A Heterogeneous Label Propagation Algorithm for Disease Gene Discovery

TaeHyun Hwang* and Rui Kuang*†

Abstract

Label propagation is an effective and efficient technique to utilize local and global features in a network for semi-supervised learning. In the literature, one challenge is how to propagate information in heterogeneous networks comprising several subnetworks, each of which has its own cluster structures that need to be explored independently. In this paper, we introduce an intutitive algorithm MINProp (Mutual Interaction-based Network Propagation) and a simple regularization framework for propagating information between subnetworks in a heterogeneous network. MINProp sequentially performs label propagation on each individual subnetwork with the current label information derived from the other subnetworks and repeats this step until convergence to the global optimal solution to the convex objective function of the regularization framework. The independent label propagation on each subnetwork explores the cluster structure in the subnetwork. The label information from the other subnetworks is used to capture mutual interactions (bicluster structures) between the vertices in each pair of the subnetworks. MINProp algorithm is applied to disease gene discovery from a heterogeneus network of disease phenotypes and genes. In the experiments, MINProp significantly output-performed the original label propagation algorithm on a single network and the state-of-the-art methods for discovering disease genes. The results also suggest that MINProp is more effective in utilizing the modular structures in a heterogenous network. Finally, MINProp discovered new disease-gene associations that are only reported recently.

Keywords: Label propagation, Random walk, Semi-supervised learning, Data integration, Disease gene prioritization

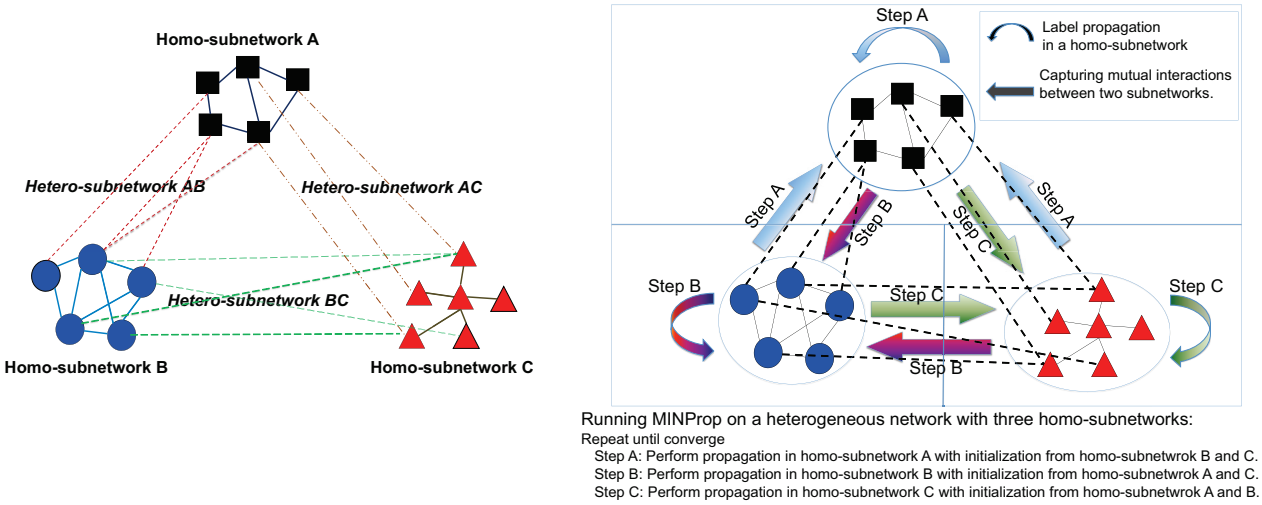
*Department of Computer Science and Engineering, University of Minnesota Twin Cities

†Correspondence: kuang@cs.umn.edu

1 Introduction

Network-based data analysis is recently attracting increasing attention in practical applications of data mining and machine learning [9, 17, 19, 13]. It is believed that many real world networks contain useful hidden neighborhood structures among the objects in the network. Several graph-based semi-supervised learning algorithms have been developed to utilize the global network structure to improve performance in different learning tasks on networks such as classification and ranking [16, 24, 2, 23]. In these algorithms, some of the vertices (labeled data) are initialized with labels or activation values and the other vertices (unlabeled data) are initialized with 0, and the learning problem is to assign label/activation value to the unlabeled data for classification or ranking. The common property of the graph-based semi-supervised learning algorithms is the "cluster assumption": nearby data points in a network should be labeled similarly and data points in the same global cluster in the network should also be labeled similarly. These algorithms are typically formulated as label propagation on a network: the label information on the vertices is iteratively propagated between the neighboring vertices and the propagation process will finally converge toward the unique global optimum minimizing a quadratic criterion [3].

One important problem not previously addressed is how to design an algorithm for propagating label information across several subnetworks of different types of vertices and edges. Under this scenario, several subnetworks describing relations between different objects are present in a heterogeneous network (Fig 1A). Without loss of generality, we define two types of subnetworks in the heterogeneous network: subnetworks containing the same type of objects (*homo-subnetwork*) and bipartite subnetworks with edges connecting two types of objects (*hetero-subnetwork*). Simply propagating the label on the combined network is not a principled method to explore the cluster structures in the network since each homo-subnetwork may have its own cluster structure and each hetero-subnetwork may also have its own bicluster structures (A bicluster is a set of densely connected vertices in the two joint vertex sets in a bipartite graph.). In principle, these cluster and bicluster structures should be explored independently, rather than be-



(A) A heterogeneous network (B) MINProp workflow

Figure 1: **Heterogenous network and MINProp.** (A) This heterogenous network contains three types of vertices, and accordingly three homo-subnetworks and three hetero-subnetworks. (B) Illustration of the MINProp algorithm. Label propagation initialized by the interactions with the other homo-subnetworks is sequentially performed on each individual homo-subnetwork.

ing combined together as a single structure in the combined network, especially when there exist biases among the subnetworks introduced by the heterogeneity in the combined network. The biases can be unbalanced sizes, different noisy levels and different edge-weight scales among the subnetworks. Ignoring the biases can possibly lead to significantly deteriorating performance since some of the independent cluster structures might be lost and cannot be utilized for neighborhood leveraging by label propagation anymore. This example in Fig 1A can be easily generalized to a general heterogenous network with K homo-subnetworks and $K(K-1)/2$ hetero-subnetworks.

In this paper, we introduce a general regularization framework and an efficient algorithm MINProp (Mutual Interaction-based Network Propagation) for propagating information between subnetworks in a heterogeneous network. Instead of treating the subnetworks as parts of the heterogenous network for label propagation, each subnetwork is explored as an independent unit. Graph-based learning is defined on both the homo-subnetworks and the hetero-subnetworks, where the homo-subnetworks are used to capture the cluster structure among the same type of vertices and the hetero-subnetworks are used to capture mutual interactions between the homo-subnetworks. In our regularization framework, the objective is to minimize a convex function of cost terms for smoothness on each individual homo-subnetwork and hetero-subnetwork and fitting to the initial labeling. A novel efficient iterative label propagation algorithm MINProp is then introduced to compute the global optimal solution to the objective function. As illustrated in Fig 1B, MINProp performs label propagations on each individual homo-subnetwork with the current label information derived from the hetero-subnetworks at each step and repeats the step until convergence. The independent network propagations at each step explore the clusters in each homo-subnetwork but uses the label information derived from the hetero-subnetworks to capture mutual interactions (bicluster structures) between each two types of vertices. The MINProp algorithm is essentially an alternating optimization technique [5] for solving convex optimization problems, in which a subset of variables are fixed and optimization is performed on the remaining variables in each iteration. The MINProp algorithm will finally converge on each individual network to the global optimal solution to the convex objective function in the regularization framework.

