suboptimal search procedure for estimating those from training data [9]. There are also other kinds of models for ranking in heterogeneous networks, such as RankClus [14] and NetClus [15] that perform simultaneous clustering and ranking (within clusters). (See Appendix B)

In this work, we consider the ranking problem for heterogeneous networks with side information. Specifically, we leverage the *location* of the entities to carefully account for (a) the distance as well as (b) the competition between them in ranking these entities. Simply put, the notion of distance between two connected entities becomes important in quantifying importance, especially in settings where distance incurs a 'cost' on the relation. For instance, in the physician referral network, patients travel the distance from the referrer to the referee. In a collaboration network, maintaining a working relation with distant colleagues speaks more to their importance. The notion of location also induces competition between the entities, as nearby entities can be thought to compete for drawing inlinks. This is evident for the pyhsician referrals scenario, where a referral to a specific physician can be thought as a 'preference' over the other (competing) physicians of the same expertise in their vicinity.

With the emergence of rich networks, such as heterogeneous information networks with meta-data (e.g., geo-coordinates), it becomes essential to build models that can use all available information in a meaningful way. Provided an appropriate formulation, one can *do better* in ranking (for example in aforementioned scenarios) than solely relying on the network structure. To the best of our knowledge, ours is the *first work* to leverage *location/distance* and to model the factor of *competition* between the entities for the ranking problem in HINs—hence is the first step toward ranking with *side information*. Our contributions are as follows:

- **Model formulation:** We propose HINSIDE, a new ranking model for heterogeneous networks in which entities exhibit location information. It carefully incorporates five elements for quantifying node authority: relation strength, distance, neighbor authority, authority transfer rates, and competition. (Sec. 3)
- **Closed-form solution:** Our model yields an intuitive local formula to compute the authority of an individual node. We also show how to write it in closed form, where the solution for all nodes corresponds to the left singular vector of a non-negative matrix. (Sec. 3.5)
- **Model estimation:** HINSIDE contains as parameters the authority transfer rates between the different entity types. We show how to estimate these parameters from training data, where our model lends itself to two different kinds of learning-to-rank objectives and efficient estimation procedures. (Sec. 4)

We evaluate our model and parameter estimation techniques on samples from two real-world heterogeneous networks, and show its advantages over various baselines including in-weight centrality, Pagerank, and a homogeneous model that ignores authority transfer rates. (Sec. 5)

Reproducibility: Source code of HINSIDE, parameter estimation algorithms, and the datasets used in this work are openly shared at https://github.com/abhimm/HINSIDE.

2 Motivation & The Problem

In this work we focus on the problem of entity ranking, where (1) besides the relationships among the entities, (2) the types of the entities as well as (3) external or *side information* such as (3a) the distance in-between the entities and (3b) their competition matter. We motivate this problem setting with the following example scenario.

Example. In the medical domain, consider a graph in which nodes represent medical providers or physicians, the type of a node depicts its expertise (e.g., cardiologist, physiologist, psychiatrist, etc.), and (directed) edges between the physicians capture "referral" relations, where one physician refers patients to another. The goal is to rank the physicians of a certain kind, called the *target type*, by their *authority*, e.g., to answer questions like identifying the best (highest authority) cardiologists in the database.

In this scenario, different types of physicians referring patients to the target type, say $\mathrm{cardiologist}$, would play different roles. Specifically, a $\mathrm{dietician}$ vs. a $\mathrm{cardiologist}$ referring their patients to a certain other $\mathrm{cardiologist}$ C would depict different information regarding the authority of C (intuitively, the latter is a stronger signal of C's authority). Moreover, the physical distance between the physicians is an important factor. Intuitively, a long-distance referral would indicate a stronger signal than a shorter one, as it implies that patients travel a long way to see a physician. Finally, the competition aspects should be modeled properly, as the *choice* of referring to a particular physician over another is an indication of their relative authorities.

DEFINITION 1. (HIN WITH SIDE INFORMATION) *A heterogeneous information network (HIN) is a (directed) graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, containing $|\mathcal{V}| = n$ entities of the same category (e.g., physician, company, etc.), where $w(i, j) \in \mathcal{W}$ depicts the weight of edge $e(i, j) \in \mathcal{E}$. A mapping function $t : \mathcal{V} \to \mathcal{T}$ maps each entity $i \in \mathcal{V}$ to one particular type $t_i \in \mathcal{T}$. $|\mathcal{T}| = m$ denotes the number of entity types. In addition, each entity $i \in \mathcal{V}$ is associated with a location l_i, where the symmetric function $d(l_i, l_j) \in \mathbb{R}_{\geq 0}$ returns the distance between nodes i and j.*

Given a HIN as described above, we address the entity ranking problem. Our goal is to rank the entities of the same kind (e.g., ranking of cardiologists in a medical referral network)—comparing entities of different types is not only meaningless (apples-to-oranges) but also not useful. Moreover, we aim to do a *global* ranking of entities, unlike proximity-based ranking.[1] Our goal is to identify the highly visible entities in the network, rather than entities similar or close-by to a given entity or set of entities.

DEFINITION 2. (ENTITY RANKING PROBLEM) ***Given** a HIN G with side information (in which each node v is associated with a type t_v and location l_v), and a target type $t_* \in \mathcal{T}$; **Find** authority scores r_v's for all the nodes $v \in \mathcal{V}$ of the target type, where $t_v = t_*$.*

As an example, consider a physician i of type t_i in location l_i referring a patient to a physician j of type t_j in location l_j. To quantify the significance and contribution of a link (referral) from i to (target physician) j's authority, we utilize *five main factors* in our ranking problem.

1. **Relation Strength:** The weight of the edge between two entities (in the example, number of referrals from i to j) is related to the magnitude of authority transfer.
2. **Relation Distance:** The larger the distance between i and j, the more authority j would receive. Intuitively,

[1] Well-known proximity-based ranking methods include Personalized PageRank [7] and SimRank [6] on homogeneous networks, and PathSim [13] on heterogeneous networks.

distance traveled (in this case by i's patients to visit j) speaks to the quality of (physician) j.

3. **(In-)Neighbor Authority:** The more authority the source (physician) i has, the more authority the target (physician) j obtains through a link (referral) from i to j. Similar to Pagerank [2], authority of a node is a function of the authority of its (in-)neighbors.
4. **Authority Transfer Rates:** The authority the target (physician) j obtains through a link (referral) from i also depends on i's type along with j's type itself. In the example case, while an optometrist referring a patient to an ophthalmologist may be ordinary in case the patient needs a surgery, an ophthalmologist referring their patient to another ophthalmologist may imply a significant (rate of) authority transfer.
5. **Competition:** The number and the authorities of the entities (physicians) of type t_j that are in close physical distance to i is another important factor. The more and the higher-rated entities of type t_j around i exist, the larger the authority score of j would get by a link (referral) from i—as such a link implies i's preference of j over other entities of type t_j in its vicinity.

