Link Prediction with Cardinality Constraint

Jiawei Zhang[⋆], Jianhui Chen[†], Junxing Zhu[‡], Yi Chang[♯] and Philip S. Yu[⋆,¶]
[⋆]University of Illinois at Chicago, Chicago, IL, USA
[†]Yahoo Research, Sunnyvale, CA, USA
[‡]National University of Defense Technology, Changsha, Hunan, China
[♯]Search Lab, Huawei Research America, CA, USA
[¶]Institute for Data Science, Tsinghua University, Beijing, China
jzhan9@uic.edu, jianhui@yahoo-inc.com, zhujunxing123@gmail.com, yi.chang@huawei.com, psyu@cs.uic.edu

ABSTRACT

Inferring the links among entities in networks is an important research problem for various disciplines. Depending on the specific application settings, the links to be inferred are usually subject to different *cardinality constraints*, like *one-to-one*, *one-to-many* and *many-to-many*. However, most existing research works on link prediction problems fail to consider such a kind of constraint. In this paper, we propose to study the link prediction problem with general *cardinality constraints*, which is formally defined as the **C**LP (Cardinality Constrained Link Prediction) problem. By minimizing the projection loss of links from feature vectors to labels, the **C**LP problem is formulated as an optimization problem involving multiple variables, where the *cardinality constraints* are modeled as mathematical constraints on node degrees. The objective function is shown to be not jointly convex and the optimal solution subject to the *cardinality constraints* can be very time-consuming to achieve. To solve the optimization problem, an iterative variable updating based link prediction framework ITERCLIPS (Iterative Constrained Link Prediction & Selection) is introduced in this paper, which involves the steps on link updating and selection alternatively. To overcome the high time cost problem, a greedy link selection step is introduced in this paper, which picks links greedily while preserving the link *cardinality constraints* simultaneously. Meanwhile, to ensure the effectiveness of ITERCLIPS on large-scale networks, a distributed implementation of ITERCLIPS is further presented as a scalable solution to the **C**LP problem. Extensive experiments have been done on three real-world network datasets with different types of *cardinality constraints*, and the experimental results achieved by ITERCLIPS on all these datasets can demonstrate the effectiveness and advantages of ITERCLIPS in solving the **C**LP problem.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

Keywords

Link Prediction; Cardinality Constraint; Social Networks; Data Mining

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom

DOI: http://dx.doi.org/10.1145/3018661.3018734

1. INTRODUCTION

In the real-world, information entities are extensively connected to each other via various kinds of links, which together will form different categories of information networks in different disciplines. Depending on the application settings, the physical meaning of links in information networks can be quite diverse. For instance, the co-author links in bibliographic networks denote the *collaboration relations* among researchers [18]; the supervision links among the employees in enterprise organizational charts denote the *management relations* from managers to subordinates [26]; while the anchor links connecting users across different social platforms represent the corresponding relationships between accounts of the same users [28]. What's more, according to the specific physical meanings, these different kinds of links introduced above are usually subject to different *cardinality constraints*. Cardinality constraint is one of the most important constraints in link conceptual modeling. In addition to constraining the population of links, cardinality constraints help illustrate the meaning of the link types, and they also play an important role in steering the link formation among information entities.

Three most common *cardinality constraints* that links (e.g., the co-author links, supervision links and anchor links) are subject to are provided as follows:

- *Many-to-Many Constraint*: In traditional bibliographic networks [18], researchers can collaborate with each other to carry out research projects and write academic papers. Generally, there are no limitations on the number of co-authors that researchers can work with. Viewed in this perspective, the *cardinality constraint* on co-author links in bibliographic networks is *many-to-many*.
- *One-to-Many Constraint*: In real-world companies, the management relationships among the employees can be represented as the enterprise *organizational charts* involving employees and the directed *supervision links* from managers to subordinates. Generally, for companies employing tree-structured organizational charts, each employee only needs to report to one single manager; meanwhile, each manager can supervise multiple subordinates at the same time. Therefore, the *cardinality constraint* on the *supervision links* among the employees in organizational charts is *one-to-many* [26].
- *One-to-One Constraint*: Across different online social networks, the correspondence relationships between the shared users are formally denoted as the bi-directional *anchor links* [28]. According to the existing works [28], each user in an

online social network is assumed to be connected by one *anchor link* with one user from another social network. In other words, the *cardinality constraint* on *anchor links* across different social sites is *one-to-one*.