There are also other attempts to handling learning on heterogeneous networks [21, 10, 22], with which MINProp's regularization shares a similar philosophy. Link Fusion [21] introduced for link analysis shares an almost identical regularization framework with MINProp except the framework is formulated as a random walk with a different normalization. In the Link Fu-

sion model, a heterogeneous random walk is defined and standard PageRank or HITS can be used to derive a solution (stationary distribution or the principal eigenvector) from the heterogenous network. The main difference between MINProp and Link Fusion is that instead of relying on stander techniques running directly on the large random walk matrix, MINProp divides the optimization problem into several correlated sub-problems and performs a sequential procedure on each sub-problem. Thus, MINProp is a more intuitive and interpretable algorithm based on direct learning on each subnetwork. The method by Huang et al. [10] integrates two types of objects, authors and papers, with a marginalized random walk. Ding et al. [22] generalized the method to combine two coupled random walks in the bipartite network with iterations. This method defines the subproblems on bipartite graphs for iterative bi-random walks, which MINProp defines the subproblems on label propagation on the homo-subnetworks. Although the method by Ding et al. [22] might be generalized to compute the same solution for Link Fusion and MINProp, it is a different variation in formulating the sequential learning procedure.

The main methodology contribution of this paper is the new iterative procedure for heterogeneous label propagation. Our focus is on how to design an iterative algorithm that can effectively and intuitively solve a well-defined optimization framework. While computing the unique global optimal solution of the regularization function, the algorithmic form of MINProp provides an interesting interpretation of the learning framework as utilizing mutual interactions between homo-subnetworks as initialization for each round of label propagations. We also used MINProp as a tool to study disease gene discovery from a heterogeneous disease-phenotype and gene network. MINProp achieved very promising improvement over existing approaches. The problem of disease gene discovery will be introduced later in section 4.2.

2 Preliminaries

In this section, we first review the label propagation algorithm on a single network and then introduce the notations for heterogeneous networks.

2.1 Label Propagation on a Single Network

Various graph-based algorithms have been introduced for label propagation on a similarity network [16, 24, 2, 23]. These algorithms simply propagate labels among the neighbors in the network. The propagation repeats until convergence. The set of final label confidence scores on the vertices is the optimal solution to optimizing the quadratic criteria of the semi-supervised learning problem. In this paper, we will base our label propagation algorithm for a heterogeneous network on the iterative algorithm and the regularization framework proposed in [23]. Given an undirected graph $G = (V, E)$, where V is the vertex set and E is the edge set, the similarity matrix W, the initial label y and a diffusion parameter α, Zhou *et al* in [23] proposed the following label propagation algorithm,

1. Normalize W by computing $S = D^{-\frac{1}{2}} * W * D^{-\frac{1}{2}}$, where the diagonal degree matrix D has $D_{ii} = \sum_j W_{ij}$.
2. Choose parameter α and perform propagation, until convergency (t denotes the time step):
$$f^t = (1-\alpha)y + \alpha S f^{t-1}. \tag{2.1}$$
3. The sequence f^t converges to its limit f^* and f^* gives the class labels on the unlabeled vertices.

This algorithm propagates the label information of every vertex to its neighbors in step 2. The propagation process will leverage the label scores of the vertices in a densely connected neighborhood. This algorithm optimizes the following objective function,

$$\Omega(f) = f^T(I-S)f + \mu \parallel f - y \parallel^2, \tag{2.2}$$

where $\mu = \frac{1-\alpha}{\alpha}$. The first term is the *smoothness constraint*, indicating a good classification function should assign similar labels/activation values to strongly connected vertex pairs. The second term is the *fitting constraint*, indicating a good classification function should keep the new label assignment consistent with the initial labeling. This label propagation algorithm is mathematically identical to random walk with restart if W is normalized as $S = D^{-1} * W$. Thus, we call this algorithm Random Walk with restart in this paper.

2.2 Heterogeneous Network

Given a heterogeneous graph $G = (V, E)$ with k homo-subnetworks and $k(k-1)/2$ hetero-subnetworks (see Fig 1A), each homo-subnetwork is defined as $G^{(i)} = (V^{(i)}, E^{(i)})$ and each hetero-subnetwork is defined as $G^{(i,j)} = (V^{(i)} \bigcup V^{(j)}, E^{(i,j)})$. Here, each $E^{(i)}$ is the set of edges between vertices in the vertex set $V^{(i)}$ of homo-subnetwork $G^{(i)}$ and each $E^{(i,j)} \in V^{(i)} \times V^{(j)}$ is the set of edges connecting vertices in $V^{(i)}$ and $V^{(j)}$. Note that $V = \{V^{(1)}, V^{(2)}, ..., V^{(k)}\}$ and $E = \{E^{(1)}, E^{(2)}, ..., E^{(k)}\} \cup \{E^{(1,2)}, E^{(1,3)} ... E^{(k-1,k)}\}$. Let $W^{(i)}$ denote the weight matrix of the homo-subnetwork $G^{(i)}$ and $W^{(i,j)}$ denote the weight matrix of the hetero-subnetwork $G^{(i,j)}$. We normalize $W^{(i)}$ as $S^{(i)} = (D^{(i)})^{-\frac{1}{2}} W^{(i)} (D^{(i)})^{-\frac{1}{2}}$ and $W^{(i,j)}$ as $S^{(i,j)} = (D^{(i,j)})^{-\frac{1}{2}} W^{(i,j)} (D^{(i,j)})^{-\frac{1}{2}}$, where $D^{(i)}$ is the diagonal

matrix with $D^{(i)}_{ll} = \sum_j W^{(i)}_{lj}$ and $D^{(i,j)}$ is the diagonal matrix with $D^{(i,j)}_{ll} = \sum_p W^{(i,j)}_{lp}$.

Next we define graph laplacians on the subnetworks in G. We introduce the graph laplacians in the normalized form, since the unnormalized version is straightforward to derive from the normalized version [23]. Let the normalized graph laplacian matrix of homo-subnetwork $G^{(i)}$ be $\Delta^{(i)} = I - S^{(i)}$, where I is identity matrix. The normalized graph laplacian matrix $\Sigma^{(i,j)}$ of hetero-subnetwork $G^{(i,j)}$ is defined as

$$\Sigma^{(i,j)} = I - \begin{bmatrix} 0 & S^{(i,j)} \\ (S^{(i,j)})^T & 0 \end{bmatrix}.$$

Label propagation associated with the graph laplacian of a single network through a regularization framework [3] ignores the difference among the subnetworks in a heterogeneous network. In a complex heterogeneous network, each subnetwork has a specific graph laplacian that needs to be normalized and explored independently. Thus, a regularization framework on the single network is not appropriate for label propagation on a heterogeneous network.