Figure 1 gives an illustrative example. The network contains 6 nodes of 2 types (gray and white circles), from two different geo-regions (boxes). The transfer rates are set to 0.7 for within same-type edges and 0.3 for across the types. HINSIDE ranks node 3 highest, as it has many in-links, particularly $(1, 3)$ and $(2, 3)$ from *distant* and *same-type* nodes. In comparison, Pagerank ranks node 1 with the largest total relation strength the highest. Node 2 is ranked second by Pagerank, whereas HINSIDE ranks node 1 at second position, above 2. This is mainly due to link $(4, 1)$ that 'prefers' node 1 over the *competing* node 2 of the same type. Among type-white nodes, node 6 has highest Pagerank. HINSIDE in contrast ranks node 5 above 6, due to the link from the highest ranked node 3 that makes node 5 more competent than 6. Both models rank node 4 the lowest.

3 Proposed HINSIDE Model

We describe our ranking model by incrementally incorporating the five main elements as listed in the previous section.

3.1 Relation Strength and Distance Let W denote the $n \times n$ log-weighted adjacency matrix of G, where $W(i, j) = \log(w(i, j) + 1)$. Similarly, we define the $n \times n$ distance matrix D such that $D(i, j) = \log(d(l_i, l_j) + 1)$.

To account for the relation distance, we combine the adjacency matrix W with the distance matrix D, in order to increase the value of the edges that connect nodes with longer distance and subsequently decrease the value of those edges that connect nodes with less distance. That is,

$$M = W \odot D \tag{3.1}$$

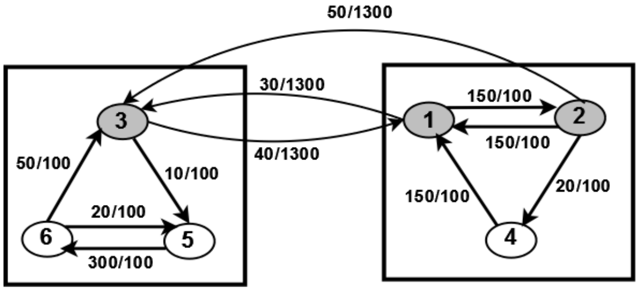

Figure 1: Example network with two node types (colors). Edges annotated by weight/distance.

where $\odot$ is the Hadamard or element-wise product.

3.2 (In-)Neighbor Authority To compute the authority of each node, we take the weighted sum of the authorities of its (in-)neighbors in (directed) G. The (in-)edges are weighted by relation strength (i.e., edge weight) and distance as described above.

$$r_i = \sum_{j \in \mathcal{V}} M(j, i)\, r_j \tag{3.2}$$

where r_i denotes the authority score of node i. Thus far, our model is similar to PageRank. The key difference is in modifying the adjacency matrix by accounting for the distance between the connected pairs.

3.3 Authority Transfer Rates In this work we consider a typed, i.e. heterogeneous network. As motivated in the previous section, the neighbors of different types of a node should count differently and have different impact on the authority of the node. As such, we incorporate what is called "authority transfer rates [1]" (ATR) $\Gamma(a, b) \geq 0$ between type $\mathcal{T}(a)$ and type $\mathcal{T}(b)$, $\forall a, b = \{1, \ldots, m\}$. These rates represent the impact or importance of links between nodes of various types.

The authority score of a node i then becomes

$$r_i = \sum_{j \in \mathcal{V}} \Gamma(t_j, t_i)\; M(j, i)\; r_j. \tag{3.3}$$

The $m \times m$ ATR matrix Γ contains vital parameters for our ranking model since a meaningful ranking can only be achieved by using the appropriate transfer rates.

3.4 Competition Finally we consider what we call the concept of "competition in the vicinity of the source". Consider the edge $e(j, i)$ from a type t_j node to a type t_i node. Intuitively, when j links to i, it "prefers" i over other nodes of type t_i that are in close proximity to j (see Figure 1). Recalling our earlier examples, if a dietician j refers a patient to a cardiologist i while there exist other cardiologists close-by to j, then i has supposedly higher authority than those others. Similarly, if an energy company j trades goods with a transportation company i while there exist other transport industries in j's vicinity, we assume i's authority to be higher than those.[2]

To capture this intuition, for each edge $e(j, i)$ we think of "ghost" edges from nodes of type t_i in j's vicinity to i. Instead of a fixed vicinity, we define a smooth neighborhood function that is a decreasing function of distance:

$$N(u, v) = \begin{cases} g(d(l_u, l_v)) & u, v \in \mathcal{V},\ u \neq v \\ 0 & u = v \end{cases}$$

where $g(.)$ is monotonically decreasing, e.g. $g(z) = e^{-z}$.

Then, we transfer a weighted sum of the authority scores of type t_i nodes in j's *neighborhood* along with j's authority itself to compute i's score. Building on (3.3) we get:

$$(3.4)\quad r_i = \sum_j \Gamma(t_j, t_i)\, M(j, i)\; (\, r_j + \sum_{v:t_v = t_i} N(v, j)\; r_v\;) \; .$$

3.5 Solving the HINSIDE Model. Our proposed model given in Eq. (3.4) can be written and solved in a compact form. Let T denote the $n \times m$ boolean type matrix with $T(i, c) = 1$ if $t_i = \mathcal{T}(c)$ and 0 otherwise, $\forall i \in \mathcal{V}$ and $c = \{1, \dots, m\}$. Based on this, we define $L = M \odot (T\, \Gamma\, T')$ where $'$ denotes transpose operation. We also introduce the type equality matrix E where

$$E(u, v) = \begin{cases} 1 & if\ t_u = t_v \\ 0 & \text{otherwise} \end{cases}$$

In matrix form, $E = TT'$. We then rewrite (3.4) as

$$r_i = \sum_j L(j, i)\; r_j + \sum_j \sum_v L(j, i) N(v, j)\; r_v\; E(i, v)$$

$$r_i = \sum_j L(j, i)\; r_j + \sum_v r_v\; E(i, v) \big[\sum_j N(v, j) L(j, i) \big]$$

$$\mathbf{r} = L'\mathbf{r} \; + \; (\; E \odot (NL)'\;)\mathbf{r}$$

As such, we obtain

$$(3.5)\qquad \mathbf{r} \; = \; \big[L' + (L'N' \odot E) \big]\; \mathbf{r} \; = \; H\; \mathbf{r}$$

where $\mathbf{r} \in \mathbb{R}^n$ is a column vector of length n containing the authority scores of all nodes. Note that N is computed from data (Sec. 3.4), $E = TT'$ and $L = M \odot (T\; \Gamma\; T')$ in which T and M are also known. As such, the ATR matrix Γ is the only unknown of our model.

We can solve for $\mathbf{r}$ using the power method [16]. As Γ, T, N and M are all non-negative, so is H. Starting with an arbitrary initial vector $\mathbf{r}^{(0)} \in \mathbb{R}^n$, we form the vector sequence $\{\mathbf{r}^{(p)}\}_{p=0}^{\infty}$. If $||H|| \neq 1$, the power method would underflow or overflow for large p, and not converge to a fixed $\mathbf{r}$. As such, we introduce a normalization at every step:

$$(3.6)\qquad \mathbf{r}^{(p+1)} \leftarrow \frac{H\; \mathbf{r}^{(p)}}{||H\; \mathbf{r}^{(p)}||}, \;\; p = 0, 1, 2, \dots$$

It can be shown that the power method converges to the left singular vector of H under some mild conditions [12]:

THEOREM 3.1. *Let the singular values of $H \in \mathbb{R}^{n \times n}$ be arranged such that $|\sigma_1| > |\sigma_2| \geq \dots \geq |\sigma_n|$. Let $\mathbf{u}_1$ and $\mathbf{v}_1$ be the left and right singular vectors of H corresponding to σ_1 respectively. Then, the vector sequence generated by (3.6) converges to $\mathbf{u}_1$, where $||\mathbf{r}^{(p)}||$ converges to $|\sigma_1|$ for large p, provided that $\mathbf{v}_1'\mathbf{r}_0 \neq 0$ and $|\sigma_1| \neq |\sigma_2|$.*

For a HIN with m types, Γ contains m^2 parameters. Even for moderate m, it would be challenging to set these ATR values manually. Next we propose two new algorithms for parameter estimation provided training data.