3 MINProp Algorithm

In this section, we first introduce the MINProp algorithm for propagating information between subnetworks in a heterogeneous network and then develop a regularization framework for MINProp.

3.1 Mutual Interaction-based Propagation

To handle label propagation on a complex heterogeneous network, MINProp sequentially performs network propagations on each individual homo-subnetwork with the current label information derived from the other homo-subnetworks and repeats this step until convergence. The MINProp algorithm performs label propagation on the ith homo-subnetwork $G^{(i)} = (V^{(i)}, E^{(i)})$ sequentially for $i = 1...k$. The label propagation on each homo-subnetwork is the same as that in the algorithms for a single network in Equation (2.1), but the initialization of the vertices in $G^{(i)}$ is a combination of the initial labeling of the vertices and the current labeling of the vertices in the other homo-subnetworks. The labeling information on the other homo-subnetworks is collected as the mutual interactions through $G^{(i,j)} = (V^{(i)} \bigcup V^{(j)}, E^{(i,j)})(1 \le j \le k \text{ and } i \ne j)$, the hetero-subnetworks between the ith homo-subnetwork and the other homo-subnetworks. The mutual interaction information between $G^{(i)}$ and the other homo-subnetworks is collected as

$$\sum_{j \ne i} S^{(i,j)} f_j,$$

where f_j is the current labeling of $V^{(j)}$. The introduction of the labeling information though the hetero-subnetworks can capture the bicluster structures between the vertices in each pair of the subnetworks. The complete MINProp algorithm is described in Algorithm 1.

Algorithm 1 MINProp

Input
k: number of homo-subnetworks
σ: convergence threshold
$y_1, y_2, ..., y_k$: vectors of initial label values
$\alpha_1, \alpha_2, ..., \alpha_k$: diffusion parameters
$S^{(1)}, S^{(2)}, ..., S^{(k)}$: homo-subnetwork matrices
$S^{(1,2)}, ..., S^{(k-1,k)}$: hetero-subnetwork matrices

Output
$f_1, f_2, ..., f_k$: vectors of final label values

1: $f_i = 0$ for $i = 1...k$;
2: **do**
3: $\quad f_i^{old} = f_i$ for $i = 1...k$;
4: $\quad$ **for** $i = 1...k$
5: $\quad\quad t = 0, f_i^0 = 0$;
6: $\quad\quad y' = \frac{1-k\alpha_i}{1-\alpha_i} y_i + \frac{\alpha_i}{1-\alpha_i} \sum_{j \ne i} S^{(i,j)} f_j$;
7: $\quad\quad$ **do**
8: $\quad\quad\quad t = t + 1$;
9: $\quad\quad\quad f_i^t = (1 - \alpha_i) y' + \alpha_i S^{(i)} f_i^{t-1}$;
10: $\quad\quad$ **while**$(\| f_i^t - f_i^{t-1} \| > \sigma)$;
11: $\quad\quad f_i = f_i^t$;
12: $\quad$ **end for**
13: **while** $(\exists i$ **s.t.** $\| f_i - f_i^{old} \| > \sigma)$;
14: **return** $f_1, f_2, ..., f_k$;

The normalized weighted graphs ($S^{(i)}$ and $S^{(i,j)}$) of all homo-subnetworks and hetero-subnetworks are precomputed as described in section 2.2 as inputs. There are three loops in the main body of the MINProp algorithm. The outer do-while-loop between line 2 and line 13 checks if the label values have converged on each of the k homo-subnetworks. The convergence is defined as the 2-norm of the score change after one iteration is less than a threshold σ. The second outer for-loop between line 4 and line 12 sequentially goes through each homo-subnetwork. The inner do-while-loop between line 7 and line 10 is similar to the algorithm in [23]. In line 6, for each vertex in $G^{(i)}$, the initial labeling y' is initialized as the addition of its initial label score and the label of its immediate neighbors in the other homo-subnetworks. The iterative

propagation step at line 9 can be rewritten as

$$f_i^t = (1-\alpha_i)(\frac{1-k\alpha_i}{1-\alpha_i}y_i + \frac{\alpha_i}{1-\alpha_i}\sum_j S^{(i,j)}f_j) + \alpha_i S^{(i)}f_i^{t-1}. \tag{3.3}$$

Equation (3.3) is equivalent to the propagation step in equation (2.1), if $y = \frac{1-k\alpha_i}{1-\alpha_i}y_i + \frac{\alpha_i}{1-\alpha_i}\sum_j S^{(i,j)}f_j$ in equation (2.1). Thus, this step can be thought of as label propagation on a single network with an enriched initialization from the hetero-subnetworks and the proof of convergence is identical to the proof in [23]. Finally, the sequence f_i^t converges to its limit f_i^* and f_i^* gives the class labels/activation values on the vertices in all subnetworks.

The runtime for calculating the initial label values for propagation on homo-subnetwork $G^{(i)} = (V^{(i)}, E^{(i)})$ (line 6) is $O(|V^{(i)}|\sum_{j\neq i}|V^{(j)}|)$. The runtime of the inner do-while-loop between line 7 and 10 is $O(t_i|V^{(i)}|^2)$, where t_i is the number of time steps to reach convergence. These two steps will repeat on each homo-subnetwork (line 4 to line 12), which gives the time complexity $O(\sum_i(|V^{(i)}|\sum_{j\neq i}|V^{(j)}| + t_i|V^{(i)}|^2)$ for finishing one round of propagations on each homo-subnetwork. A more efficient but less intuitive implementation of MINProp can pre-compute $(I-\alpha_i S^{(i)})^{-1}$ and the inner do-while-loop between line 7 and 10 only needs to compute $(1-\alpha_i)(I-\alpha_i S^{(i)})^{-1}y'$ [23]. Let t be the total number of iterations to reach the convergence of MINProp. The total time complexity of the efficient implementation of MINProp algorithm is

$$\begin{aligned} & O(t\sum_i(|V^{(i)}|\sum_{j\neq i}|V^{(j)}| + |V^{(i)}|^2) + t_i|V^{(i)}|^2) \\ = \;& O(t|V|^2 + \sum_i t_i|V^{(i)}|^2). \end{aligned}$$

The time complexity of MINProp crucially depends on the t_is, the convergence rate of propagations on each homo-subnetwork and the t, the number of iterations going through all the homo-subnetworks. The convergence rate of label propagation on each homo-subnetwork closely relates to the property of the spectrum of the graph [3]. In practice, it converges within tens of iterations even on large networks [19, 13, 11]. The iterations going through the homo-subnetworks needed for convergence can be estimated by the theory of alternating optimization [5]. In the next section, we will show that the MINProp algorithm is essentially an alternating optimization algorithm that efficiently calculates the closed-from solution of a convex objective function.

3.2 Regularization Framework

A natural regularization framework for learning on a heterogeneous network $G = (V, E)$ is given as follows,

$$\begin{aligned} \Omega(f) = & \sum_{i=1}^{k}(f_i^T\Delta^{(i)}f_i + \mu_i \parallel f_i - y_i \parallel^2 \\ & + \frac{1}{2}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\mu_{ij}[f_i^T f_j^T]\Sigma^{(i,j)}\begin{bmatrix} f_i \\ f_j \end{bmatrix}), \end{aligned} \tag{3.4}$$

where $f \in R^{|V|}$ is the label variable, and μ_i and μ_{ij} are positive constants that balance the cost terms in the objective function. The cost term $f_i^T\Delta^{(i)}f_i$ is the smoothness constrain on homo-subnetwork $G^{(i)} = (V^{(i)}, E^{(i)})$ that enforces a consistent labeling of the strongly connected vertices in $V^{(i)}$. The cost term $\parallel f_i - y_i \parallel^2$ is a fitting term which keeps the final label values consistent with the initial labels. If we put the above two cost terms together, each $f_i^T\Delta^{(i)}f_i + \parallel f_i - y_i \parallel^2$ is exactly the objective function for learning on $G^{(i)}$ identical to the objective function for learning on a single network in equation (2.2). The last additional cost term $[f_i^T f_j^T]\Sigma^{(i,j)}\begin{bmatrix} f_i \\ f_j \end{bmatrix}$ on the second line is the same smoothness constraint on the hetero-subnetwork $G^{(i,j)} = (V^{(i)}\bigcup V^{(j)}, E^{(i,j)})$ that enforces a consistent labeling between the strongly connected vertex pairs in $V^{(i)}\bigcup V^{(j)}$. The normalized graph laplacian $\Sigma^{(i,j)}$ is defined on the bipartite graph $G^{(i,j)}$ in this case [11]. For simplicity of analysis and implementation, we set all the $\mu_{ij} = 1$, assuming that the homo-subnetworks and hetero-subnetworks are equally informative for learning. Next, we show that the MINProp algorithm actually minimizes the cost function $\Omega(f)$.

PROPOSITION 3.1. *$\Omega(f)$ is strictly convex.*

Proof. Since all $\Delta^{(i)}$ and $\Sigma^{(i,j)}$ are graph laplacians, they are all positive semi-definite [6]. Thus, the cost terms $f_i^T\Delta^{(i)}f_i$ and $[f_i^T f_j^T]\Sigma^{(i,j)}\begin{bmatrix} f_i \\ f_j \end{bmatrix}$ are all convex functions in f. Since $\parallel f_i - y_i \parallel^2$ is also convex in f and μ_i and μ_{ij} are positive constants, $\Omega(f)$ is a non-negative-weighted sum of convex functions. Thus, $\Omega(f)$ is convex. If we take the second derivative of $\Omega(f)$ to obtain its Hessian matrix, the Hessian is the summation of $\Delta^{(i)}$, $\Sigma^{(i,j)}$ and I (the hessian of $\parallel f_i - y_i \parallel^2$). Since $\Delta^{(i)}$ and $\Sigma^{(i,j)}$ are positive semi-definite and I is positive definite, the Hessian of $\Omega(f)$ is positive definite. Hence, $\Omega(f)$ is strictly convex.

PROPOSITION 3.2. *The optimal solution to the alternating optimization step on each f_i in the objective function $\Omega(f)$ is $f_i^* = (1-\alpha_i)(I_i - \alpha_i S^{(i)})^{-1}(\frac{1-k\alpha_i}{1-\alpha_i}y_i + \frac{\alpha_i}{1-\alpha_i}\sum_j S^{(i,j)}f_j)$.*

Proof. By proposition 3.1, $\Omega(f)$ is strictly convex. Thus, it can be minimized with alternating optimization (See [5] for a rigorous proof). Specifically, for each f_i, we fix the f_j for all $j \in \{j|1 \leq j \leq k, j \neq i\}$ and then differentiate $\Omega(f)$ with respect to f_i to compute the closed-form solution f_i^* for minimizing $\Omega(f)$. We take partial derivative of $\Omega(f)$ with respect to f_i, $\frac{\partial \Omega}{\partial f_i}$, and set it to zero,

$$(I_i - S^{(i)})f_i^* + \mu_i(f_i^* - y_i) + (k-1)f_i^* - \sum_j^k S^{(i,j)} f_j = 0$$

Let $\alpha_i = 1/(k+\mu_i)$ and after rearrangement, the closed-from solution f_i^* can be computed as follows,

$$f_i^* = (I_i - \alpha_i S^{(i)})^{-1}((1 - k\alpha_i)y_i + \alpha_i \sum_j S^{(i,j)} f_j).$$

This concludes the proof. Here, $I_i - \alpha_i S^{(i)}$ is positive definite since $0 < \alpha_i < 1$ and the largest eigenvalue of $S^{(i)}$ is 1 [23].

PROPOSITION 3.3. *The MINProp algorithm minimizes the objective function $\Omega(f)$ of the regularization.*

Proof. We have showed that the iteration step between line 4 and line 11 computes the propagation operation defined by equation (3.3), which is identical to equation (2.1) proposed by [23]. Follow the proof by [23], we can show that the sequence $f_i^{(t)}$ converges to $f_i^* = (1-\alpha_i)(I_i - \alpha_i S^{(i)})^{-1}(\frac{1-k\alpha_i}{1-\alpha_i} y_i + \frac{\alpha_i}{1-\alpha_i} \sum_j S^{(i,j)} f_j)$. By Lemma 3.2, we conclude that the iteration step computes the optimal solution to the alternating optimization of each f_i in minimizing $\Omega(f)$. The two outer loops in Algorithm 1 fix f_j for all $j \in \{j|1 \leq j \leq k, j \neq i\}$ to find the optimal f_i sequentially and repeat until converge. Algorithm 1 exactly performs alternating optimization for minimizing $\Omega(f)$. Thus, the MINProp algorithm essentially minimizes the objective function $\Omega(f)$ of the regularization framework.

Note that the Hessian matrix of the MINProp objective function can be rewritten as follows,

$$\begin{bmatrix} S^{(1)} & \mu_{12}S^{(1,2)} & ... & \mu_{1k}S^{(1,k)} \\ \mu_{21}S^{(2,1)} & S^{(2)} & ... & ... \\ ... & ... & ... & ... \\ \mu_{k1}S^{(k,1)} & ... & ... & S^{(k)} \end{bmatrix}$$

This matrix is almost identical to the heterogeneous random walk introduced in Link Fusion [21]. However, our algorithm and proofs are motivated by optimization with an alternating procedure, while the study in Link Fusion showed that the heterogeneous random walk is non-negative and row stochastic, and that the standard algorithms can be applied to the framework. Thus, the primary focus of our work is different, although a common objective is shared. In our proofs, we simplified the parameter selection by assuming a uniform weights on the heterogenous networks. Nevertheless, simple strategies for parameter selection such as the constraint introduced in Link Fusion can be used. More sophisticated optimization techniques such as QCQP might be applied to find the optimal parameters. But it is not straightforward to integrate MINProp with the optimization techniques.