4 Parameter Estimation

The authority transfer rates, in other words the values in Γ, depend on the problem domain and may be hard to assign by humans. To estimate Γ, we consider learning from partially ranked lists, given by humans, as providing partial lists is more practical than assigning absolute rates. We propose two approaches for estimating Γ; (1) a RankSVM approach, and (2) a gradient based approach.

Let us first represent r_i in the form of a linear function of a feature vector $\mathbf{x}_i$ and a weight vector $\mathbf{w}$, such that $r_i = f(\mathbf{x}_i) =< \mathbf{w}, \mathbf{x}_i >$. This is a convenient and common representation to be used in many learning algorithms.

We start by rewriting Eq. (3.4) as

$$r_i = \sum_t \Gamma(t, t_i) \sum_{j:t_j = t} \big[M(j, i)(r_j + \sum_{v:t_v = t_i} N(v, j)\; r_v) \big]$$

Let us define a $m \times n$ matrix X where

$$(4.7)\quad X(t, i) = \sum_{j:t_j = t} M(j, i)\; (\; r_j + \sum_{v:t_v = t_i} N(v, j)\; r_v\;)$$

using which we can write

$$(4.8)\quad r_i = \sum_t \Gamma(t, t_i) X(t, i) = \Gamma'(t_i, :) \cdot X(:, i) = \mathbf{\Gamma}'_{t_i} \cdot \mathbf{x}_i$$

where $\mathbf{x}_i$ is the i^{th} column of X and $\mathbf{\Gamma}_{t_i}$ is the t_i^{th} column of Γ. As such, we can compute r_i by the vector-vector product

$$(4.9)\qquad r_i = f(\mathbf{x}_i) =< \mathbf{\Gamma}_{t_i}, \mathbf{x}_i > \; .$$

In this formulation, $\mathbf{\Gamma}_{t_i}$ is a length m vector of unknown parameters and $\mathbf{x}_i$ is considered as the "feature vector" of node i. Now in order to estimate Γ we need access to $\mathbf{x}_i$'s, and to construct the $\mathbf{x}_i$'s we need to know the authority scores $\mathbf{r}$ (Eq. 4.7), which in turn requires Γ (Eq. 3.4). That is, $\Gamma \xrightarrow{Eq.(3.4)} \mathbf{r} \xrightarrow{Eq.(4.7)} X \xrightarrow{\text{estimate}} \Gamma$. In this section we describe algorithms for exactly the last step. The dependences suggest that an alternating optimization scheme is an appropriate approach to estimating Γ. The sketch

[2] The referrals are directed, while trade relations are undirected. In the latter we also assume the vice versa, i.e., j's authority to be higher than other energy companies in i's vicinity.

Algorithm 1 Alternating Estimation of Γ

```
Input: graph $G$, partial ranked lists $\mathcal{L}$, $T_{\max}$, $\epsilon$
Output: $\Gamma$
$\Gamma^0(a,b) = rand(0,1),\ \ \forall a,b \in \{1,\dots,m\}, k = 0$
$\mathbf{r} \leftarrow$ compute authority scores by (3.6) using $\Gamma^0$
repeat
    $X^k \leftarrow$ compute feature vectors by Eq. (4.7) using $\mathbf{r}$
    $\Gamma^{k+1} \leftarrow$ learn new param.s by RankSVM($\mathcal{L}, X^k$)
    or Gradient($\mathcal{L}, X^k, \Gamma^k$)
    $\mathbf{r} \leftarrow$ compute authority scores by (3.6) using $\Gamma^{k+1}$
    $diff \leftarrow trAccuracy(\mathcal{L}, \mathbf{r}) - trAccuracy(\mathcal{L}, \mathbf{r}_{best})$
    if $diff > 0$ then $\mathbf{r}_{best} \leftarrow \mathbf{r}$, $\Gamma_{best} \leftarrow \Gamma^{k+1}$ end if
    $k = k + 1$
until $||\Gamma^k - \Gamma^{k-1}|| \le \epsilon$ or $k > T_{\max}$
return $\Gamma_{best}$
```

of our iterative meta-approach is given in Algorithm 1. Over iterations the best Γ with the largest $trAccuracy$ is maintained. Here any IR metric can be used for accuracy, such as discounted cumulative gain (See Sec. 5).

Given a HIN G, a partial ranked list L_t consists of an ordering of a subset of nodes $V_t \subset \mathcal{V}$ of the same type, i.e., $t_v = t,\ \forall v \in V_t$. Let ϕ_v denote the order or position of node v in L_t, where lower positions correspond to higher ranks or authority scores, that is $r_u \ge r_v$ if and only if $\phi_u < \phi_v$.

Our estimation algorithms take as input one or more partial ranked lists $\mathcal{L}$ for each type $t \in \mathcal{T}$. It first randomly guesses Γ, and then iteratively and alternatingly computes $\mathbf{r}$ and X, followed by estimating Γ for which we propose two main aproaches; RankSVM and Gradient.

4.1 RankSVM formulation

Given a partial ranked list, there are several ways of constructing training data from it. A common way is the pair-wise approach, where for each pair of entities (i.e., nodes) (u, v) in the ranked list, we construct a training instance $((\mathbf{x}_u, \mathbf{x}_v), 1)$ if u is ranked ahead of v (that is, if $\phi_u < \phi_v$), and $((\mathbf{x}_v, \mathbf{x}_u), -1)$ otherwise. As a result, training data $\mathcal{D}$ is available in the form of $\{((\mathbf{x}_d^1, \mathbf{x}_d^2), y_d)\}_{d=1}^{|\mathcal{D}|}$, where each instance consists of two feature vectors that belong to two nodes of the same type, and a label $y_d \in \{-1, 1\}$.

Having constructed such a training data $\mathcal{D}$, we can use the hinge-loss function as shown in (4.10) to estimate the model parameters by RankSVM [5].

(4.10)
$$L((\mathbf{x}_d^1, \mathbf{x}_d^2), y_d) = \max(\ 0,\ 1 - (\mathbf{\Gamma}_t' \cdot (\mathbf{x}_d^1 - \mathbf{x}_d^2))y_d\ ),$$
$$\text{such that } \ t_{\mathbf{x}_d^1}, t_{\mathbf{x}_d^2} = t$$

Note that each column of Γ that belongs to each type t is estimated independent of others, provided the feature vectors $\mathbf{x}_v$'s where $t_v = t$. Since Γ is a non-negative matrix, we also introduce non-negativity constraints to the SVM formulation, given in (4.11).