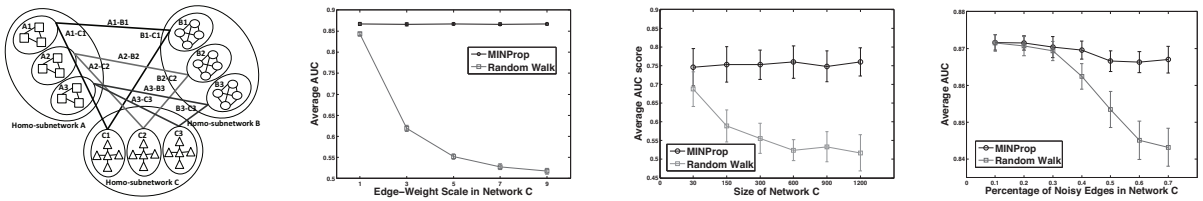
4 Experiments

We evaluated the MINProp algorithm with simulations on artificial datasets and application to disease gene prioritization. To show in the simulations that MINProp can remove biases introduced by network heterogeneity, we compared MINProp with Random Walk with restart [23]. Note that Random Walk with restart ignores the heterogeneity of the subnetworks and simply propagates label information on the combined network. In the experiments on disease gene prioritization, we compared MINProp with Random Walk with restart and CIPHER [20], one of the-state-of-the-art algorithms that explores network information for disease phenotype-gene association discovery.

In all experiments, leave-one-out cross-validation is performed to evaluate the methods. We initialized a query vertex with 1 and all other vertices with 0 before label propagation, and perform this for all the vertices in the subnetwork of query interest. The ranking performance for each query was evaluated by AUC calculated on the ranking of true positives among false negatives by each method. AUC is the normalized area under a curve plotting the number of true positives against the number of false positives by varying a threshold on the decision values. We report the average and a pairwise *win/draw/loss* comparison of the AUC scores of all the queries. The pairwise *win/draw/loss* comparison is the counts of number of times that a method win over or lose to or draw a tie with another method across all the queries. In the leave-one out cross validations, the parameters used in the experiments for MINProp are $\alpha_i \in \{0.1, 0.2, 0.3, 0.4, 0.47\}$, and μ_{ij} is fixed as 1.0. The parameters used in the experiments for Random Walk with restart are $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

4.1 Simulations

To show that MINProp can remove biases introduced from network heterogeneity by exploring the independent cluster structures in each subnetwork, we compared MINProp and Random Walk with restart [23]



(A) Network topology (B) Weighting scales of edges (C) Unbalanced subnetworks (D) Noisy subnetworks

Figure 2: **Simulation results.** (A) The structure of the heterogenous networks used in the simulations. There are three homo-subnetworks and three hetero-subnetworks in the heterogenous network.The three global clusters ($A1 \bigcup B1 \bigcup C1$, $A2 \bigcup B2 \bigcup C2$ and $A3 \bigcup B3 \bigcup C3$) contain cluster members across the three homo-subnetworks. (B)-(D) Ranking performance on heterogenous networks with different biases.

in artificial heterogeneous networks with various settings. We first generated heterogeneous networks with three homo-subnetworks (A, B and C) and three hetero-subnetworks (A-B, A-C and B-C) as plotted in Fig 2A. Each homo-subnetwork has three densely connected clusters (e.g. $A1$, $A2$ and $A3$ in homo-subnetwork A). With probability 0.8, two vertices in the same cluster are connected. Globally, the clusters numbered the same in the three homo-subnetworks are assumed to have the true associations, i.e. there are three global clusters $A1 \bigcup B1 \bigcup C1$, $A2 \bigcup B2 \bigcup C2$ and $A3 \bigcup B3 \bigcup C3$. The vertices in the same global cluster but different homo-subneworks are connected with probability 0.7. To mimic the noisy nature of real networks, random edges connecting vertices in different global clusters are generated with probability 0.3.

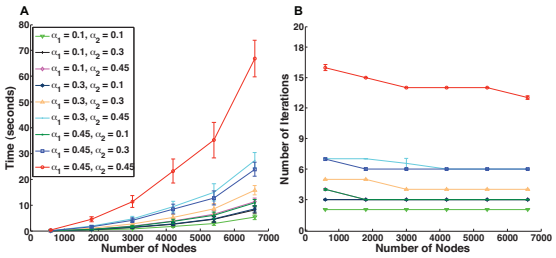
In the simulation, the task is to retrieve the true associations defined by hetero-subnetwork A-B. Specifically, we use each vertex in the homo-subnetwork A as a query and remove the direct links between the query vertex and the vertices in homo-subnetwork B to test whether a method can rank the vertices in the cluster associated with the query vertex in homo-subnetwork B high. For example, if we query with a vertex in cluster $A1$, a good method should rank the vertices in cluster $B1$ above the vertices in $B2$ and $B3$. Both algorithms were tested on 50 randomly generated heterogeneous networks. In the experiment on each heterogeneous network, we perform leave-one-out cross-validation. We calculated the AUC scores from the ranking of the vertices and reported the average AUC scores across all the queries.

4.1.1 Biased edge-weight scales. We fixed the size of the three homo-subnetworks, and varied the edge weights between 1 to 9 in the homo-subnetwork C while assigning constant weight 1 to all other edges. As shown in Fig 2B, when all the subnetworks have the same edge weight 1, MINProp and Random Walk perform similarly well. However, when larger weighting scales are introduced into homo-subnetwork C, the performance of Random Walk deteriorates quickly. The plot indicates that label propagation on a network with heterogeneous edges of different weighting scales can possibly destroy the cluster structures in the subnetworks, but MINProp can properly avoid the scaling problem by performing independent label propagation on each homo-subnetwork.

4.1.2 Unbalanced subnetwork sizes. In this experiment, all the edge weights are set to 1 and the size of subnetwork A and B are set to 30. We started with testing the algorithms with three homo-subnetworks of the same size, and then gradually increased the size of homo-subnetwork C. Interestingly, it is clear in Fig 2C that, as the size of homo-subnetwork C grows, the performance of Random Walk gets worse while MINProp performs stably and robustly well for all the sizes. The result indicates that if the heterogeneous network consists of unbalanced subnetworks, the cluster structures in the larger subnetworks are dominating. Thus, the cluster structure in the smaller subnetworks can diminish.

4.1.3 Noisy subnetworks. We fixed the edge weights and sizes of the three homo-subnetworks, and then introduced different percentage of noisy edges in the hetero-subnetworks A-C and B-C. The noisy edges are introduced as the random connections between the vertices in subnetwork C and the vertices in subnetwork A and B. Fig 2D shows that when all the subnetworks are equally informative, the difference of the performances of the two algorithms is small. However, MINProp clearly outperforms Random Walk when the percentage of noisy edges is high. The plot shows that MINProp is more robust in handling noisy subnetworks

than Random Walk.



(A) Running Time (B) Convergence

Figure 3: **Time complexity of MINProp.** The plots show the scalability and the convergence of MINProp under various sizes of heterogenous networks. Nine different pairs of (α_1, α_2) are tested. The x-axis is the size of the heterogenous network. (A) The y-axis is the running time for reaching the convergence. (B) The y-axis is the number of iterations for reaching the convergence.

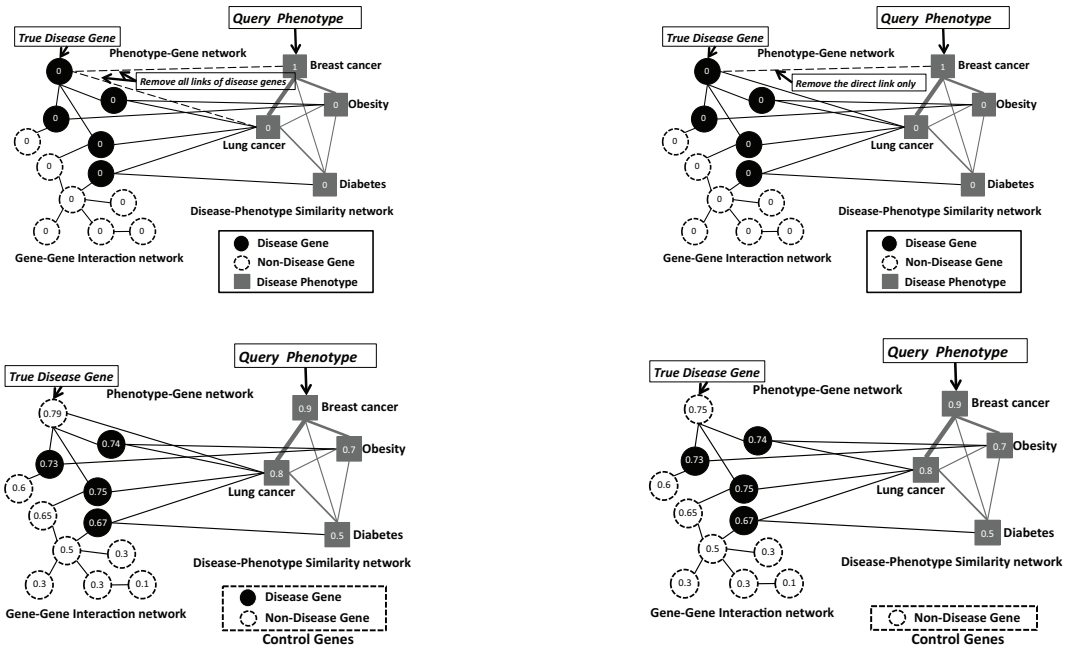
4.1.4 Convergence rate and running time. To test the convergence and the scalability, we measured the convergence rate and the running time of MINProp on artificial heterogenous networks of two homo-subnetwork and one hetero-subnetwork generated in the same setting with the previous simulations. The convergence is defined as the maximum change of activation values over all the graph nodes being smaller than $1e-9$. Nine combinations of the α_i parameters are tested. The running time and the number of iterations for reaching convergence are reported in Fig 3. For all the choices of the parameters, MINProp converged within 16 iterations and the number of iterations are similar on the heterogenous networks of different sizes (Fig 3B). The running time almost scales quadratically in the size of the heterogenous network (Fig 3A).

4.2 Disease Gene Prioritization

Disease phenotype-gene association discovery is one of the principal goals in genomics research for studying diseases [7]. The overall objective is to identify the relation between thousands of complex human disease phenotypes and the susceptive causative genes of the disease phenotypes in human genome. Although a wealth of data, such as gene function annotation databases, protein-protein interactions and disease phenotype and gene association databases, are available, developing a reliable model for integrating the heterogeneous data to identify novel association between disease phenotypes and their causative genes is still a hard problem due to the difficulty in joint-learning from the data of different nature. The learning task is to rank candidate disease genes for each disease phenotype based on a heterogeneous network of three subnetworks, a gene-gene interaction subnetwork, a phenotype-phenotype similarity subnetwork, and a gene-phenotype association subnetwork. With similar settings in [20], experiments on disease gene prioritization are defined as querying with a disease phenotype to rank the candidate disease genes in the gene-gene interaction subnetwork (Fig 4). We compared MINProp with Random Walk with restart adopted from the method proposed by [14] and CIPHER DN (Direct Neighbors) or SP (Shortest Path) [20], two state-of-the-art algorithms that also explore network information for disease gene prioritization. CIPHER [20] is a method proposed for disease phenotype-gene association prediction. Given a query phenotype, the CIPHER algorithm ranks the genes based on the correlation between the direct connectivity of the query phenotype and each gene with the other disease phenotypes. However, CIPHER does not fully explore the cluster structure in the networks.

4.2.1 Data preparation. The three subnetworks in the heterogeneous network were prepared. The gene-gene interaction network was derived from the human protein-protein interaction (PPI) network introduced by Wu et al. [20]. The PPI network contains 34,364 binary-valued undirected interactions between 8919 human proteins (genes). The information of phenotypes and disease genes was extracted from OMIM database (Version May-2007). The disease phenotype similarity network is an undirected graph with 5080 OMIM disease phenotype vertices [8]. The edges are weighted by the pairwise quantitative measurements of the phenotypic overlap in text and clinical synopsis of OMIM records calculated by text mining techniques [18]. The phenotype-gene association network is an undirected bipartite graph with disease phenotype vertices and gene vertices. Binary edges connect 1126 disease phenotype vertices and 916 gene vertices based on the associations in OMIM [8].

4.2.2 Experiment design. The task of disease gene prioritization is, for a given query phenotype, to rank the disease genes associated with the phenotype among a set of control genes. The optimal performance will be ranking the disease genes associated with the phenotype above all the control genes. We designed two experiment settings with different sets of control genes: all other genes (including both the other disease genes and the non-disease genes) or the non-disease genes. The non-disease genes are the genes that have not known association with any disease phenotypes.



(A) Uncovering associations with known disease genes (B) Discovering associations with new disease genes

Figure 4: **Disease gene prioritization** (A) In the experiment of uncovering associations with known disease genes, only the link between the disease gene and the query was removed. The links between the gene and other phenotypes were kept. The final ranking is measured on all the genes in the gene-gene interaction network. (B) In the experiment of discovering new disease susceptive genes, we removed the link between the disease gene of the query and the link between this gene and other phenotypes (the dashed lines connecting the true disease gene of the query with all the phenotypes). The final ranking is only measured on the disease gene plus the non-disease genes. The top figures show the initialization to run MINProp and Random Walk with restart. We first initialize the label score of the query vertex with 1 and the other vertices with zero. The bottom figures shows the results after label propagation converges. After label propagation converges, each vertex has a final label score. We rank the true disease gene of the query phenotype and the control genes according to the label scores assigned by the algorithm. Note that the final label scores in the figure are artificially assigned in this example. This specific example shows the gene ranking task in discovering a disease gene of breast cancer: we assign label score one to breast cancer and zero to the other disease phenotypes and the genes. We remove the edges between the disease gene and breast cancer (or all the phenotypes) from the network. After label propagation converges, we rank the true disease gene of breast cancer with non-disease genes according to the label scores.

Setting 1: Uncovering associations with known disease genes. The first setting is to discover new phenotype-gene associations between the query phenotype and all the genes including both the non-disease genes and the disease genes (Fig 4A). This experiment is a comprehensive evaluation of how well a method can discover phenotype-gene associations for a given new disease phenotype without any known disease genes. Specifically, we want to discover novel associations between a query phenotype and all the genes including the disease genes that already have associations with the other disease phenotypes. Thus, in the leave-one-out cross-validation, we only removed the direct link between the query phenotype and its disease genes but kept the links between its disease genes and other phenotypes. The disease genes of the query phenotypes were then ranked among all the genes including both disease genes and non-disease genes. The learning task is to rank the disease genes of the query phenotype as high as possible above all other genes in the rank list.