Algorithm 2 Estimate Γ by RankSVM

```
Input: feature vectors $X$, partial ranked lists $\mathcal{L}$
Output: $\Gamma$
for each type $t \in \mathcal{T}$ do
    $\mathbf{\Gamma}_t \leftarrow$ compute column $t$ of $\Gamma$ by (4.11) using $X$, $\mathcal{L}$
end for
return $\Gamma$
```

(4.11)
$$\begin{aligned} &\min_{\mathbf{\Gamma}_t}\ ||\mathbf{\Gamma}_t||_2^2\ +\ \gamma \sum_{d \in \mathcal{D}} \epsilon_d \\ &\text{s.t. } \ \mathbf{\Gamma}_t'(\mathbf{x}_d^1 - \mathbf{x}_d^2)y_d \ge 1 - \epsilon_d,\ \forall d \in \mathcal{D} \text{ and } t_{\mathbf{x}_d^1}, t_{\mathbf{x}_d^2} = t \\ &\qquad \epsilon_d \ge 0,\ \forall d \in \mathcal{D} \\ &\qquad \mathbf{\Gamma}_t(c) \ge 0,\ \forall c = 1, \dots, m \end{aligned}$$

where γ is a regularization hyperparameter estimated through cross validation. With the additional constraints, the optimization remains a convex problem. We solve the program in (4.11) m times independently for each type t to estimate all the columns of Γ, as shown in Algorithm 2.

4.2 Gradient-based estimation

In addition to adapting RankSVM formulation, we can also construct other learning-to-rank objective functions and leverage a gradient-based method to solve our learning problem. In this section, we introduce two different objectives with different requirements.

4.2.1 Learning-to-Rank Objective-I

Consider the case where besides the partial ranked lists, for each pair of entities (u, v) in a training instance, the probability that one is ranked ahead of the other is also given: i.e., the training instances are in the form of $((\mathbf{x}_u, \mathbf{x}_v), \bar{p}_{uv})$ where $\bar{p}_{uv} = P(r_u > r_v)$. For example, one can use the sigmoid function $\sigma(r_u - r_v) = \bar{p}_{uv}$ to compute this probability, if the original/ground-truth authority scores (r_u, r_v) of the training entities are provided (note that this is a strict requirement), where $\sigma(x) = \frac{e^x}{1+e^x}$ is the sigmoid function.

Recalling function $f : \mathbb{R}^m \to \mathbb{R}$ given in Eq. (4.9), let

$$o_u = f(\mathbf{x}_u) = < \mathbf{\Gamma}_{t_u}, \mathbf{x}_u >, \quad \text{and}$$

$$o_{uv} = f(\mathbf{x}_u - \mathbf{x}_v) = f(\mathbf{x}_{uv}) = < \mathbf{\Gamma}_{t_u = t_v}, \mathbf{x}_{uv} > \ .$$

We then utilize the cross entropy as our cost function for each training instance (u, v) as proposed in [11]:

$$c_{uv} = -\bar{p}_{uv} \log(p_{uv}) - (1 - \bar{p}_{uv}) \log(1 - p_{uv}) \tag{4.12}$$

where, mapping from the output of our model to probabilities is acquired using the logistic function

$$p_{uv} = \frac{e^{o_{uv}}}{1 + e^{o_{uv}}} \ . \tag{4.13}$$

Substituting Eq. (4.13) into Eq. (4.12), c_{uv} can equivalently be written as

$$c_{uv} = -\bar{p}_{uv} o_{uv} + \log(1 + e^{o_{uv}}) \tag{4.14}$$

The total cost is then $C = \sum_{(u,v)\in\mathcal{D}} c_{uv}$ and the objective function becomes

$$\min_{\mathbf{\Gamma}_t} C = \sum_{(u,v)\in\mathcal{D}} c_{uv} = \sum_{(u,v)\in\mathcal{D}} -\bar{p}_{uv} o_{uv} + \log(1 + e^{o_{uv}}) \tag{4.15}$$

for training instances that belong to type t.

To solve the above objective, which is convex, we can utilize a gradient-based approach [3]. In particular, taking the derivative of C with respect to $\mathbf{\Gamma}_t$ we get

$$\frac{\partial C}{\partial \mathbf{\Gamma}_t} = \sum_{(u,v)\in\mathcal{D}} -\bar{p}_{uv}(\mathbf{x}_u - \mathbf{x}_v) + \frac{e^{o_{uv}}}{1 + e^{o_{uv}}}(\mathbf{x}_u - \mathbf{x}_v) \tag{4.16}$$

Using Eq. (4.13) we can write Eq. (4.16) as

$$\begin{aligned} \frac{\partial C}{\partial \mathbf{\Gamma}_t} &= \sum_{(u,v)\in\mathcal{D}} -\bar{p}_{uv}(\mathbf{x}_u - \mathbf{x}_v) + p_{uv}(\mathbf{x}_u - \mathbf{x}_v) \\ &= \sum_{(u,v)\in\mathcal{D}} (p_{uv} - \bar{p}_{uv})(\mathbf{x}_u - \mathbf{x}_v) \end{aligned} \tag{4.17}$$

The parameter vector then can be updated at every step s of the gradient descent by

$$\mathbf{\Gamma}_t^{s+1} \leftarrow \mathbf{\Gamma}_t^{s} - \eta \frac{\partial C}{\partial \mathbf{\Gamma}_t^{s}} \tag{4.18}$$

where η is the step size[3], and where p_{uv} in (4.17) at step s is computed using $\mathbf{\Gamma}_t^s$. The outline of this gradient-based method for estimating the ATR is given in Algorithm 3.

Algorithm 3 Estimate Γ by GRADIENT

```
Input: feature vectors $X$, partial ranked lists $\mathcal{L}$, $\Gamma^k$
Output: $\Gamma$
for each type $t \in \mathcal{T}$ do
    $\mathbf{\Gamma}_t^0 = \Gamma_t^k,\quad s = 0,\quad \eta = 1/|\mathcal{D}|$
    repeat
        $\mathbf{\Gamma}_t^{s+1} \leftarrow$ compute column $t$ of $\Gamma$ by (4.18),
            using $\mathcal{L}$, $X$, $\Gamma_t^s$, $\eta$, based on (4.17) or (4.21).
        $s = s + 1,\quad \eta = 1/(|\mathcal{D}|\sqrt{s})$
    until convergence
end for
return $\Gamma$
```

4.2.2 Learning-to-Rank Objective-II The Objective-I in (4.15) that we considered in the previous section assumes that the ground-truth authority scores of entities in the training data are known, to compute the $\bar{p}_{uv}$'s. This is also the case for some point-wise and list-wise learning to rank approaches such as Subset Ranking [4] and SVM MAP [18]. Unfortunately, this requirement is not realistic to be put in practice—providing absolute scores is quite impractical compared to simply providing ranked lists.

Therefore, an alternative objective function we utilize is by Rendle et al. [10] that aims to minimize the total negative log likelihood $-\log p_{uv}$ for ordered training pairs (u, v):

$$\min_{\mathbf{\Gamma}_t} C = \sum_{(u,v)\in\mathcal{D}:\phi_u<\phi_v} -\log\ \sigma(<\mathbf{\Gamma}_t, \mathbf{x}_{uv}>) \tag{4.19}$$

Using the previous notation, we can write (4.19) as

$$\min_{\mathbf{\Gamma}_t} C = \sum_{(u,v)\in\mathcal{D}:\phi_u<\phi_v} -o_{uv} + \log\ (1 + e^{o_{uv}}) \tag{4.20}$$

This new objective in (4.20) is also convex. Taking its derivative with respect to $\mathbf{\Gamma}_t$ we get

$$\begin{aligned} \frac{\partial C}{\partial \mathbf{\Gamma}_t} &= \sum_{(u,v)\in\mathcal{D}:\phi_u<\phi_v} -\mathbf{x}_{uv} + \frac{e^{o_{uv}}}{1 + e^{o_{uv}}}\mathbf{x}_{uv} \\ &= \sum_{(u,v)\in\mathcal{D}:\phi_u<\phi_v} (p_{uv} - 1)(\mathbf{x}_u - \mathbf{x}_v) \end{aligned} \tag{4.21}$$

Comparing Objective-I and Objective-II: Notice that the gradient in (4.21) is very similar to that in (4.17): the main difference being $(p_{uv} - \bar{p}_{uv})$ vs. $(p_{uv} - 1)$. Intuitively, the former objective is aiming to obtain a $\mathbf{\Gamma}_t$ such that the estimated p_{uv}'s are as close to the given $\bar{p}_{uv}$'s as possible. On the other hand, the latter is aiming to obtain a large difference between the estimated authority score of a higher ranked node and a lower ranked node, i.e., obtain a large *positive* $\mathbf{\Gamma}_t\mathbf{x}_u - \mathbf{\Gamma}_t\mathbf{x}_v =< \mathbf{\Gamma}_t, \mathbf{x}_{uv} >= o_{uv}$, where $\phi_u < \phi_v$ as u is ranked higher than v in L_t. The larger the o_{uv}, the closer the $p_{uv} = \sigma(o_{uv})$ to 1.

Most importantly, our second objective does not require $\bar{p}_{ij}$'s to be given as part of the training data, which is quite harder to obtain than the partial ranked lists alone.

Projected gradient descent for non-negativity: We presented our gradient-based approaches without any constraints on Γ. As such, we may end up finding a solution with negative entries. To enforce non-negativity we employ projected gradient descent, where we split each gardient iteration as given in (4.18) into two steps:

$$\hat{\mathbf{\Gamma}}_t^{s+1} \leftarrow \mathbf{\Gamma}_t^{s} - \eta\nabla(\mathbf{\Gamma}_t^{s}) \tag{4.22}$$

$$\mathbf{\Gamma}_t^{s+1} \leftarrow \min_{\mathbf{\Gamma}_t\in R} \frac{1}{2}\|\hat{\mathbf{\Gamma}}_t^{s+1} - \mathbf{\Gamma}_t\|_2^2\ . \tag{4.23}$$

The goal is to find a $\mathbf{\Gamma}_t$ that is (1) closest to the updated solution at step $s + 1$ but (2) one that is within the feasible/constrained region R.

For non-negativity, solving a constrained minimization as in (4.23) is relatively easy. Specifically, at every step s of the gradient descent, we compute the gradient ∇ w.r.t. $\mathbf{\Gamma}_t^s$ as usual (using either (4.17) or (4.21)), update to $\hat{\mathbf{\Gamma}}_t^{s+1}$ as in (4.22), and then set its negative entries to zero in order to obtain $\mathbf{\Gamma}_t^{s+1}$, which would be the optimal solution to (4.23).

[3] We set $\eta = 1/(|\mathcal{D}|\sqrt{s})$ such that the step size gradually decreases with increasing number of gradient steps.

5 Evaluation

5.1 Experiment Setup **Datasets.** We perform experiments on samples from two real-world networks (both publicly available). Two sample graphs are obtained through snowball sampling from a (directed and weighted) real-world medical-referral network, called the DocGraph.[4] It is a teaming graph in which the edges represent the referrals of patients between medicare providers on the same cases. Edge weights capture the number of referrals between two physicians, and the node locations depict the latitude-longitude of the physicians' offices.

The first graph DocGI contains the interactions between $m = 3$ types (different expertises) of physicians and consists of $n = 446$ nodes and 8537 edges. The break-down of the node counts by type is $\{279, 103, 64\}$.

The second graph DocGII is a larger subgraph with $n = 3979$ nodes and 93432 edges, containing interactions between $m = 7$ types of physicians, each with $\{1653, 663, 394, 333, 311, 323, 302\}$ nodes respectively.

In addition, we use the DBLP 4Area co-authorship dataset[5], which contains researchers from 4 research areas: database (DB), data mining (DM), machine learning (ML) and information retrieval (IR). The break-down of node counts by area is $\{2023, 1150, 2432, 1014\}$ respectively. Edges are weighted by the number of co-authored articles. Out of 27K authors, we are able to identify and crawl the institution/geo-coordinates of around 11K. We then induce the 4Area network on these 11K nodes, and use the largest connected component with 6619 nodes and 26804 edges. The final graph, called AuthGraph, contains researchers from 72 different countries across 6 continents.

There is no agreed-upon measure of authority or impact in academia, nevertheless we use h-index as a proxy (crawled from GoogleScholar as of July 2015). Our goal is not to propose a new measure of authority or reproduce h-index (as it is based on citations while we use collaborations), but to study the relation of HINSIDE with external measures, and to identify those cases (i.e., researchers) which are ranked differently by HINSIDE than e.g., h-index and Pagerank.

Compared methods. We evaluate the proposed algorithms:

- RSVM-NN: RankSVM formulation in (4.11) with non-negativity constraint on Γ,
- GD-I-NN: GraDient-based approach with Objective-I in (4.15) and non-negativity constraint on Γ,
- GD-II-NN: GraDient-based approach with Objective-II in (4.19) and non-negativity constraint on Γ,
- RSVM-NC: RankSVM with no constraint,
- GD-I-NC & GD-II-NC: Gradient with no constraint.

We also compare to the following baselines:

- RG: Randomly Guess Γ and run HINSIDE,
- RO: Randomly Order all nodes (no model involved),
- INW: Rank nodes by their total in-weight (centrality),
- PRANKW: Rank by Pagerank (centrality) on the Weighted directed input graph ($\alpha = 0.15$).

All methods except PRANKW and INW are randomized; our meta-algorithm in Alg. 1 starts with a randomly initialized Γ, RG guesses Γ randomly, and RO guesses the final ranking randomly. We run each of these methods 10 times and report Γ that achieves the best training accuracy based on the input partial ranked lists.

In evaluation, we randomly select $1/3$ of the nodes of each type and use their ground truth ranking to create the partial ranked lists for training. That is, we provide one partial ranked list per type as input to parameter estimation. The remaining $2/3$ of the nodes is used as test data.

Metrics. For evaluating ranking performance, we use two popular measures in IR: NDCG (normalized discounted cumulative gain) and AP@k (average precision at k).