Setting 2: Discovering associations with new disease susceptibility genes. The second setting is to examine how well a method can discover the disease susceptibility genes of each query phenotype from the non-disease genes (Fig 4B). The 8003 non-disease genes among the 8919 genes were selected as the control genes. In leave-one-out cross-validation of querying with a

disease phenotype, we removed the links between its disease genes and all the phenotypes including the query phenotype. This is equivalent to adding the disease genes of the query phenotype as unknown disease genes to the 8003 non-disease genes without any selection bias. The learning task is to rank the disease genes of the query phenotypes as high as possible above the non-disease genes in the rank list.

In the first setting, we focus on finding missing phenotype-gene associations between phenotypes and genes. The target genes are allowed to keep their associations with other phenotypes. Therefore, they will compete against other disease genes in the rank list. In the second setting, we focus more on finding new disease susceptibility genes from non-disease genes. Thus, the target genes are not linked with any phenotypes, and the question is whether we can discover this type of hidden disease genes (the disease genes of the query phenotype) without knowing any associations between the target genes and all disease phenotypes. The purpose of introducing the two different settings is to give a comprehensive evaluation of the methods in scenarios with different network connectivities on the target genes.

4.2.3 Performance of ranking disease genes in leave-one-out cross-validation. The 5080 disease phenotypes and the 1126 phenotype-gene associations extracted from OMIM version May-2007 were tested in leave-one-out cross-validation under the two settings. We performed leave-one-out cross validation by holding-out one query phenotype for testing at a time. The ranking performances of all the methods in the two settings are measured by AUC scores calculated based on the ranking of the true disease genes among the control genes determined by each method. One limitation of CIPHER DN is that only genes with at least one disease gene in its direct neighbors can be ranked with the other genes. Thus, for some of the query phenotypes, its disease genes cannot be ranked by CIPHER DN in leave-one-out cross-validation. After filtering of the query phenotypes with causative disease genes that have no direct neighbor to the other disease genes, 858 disease phenotypes were left for evaluation in the comparison with CIPHER DN.

The average AUCs of gene ranking across all the queries by MINProp, Random Walk with restart and CIPHER are reported in Table 1. In the first setting, MINProp outperformed CIPHER DN by 12.5%, CIPHER SP by 7.1% and Random Walk by only 0.8%. In the second setting, MINProp outperformed Random Walk by 8% and CIPHER DN by 7.3%, and achieved a

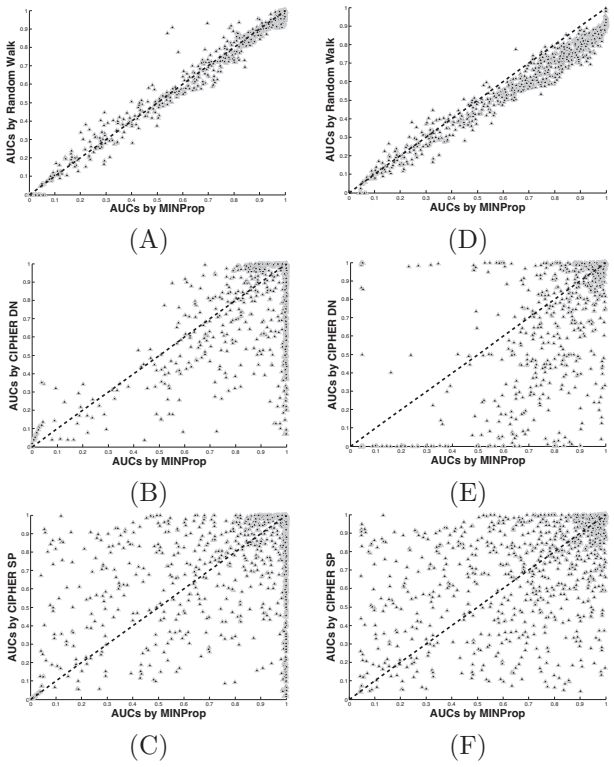


Figure 5: **Query-wise scatter plot of the AUC scores in leave-one-out cross-validation.** In the scatter plots, the x-axis and the y-axis of a dot are the AUCs on a query phenotype by MINProp and the compared method, respectively. (A)-(C) The three figures on the left are from the experiments of uncovering associations with known disease genes. (D)-(F) The right figures are from the experiments on discovering new disease susceptibility genes from unknown disease genes.

tie with CIPHER SP. The pairwise comparisons of the AUC scores in Table 1 suggest that MINProp improved the ranking of more query cases compared with the other methods, except the comparison with CIPHER SP in the second setting. A more detailed examination of the individual query cases is given in Fig 5. Interestingly, MINProp can win over Random Walk in about 66% and 93% queries in the two experiments but only by a relative small margin — most of the dots in the scatter plot in Fig 5A and Fig 5D are just below the diagonal line. Since both methods are exploring the global structures in the network, it is not supervising that the results are strongly correlated. But clearly, by exploring the independent structures in the subnetworks, MINProp was able to improve most of the predictions. In the comparison with CIPHER DN in Fig 5B and Fig 5E, MINProp and CIPHER DN produced quite

Table 1: **Disease gene prioritization performance in leave-one-out cross-validation.** This table reports the average AUC scores and the pairwise *win/draw/loss* comparisons between MINProp and the baselines, Random Walk (RW) and CIPHER DN or SP, across the query phenotypes in the two settings.

Methods	Associations with known disease genes Avg. AUC *(win/draw/loss)*	Associations with new disease genes Avg. AUC *(win/draw/loss)*
MINProp vs. Random Walk	**0.805** vs. 0.797 (738/75/313)	**0.728** vs. 0.648 (1045/2/79)
MINProp vs. CIPHER-DN	**0.863** vs. 0.738 (565/5/288)	**0.821** vs. 0.738 (515/11/332)
MINProp vs. CIPHER-SP	**0.805** vs. 0.734 (678/8/440)	0.728 vs. 0.729 (538/54/534)

different results in most of the queries. MINProp improved on many hard cases, on which CIPHER DN and SP performed poorly. This suggests that for many cases, it is not enough to just check the second-order neighbors of the genes, and it is important to explore the global structures.

MINProp achieved the best overall performance in the experiments. The explanation of the tie between MINProp and CIPHER SP in the second setting needs more analysis. CIPHER SP also explores the global structure in the gene-gene interaction network since the gene-gene connection was evaluated by shortest paths, which measure remote interaction between the genes in the network. In the second setting, because no target genes are linked with the phenotypes, the cluster structures in the phenotype similarity network only have small influence. Thus, the ranking of the disease genes relies more on the bi-clusters between the genes and the phenotypes. In this case, CIPHER SP can possibly perform better on the disease genes that are directly connected to other disease genes, while MINProp might dilute this direct information with neighborhood averaging. To show that MINProp and CIPHER SP can be complementary to each other, we further combined the gene ranking produced with MINProp and CIPHER SP by averaging the ranks of each gene in the two lists. A 3% improvement is observed in the hybrid case (last column in Table 2).