5.2 Experiment Results We first evaluate our proposed parameter estimation algorithms in Sec. 4.

Analysis on DocGI and DocGII. For the DocGI network with $m = 3$ node types, we manually set the (3×3) Γ matrix with various non-negative authority transfer rates between the types. Note that this matrix is not necessarily symmetric. We perform 15 such experiments, with 15 different ground truth Γ matrices, and compare the 10 different estimation methods as listed in the previous section across experiments.

Figure 2 shows the $AP@20$ ranking accuracy of the compared methods. The box-plots depict the accuracy (y-axis) across 15 experiments, with the minimum, maximum, median, and 25%-75% marked. The first 10 box-plots show the *average* accuracies across all 3 types for the individual methods (x-axis), followed by accuracies for each type.

All the proposed algorithms (RSVM, GD-I, GD-II; with and without non-negativity constraints) perform well, where the median $AP@20$ is well above 0.85 across experiments, for all types. Table 1 (left) lists the mean accuracy across the experiments. We find that on average the proposed RSVM-NN, the RankSVM formulation with non-negativity constraint, performs the best with mean accuracy around 0.95 (across types and experiments). The gradient-based approach GD-II using Objective-II performs as well as GD-I that uses Objective-I, despite consuming less information—recall that GD-I requires original authority *scores* of entities in the training data besides their ranking. The proposed methods *with* non-negativity constraints performed slightly better than dropping such constraints, however the differ-

[4] This data is obtained from Next Level Doctor Social Graph, Phase 1, a project by Fred Trotter and is a response to a Freedom of Information Act (FOIA) request (See http://www.medstartr.com/projects/82-next-level-doctor-social-graph-phase-1). The physician referral data for years 2009–2015 is publicly available at https://questions.cms.gov/faq.php?faqId=7977

[5] http://web.engr.illinois.edu/~mingji1/DBLP_four_area.zip

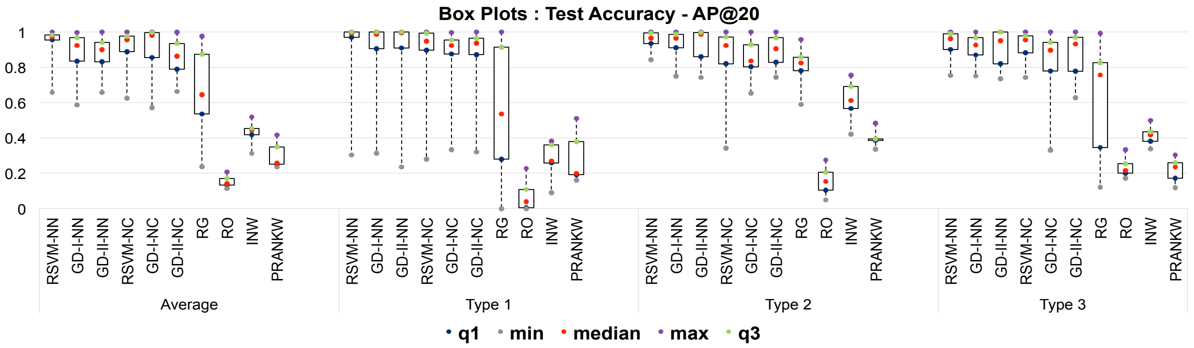

Figure 2: $AP@20$ accuracy of compared methods on DocGI ($m = 3$).

Table 1: Mean $AP@20$ across (left) 15 experiments with different Γ on DocGI with $m = 3$ types, and (right) 10 experiments with different Γ on DocGII with $m = 7$ types.

Method	Type 1	Type 2	Type 3	Average	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Average
RSVM-NN	**0.9435**	**0.9577**	**0.9361**	**0.9458**	0.8367	0.9030	0.9401	**0.9639**	**0.9753**	**0.9568**	0.9362	**0.9303**
RSVM-NC	0.9207	0.9372	0.9140	0.9240	**0.8605**	**0.9361**	**0.9701**	0.9429	0.8829	0.9330	**0.9590**	0.9263
GD-I-NN	0.9011	0.8641	0.9192	0.8948	0.7193	0.8830	0.9074	0.9357	0.8482	0.8812	0.8906	0.8665
GD-I-NC	0.8852	0.9358	0.9182	0.9131	0.6999	0.8663	0.9030	0.9015	0.9143	0.8838	0.8710	0.8628
GD-II-NN	0.8975	0.9022	0.8851	0.8949	0.8161	0.8978	0.9574	0.9485	0.9441	0.9239	0.9074	0.9136
GD-II-NC	0.8659	0.8628	0.8602	0.8630	0.7617	0.8896	0.9465	0.9599	0.9557	0.9177	0.9024	0.9048
RG	0.6231	0.8278	0.6712	0.7074	0.5358	0.6483	0.6871	0.6653	0.6796	0.6602	0.6240	0.6429
RO	0.0643	0.1520	0.2342	0.1502	0.0029	0.0109	0.0240	0.0494	0.0357	0.0301	0.0326	0.0265
PRANKW	0.2977	0.3890	0.2169	0.3012	0.0180	0.0739	0.0464	0.0852	0.0745	0.0183	0.1818	0.0711
INW	0.2862	0.5942	0.4183	0.4329	0.2143	0.2808	0.3053	0.1326	0.2725	0.3946	0.2555	0.2651

ences are not significant. Analysis showed that those without the constraints often estimated a non-negative Γ.

We give below three example (3×3) ground truth Γ matrices with small differences inbetween (i.e., swaps in bold). Corresponding Γ's estimated by RSVM-NN are also shown. Interestingly, the ratios of ATR values in this column match almost identically to those estimated by RSVM-NN.

(a) example (3×3) ground truth Γ matrices

$$\begin{bmatrix} 0.03 & 0.06 & 0.12 \\ 0.09 & 0.19 & 0.42 \\ 0.16 & 0.35 & 0.90 \end{bmatrix} \begin{bmatrix} \mathbf{0.90} & 0.06 & 0.12 \\ 0.09 & 0.19 & 0.42 \\ 0.16 & 0.35 & \mathbf{0.03} \end{bmatrix} \begin{bmatrix} 0.03 & 0.06 & 0.12 \\ 0.09 & 0.19 & 0.42 \\ \mathbf{0.90} & 0.35 & \mathbf{0.16} \end{bmatrix}$$

(b) estimated Γ by RSVM-NN

$$\begin{bmatrix} 1.13 & 4.55 & 2.89 \\ 3.39 & 0.34 & 0.10 \\ 6.02 & 0.05 & 0.02 \end{bmatrix} \begin{bmatrix} 7.02 & 0.13 & 2.63 \\ 0.68 & 0.00 & 0.08 \\ 1.17 & 0.00 & 0.00 \end{bmatrix} \begin{bmatrix} 1.81 & 0.25 & 0.10 \\ 5.39 & 0.00 & 0.00 \\ 54.08 & 0.00 & 0.00 \end{bmatrix}$$

We repeat our experiments for DocGII with $m = 7$, for which we create 10 different (7×7) Γ matrices. Table 1 (right) shows corresponding results, from which we can deduce similar conclusions (also see Figure 5 in Appendix C). Similarly, all proposed algorithms continue to do equally well, despite the increased parameter size (49 vs. 9), where RSVM-NN performs the best with mean accuracy around 0.93 (across types and experiments).