4.2.4 Exploring the modularity of genes. To evaluate how effective the methods can explore the modular structures in the gene-gene interaction network, we analyzed the ranking results of the disease genes with their involvement in the modular structures of the gene-gene interaction network in the second setting. The modularity of the disease genes of each query phenotype is measured by their cluster coefficients in the gene-gene interaction network. Large averages of cluster coefficients of the disease genes indicate high modularity-involvement of the disease genes. In Table 2, we compare the ranking performance on the query phenotypes with respect to different levels of gene cluster coefficients. Clearly, MINProp outperforms the baselines on the test queries of higher gene cluster coefficients. Since CIPHER DN also utilizes the modular structure of genes by counting gene neighbors, the trend of larger cluster coefficients suggesting larger performance difference between MINProp and CIPHER DN is not obvious. But overall, MINProp significantly improved disease gene ranking compared to CIPHER DN in all ranges of gene cluster coefficients. The analysis indicates that MINProp is more capable of effectively utilizing the modular structures in the gene-gene interaction network, and our framework might be a more robust and powerful method for the gene prioritization task on the heterogeneous network. Finally, the hybrid of MINProp and CIHPER SP can boost the performance of MINProp on the queries of small gene cluster coefficients to be similar to CIPHER SP.

4.2.5 Inferring novel disease genes. MINProp also identified recently discovered disease genes, TRAK2, MYH13, PRNP, MAPT and CHRNB2 as well as the well known APOE, of Alzheimer disease in the gene cluster associated with the neurological diseases [4]. These identified genes do not have associations with Alzheimer disease in the OMIM database, but the associations have been recently confirmed [4]. These findings suggest that exploring independent cluster structures of disease phenotype and gene-gene networks can reveal new causal relations between disease phenotypes and genes.

5 Discussion

In this paper, we introduce MINProp for label propagation on heterogeneous networks. The experiments demonstrated that MINProp could effectively explore independent cluster structures in each subnetwork to remove the biases introduced by the heterogeneity of a complex network of several different subnetworks. With the advent of high-throughput bio-technologies, there are many challenging problems requiring integration of large scale genomoic datasets in biomedical informatics and bioinformatics applications. MINProp is a general, robust and efficient algorithm for data integration in these applications. The time and space complexity of MINProp is in the same order as other standard graph-based learning algorithms – quadratic in the number of

Table 2: **Ranking performance on phenotypes with different gene cluster coefficients.** The table compares MINProp with RW, CIPHER DN & SP and Hybrid (Combined ranking of MINProp and CIPHER SP).

CC	MINProp vs. RW Avg. AUC	MINProp vs. CIPHER-DN Avg. AUC	MINProp vs. CIPHER-SP Avg. AUC	Hybrid vs. CIPHER-SP Avg. AUC
$[0.1, 1]$	**0.875** vs. 0.776	**0.889** vs. 0.855	**0.875** vs. 0.813	**0.886** vs. 0.813
$[0.01, 0.1)$	**0.902** vs. 0.799	**0.906** vs. 0.799	**0.902** vs. 0.801	**0.911** vs. 0.801
$[0, 0.01)$	**0.653** vs. 0.586	**0.770** vs. 0.688	0.654 vs. **0.693**	0.692 vs. 0.693
Total	**0.728** vs. 0.648	**0.821** vs. 0.738	0.728 vs. 0.729	**0.756** vs. 0.729

vertices multiplied by the number of iterations. Scalability is still a limitation since heterogenous networks typically contain more vertices in several subnetworks and thus, requirement on time and space complexity is often more stringent. In future, we plan to design a sparse implementation of MINProp, given the sparseness of some of the subnetworks. Another limitation is the difficulty in tuning the hyper-parameters for balancing the subgraph laplacians. Thus, we also aim to apply other optimization techniques allowing quadratic constraints to learn the hyper-parameters [1, 12, 15].

Acknowledgement

We thank Ze Tian for help on graphics.

References

[1] A. Argyriou, M. Herbster, and M. Pontil. Combining graph laplacians for semi-supervised learning. In *NIPS*, pages 67–74. MIT Press, 2005.

[2] M. Belkin and P. Niyogi. Using manifold structure for partially labelled classification. In *NIPS*, 2003.

[3] Y. Bengio, O. Delalleau, and N. L. Roux. Label propagation and quadratic criterion. In E. O. Chapelle, B. Schlkopf, and A. Zien, editors, *Semi-Supervised Learning*. MIT Press, 2006.

[4] L. Bertram and R. E. Tanzi. Thirty years of alzheimer's disease genetics: the implications of systematic meta-analyses. *Nat Rev Neurosci*, 9(10):768–778, 2008.

[5] J. C. Bezdek and R. J. Hathaway. Convergence of alternating optimization. *Neural, Parallel Sci. Comput.*, 11(4):351–368, 2003.

[6] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, February 1997.

[7] N. Freimer and C. Sabatti. The human phenome project. *Nat Genet*, 34:15–21, 2003.

[8] A. Hamosh, A. Scott, J. Amberger, D. Valle, and V. McKusick. Online mendelian inheritance in man (OMIM). *Hum Mutat*, 15(1):57–61, 2000.

[9] J. He, M. Li, H. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *ACM Multimedia*, pages 9–16, 2004.

[10] J. Huang, T. Zhu, R. Rereiner, D. Zhou, and D. Schuurmans. Information marginalization on subgraphs. In *PKDD*, pages 199–210, 2006.

[11] T. Hwang, H. Sicotte, Z. Tian, B. Wu, J.-P. Kocher, D. A. Wigle, V. Kumar, and R. Kuang. Robust and efficient identification of biomarkers by classifying features on graphs. *Bioinformatics*, 24(18):2023–2029, September 2008.

[12] T. Kato, H. Kashima, and M. Sugiyama. Integration of multiple networks for robust label propagation. In *SDM*, pages 716–726. SIAM, 2008.

[13] R. Kuang, J. Weston, W. S. Noble, and C. Leslie. Motif-based protein ranking by network propagation. *Bioinformatics*, 21(19):3711–3718, 2005.

[14] S. Khler et al. Walking the interactome for prioritization of candidate disease genes. *The American J of Hum Genet*, 82(4):949 – 958, 2008.

[15] G. R. G. Lanckriet, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based integration of genomic data using semidefinite programming. In *Kernel Methods in Computational Biology*, pages 209–231. 2004.

[16] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *NIPS*, pages 945–952, 2001.

[17] K. Tsuda, H. Shin, and B. Schölkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21:59–65, 2005.

[18] M. van Driel, J. Bruggeman, G. Vriend, H. Brunner, and J. Leunissen. A text-mining analysis of the human phenome. *Eur J Hum Genet*, 14:535–542, 2006.

[19] J. Weston, A. Elisseeff, D. Zhou, C. Leslie, and W. S. Noble. Protein ranking: from local to global structure in the protein similarity network. *Proc Natl Acad Sci USA*, 101(17):6559–63, 2004.

[20] X. Wu, R. Jiang, M. Q. Zhang, and S. Li. Network-based global inference of human disease genes. *Mol Syst Biol*, 4, 2008.

[21] W. Xi, B. Zhang, Z. Chen, Y. Lu, S. Yan, W.-Y. Ma, and E. A. Fox. Link fusion: a unified link analysis framework for multi-type interrelated data objects. In *WWW*, pages 319–327, New York, NY, USA, 2004. ACM Press.

[22] 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, Washington, DC, USA, 2007. IEEE Computer Society.

[23] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schöelkopf. Learning with local and global consistency. In *NIPS*, volume 16, pages 321–328, Cambridge, MA, 2004.

[24] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.