The performances of the baselines on DocGII decrease even further. For RG, the lower accuracy on average (and the even higher variance) can be attributed to the larger parameter size for which "guessing" is no longer as easy. As for PRANKW and INW, the decline in accuracy is likely due to their homogeneous nature—both of them ignore neighbor types, which is more of a concern for a network with 7 different types of nodes.

Similar results are observed when we use the NDCG measure. Figure 3 shows the average NDCG per experiment (across types) for each method. Again, the proposed algorithms achieve competitive accuracy, where RSVM produces slightly better results. The baselines are unable to capture the ranking by the HINSIDE model, where guessing the parameters is unavailing for various settings (e.g., experiments 1, 7, 10) (similar plot for DocGI omitted for brevity).

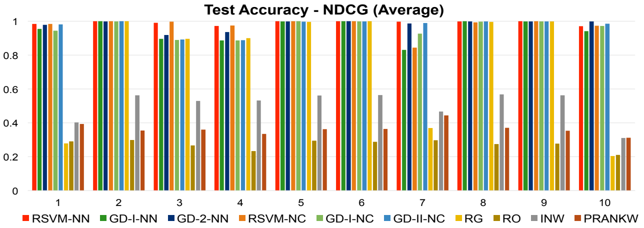

Figure 3: $NDCG$ test accuracy per experiment on DocGII (avg'ed across types).

5.2.1 Analysis on AuthGraph.

We do not have ground truth author ranking in AuthGraph, however, as four areas (DB, DM, IR, and ML) are similar to each other, we consider a Γ with equal ATR values to obtain a ranking by HINSIDE.

We first investigate the relation of the HINSIDE scores

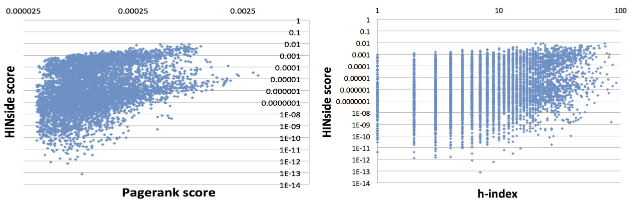

Figure 4: HINSIDE score vs. (a) Pagerank score and (b) h-index of 6619 researchers (dots) in AuthGraph.

with the Pagerank scores on the same graph as well as with the h-index of the researchers, as shown in Figure 4. Roughly speaking, there exists a positive correlation between the measures. Average correlation coefficient across types between HINSIDE and h-index is 0.32, and between HINSIDE and Pagerank it is 0.56. We expect the models to differ; Pagerank solely uses relational information while h-index is based on citation counts (and not collaborations).

We highlight a few example cases in the following table, for which the models differ significantly. The last three columns respectively give the h-index, and the rank order of researchers *within their area* by Pagerank and HINSIDE.

Name	**Area**	**Institution**	**h**	**P**	HIN
Moshe Vardi	DB	Rice U.	87	165	17
Michael R. Lyu	IR	CUHK	67	83	1
Andreas Krause	ML	ETH Zurich	45	291	4

M. Vardi is ranked high by HINSIDE not only because he has high-rank neighbors such as J. Ullman and R. Fagin from the same area, but also due to collaborators across the world, especially several high-rank ones in Italy. The same hold true for M. R. Lyu in Hong Kong. Interestingly, A. Krause is ranked quite high by HINSIDE—One reason is competition: he has high-rank co-authors from CMU, UW, and UC Berkeley. The other reason is distance: all his in-links cover over 4000 miles. These links are before Krause moved from US to Switzerland. While this is a data temporality issue, it is interesting to see HINSIDE's effectiveness in capturing this information.

6 Conclusion

We considered the ranking problem in heterogeneous graphsand proposed HINSIDE, a new model that not only accounts for interactions between different node types, but also uses geo-location information of nodes in a unique way to incorporate (i) the distance of the edges, as well as (ii) the competition induced by location. This formulation is motivated by and generalizes from its application to medical referral networks. We derived the matrix form and a closed form solution for the proposed model. HINSIDE is parameterized by the authority transfer rates between node types. Capitalizing on its closed form, we proposed various estimation algorithms that utilize different objective functions. Experiments on samples of real-world networks demonstrated the effectiveness of our proposed algorithms, and that our model captures more than the network structure that existing models solely rely on.

We share all code and data at https://github.com/abhimm/HINSIDE, for reproducibility and future research.

Acknowledgments: This research is supported in part by NSF CAREER 1452425 and IIS 1408287, ARO YIP Contract W911NF-14-1-0029, DARPA TCP Contract FA8650-15-C-7561, and Facebook and PNC Bank Faculty Gifts. Conclusions expressed in this material are of the authors and do not necessarily reflect the views of the funding parties.

References

[1] A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, 2004.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, 1998.

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005.

[4] D. Cossock and T. Zhang. Subset ranking using regression. In *COLT*, pages 605–619, 2006.

[5] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

[6] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *KDD*, pages 538–543, 2002.

[7] G. Jeh and J. Widom. Scaling personalized web search. In *WWW*, pages 271–279, 2003.

[8] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[9] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In *WWW*, 2005.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[11] R. Y. Rubinstein and D. P. Kroese. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization*. Springer-Verlag New York, Inc., 2004.

[12] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, UK, 1992.

[13] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4(11):992–1003, 2011.

[14] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT*, 2009.

[15] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, pages 797–806. ACM, 2009.

[16] R. von Mises and H. Pollaczek-Geiringer. *Praktische Verfahren der Gleichungsauflösung*. VDI-Verlag.

[17] S. Wasserman and K. Faust. *Social network analysis: Methods and Applications*. Cambridge Univ Pr, 1994.

[18] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR*, pages 271–278, 2007.

Appendix

A Preliminaries

In this short section, we provide background information on ranking in homogeneous networks. Specifically, we describe the essentials of Pagerank [2] as our HINSIDE model carries similar intuitions.

Let $G = (\mathcal{V}, \mathcal{E})$ be a (directed) graph with $|\mathcal{V}| = n$ nodes and adjacency matrix A where $A(i,j) = 1$ if there is an edge $e(i,j) \in \mathcal{E}$ and 0 otherwise. The Pagerank score of a node i is then given based on its (incoming) neighbors as

$$pr_i = \frac{1-\alpha}{n} + \alpha \sum_{j \in \mathcal{V}} A(j,i) \frac{pr_j}{deg_j}$$

where $\alpha \in (0,1)$ is a damping factor and $deg_j = \sum_v A(j,v)$ is the (out)degree of node j.

The above local formula is written in matrix form as

$$\mathbf{pr} = \frac{1-\alpha}{n} \mathbf{1} + \alpha \bar{A}' \mathbf{pr}$$

where $\bar{A}$ is the row-normalized A matrix, i.e., $\bar{A}(i,j) = A(i,j) / \sum_v A(i,v)$, and $\mathbf{1}$ is an all-one vector of length n.

The above can be solved iteratively using the power method [16] starting from a random $\mathbf{pr}^{(0)}$, which can be shown to converge to

$$\mathbf{pr} = \frac{1-\alpha}{n} (I - \alpha \bar{A}')^{-1} \, \mathbf{1}$$

under very common conditions [?], where $\bar{A}'$ needs to be irreducible (i.e., G be strongly connected which holds thanks to the damping factor) and aperiodic (which often holds in practice for real-world graphs).

B Related Work

Ranking is an important data mining task in network analysis. In homogeneous networks, PageRank [2] computes the importance of nodes based on a random walk process. HITS [8] computes both authority and hub scores for each node in a network. There also exist various centrality measures based on degree, betweenness, and closeness [17].

For bipartite networks, in which two types of objects coexist, co-ranking approaches have been proposed. Deng *et al.* propose co-HITS [22] that incorporates a bipartite graph with content information and constraints of relevance. Zhou *et al.* [32] co-rank authors and their publications by coupling two random walk processes. There also exist methods that compute the importance of both nodes and relations in multi-relational networks [24, 26]. These are limited to homogeneous graphs.

On the other hand, ranking in heterogeneous networks has been the focus of research in the last decade. ObjectRank [1] aims to rank objects in a heterogeneous database based on a keyword query. The database is represented as a graph with multiple node and edge types. The adjacency matrix is constructed by weighing the edges according to "authority transfer rates", assumed to be *known*, using which Personalized PageRank is applied, with restart to objects that contain the query keywords.

PopRank [9] extends the PageRank model from page-level to object-level ranking, where relations from both the objects of the same type and those of other types, as well as the popularity of Web pages and databases that contain the object are taken into account. They estimate the authority transfer rates (named as popularity propagation factors) on edges between two object types from training data, using a simulated annealing based search procedure.

Sun *et al.* developed integrated clustering and ranking methods called RankClus [14] and NetClus [15], where objects are first clustered and ranking is done relatively within each cluster. For instance in the DBLP domain, clusters would correspond to different research fields, and ranking researchers or conferences within their field becomes more meaningful (apples to apples). MedRank [21] extended the semantics in [14, 15] to the medical domain, to rank the treatments of a disease based on their influence.

Most recently, Li *et al.* proposed HRank [25], to rank multiple object types and different meta-paths between them in a heterogeneous graph, based on a meta-path based random walk process. For a detailed reference for ranking in networks, we refer to [28].

Different from existing work, our proposed HINSIDE introduces the concepts of (i) location/distance and (ii) competition between the objects, and is the first work to incorporate such side information into the ranking problem in heterogeneous networks. Our model also lends itself to convex formulations for parameter estimation, and consequently to effective and efficient learning procedures.

C Additional Evaluation

C.1 Performance on DocGII. See Figure 5.

C.2 Random Guessing Γ. Randomly guessing the Γ produces results with quite large variance across different experiments. (Note that for DocGI there are only 9 parameters to guess, moreover, we run RG 10 times per experiment and pick the result with the best guess on training data.) This suggests that some ground truth parameter settings were "easy-to-guess", but quite the opposite for others. Figure 6 (top) shows the variation of the guesses in terms of test accuracy per experiment, below which we show the same for RSVM-NN for 10 random initializations (See Alg. 1 line 1). We see that guessing produces notably noisy results, whereas RSVM-NN optimization finds accurate results across different starting points.

As expected, random ordering produces poor results. On the other hand, the rankings by Pagerank and in-weight

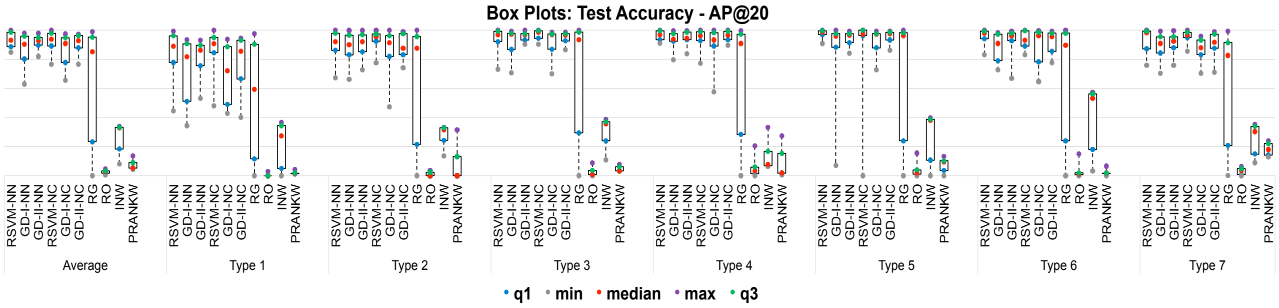

Figure 5: $AP@20$ accuracy on DocGII ($m = 7$) of compared methods.

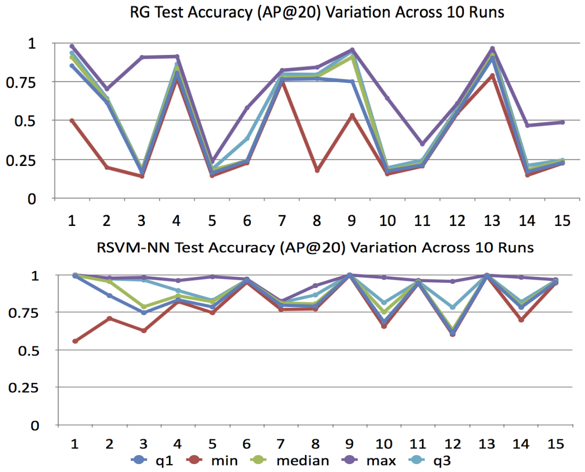

Figure 6: Variation of test accuracy per experiment for 10 random runs of RG and RSVM-NN on DocGI.

centralities are not comparable to the ground truth ranking.

These results imply that there is notably more than the graph structure that the HINSIDE model captures. In other words, models that use structure alone cannot easily reproduce our model.

References

[19] A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, 2004.

[20] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, 1998.

[21] L. Chen, X. Li, and J. Han. Medrank: Discovering influential medical treatments from literature by information network analysis. In *ADC*, pages 3–13, 2013.

[22] H. Deng, M. R. Lyu, and I. King. A generalized co-hits algorithm and its application to bipartite graphs. In *KDD*, pages 239–248. ACM, 2009.

[23] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[24] X. Li, M. K. Ng, and Y. Ye. HAR: Hub, authority and relevance scores in multi-relational data for query search. In *SDM*, pages 141–152, 2012.

[25] Y. Li, C. Shi, P. S. Yu, and Q. Chen. Hrank: A path based ranking framework in heterogeneous information network. *CoRR*, abs/1403.7315, 2014.

[26] M. K.-P. Ng, X. Li, and Y. Ye. Multirank: co-ranking for objects and relations in multi-relational data. In *KDD*, pages 1217–1225, 2011.

[27] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In A. Ellis and T. Hagino, editors, *WWW*, pages 567–574. ACM, 2005.

[28] Y. Sun and J. Han. Ranking methods for networks. In *Encyclopedia of Social Network Analysis and Mining*, pages 1488–1497. 2014.

[29] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *EDBT*, 2009.

[30] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, pages 797–806. ACM, 2009.

[31] S. Wasserman and K. Faust. *Social network analysis: Methods and Applications*. Cambridge Univ Pr, 1994.

[32] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles. Co-ranking authors and documents in a heterogeneous network. In *ICDM*, pages 739–744, 2007.