Formally, given the screenshot of an information network with partially observed links, inferring the other links that are hidden or to be formed in the future in the network is defined as the *link prediction* problem in existing works [13, 28]. Link prediction is an important research problem and can be the basis of many concrete real-world applications. For instance, the *co-author recommendation* [18], *organizational chart inference* [26] and *network alignment or user resolution* [11, 24, 25] problems can be modeled as the prediction of *co-author link* [18], *supervision link* [26, 20, 27], *anchor link* [11, 28] respectively. Link prediction problems have been studied for many years, and dozens of link prediction methods have been proposed already. However, most of the existing link prediction models treat the prediction of different link instances to be totally independent, and very few of them has ever considered the inherent *cardinality constraints* on the links. To gain a more comprehensive knowledge about existing link prediction works, please refer to the survey paper [22, 16, 8] for more information

Problem Studied: In this paper, we will study the link prediction problem in information networks with the *cardinality constraints*. Formally, the problem is defined as the "Cardinality Constrained Link Prediction" (**C**LP) problem. By incorporating the *cardinality constraints* in the link prediction problem modeling, we claim that "*the link prediction results can be improved significantly*". The **C**LP problem is a novel problem, and we are the first to propose the link prediction problem with general cardinality constraints.

The **C**LP problem is very challenging to address due to following reasons:

- *Cardinality Constraint Modeling*: Very few existing works have ever considered the *cardinality constraints* on links before. How to effectively incorporate such a kind of constraint in the link prediction problem modeling is still an open problem to this context so far.

- *Computational Difficulty*: The **C**LP problem itself is shown to be very computationally time-consuming to obtain the optimal results, and the sub-problem involved in **C**LP is shown to be NP-hard. New efficient approximation solutions which can be achieved in polynomial time (close to linear or quadratic time) are desired.

- *Scalable Framework*: For large-scale networks, storing the complete information network data within one single machine can be technically infeasible. A distributed implementation of the link prediction framework applicable to large-sized networks is another great challenge.

To overcome all the above challenges, a new unified link prediction framework ITERCLIPS (Iterative Constrained Link Prediction & Selection) is proposed in this paper. In the proposed framework, the **C**LP problem is formulated as an optimization problem, where the *cardinality constraints* on links are modeled as mathematical constraints on node degrees. The non-convex optimization function involves *multiple variables* as well as hard *cardinality constraints* simultaneously, which render the problem very hard to address. ITERCLIPS adopts an alternative updating schema to solve the objective function, and an approximated greedy method is applied to select promising links from networks subject to the *cardinality constraints* with very low time costs. Finally, a distributed implementation of ITERCLIPS is further presented as a scalable solution to the **C**LP problem.

The following part of this paper is organized as follows. In Section 2, we will define some important terminologies used in this paper and introduce the formulation of the **C**LP problem. The link prediction framework ITERCLIPS will be introduced in Section 3, and its performance will be evaluated in Section 4. Finally, in Section 5, we will talk about the related works and conclude this paper in Section 6.

2. TERMINOLOGY DEFINITION AND PROBLEM FORMULATION

In this section, we will first define the terminologies used in this paper and then provide the formulation of the **C**LP problem.

2.1 Terminology Definition

The problem **C**LP studied in this paper is a general link prediction problem, where the target network can be *bibliographic networks*, *enterprise organizational charts* and *multiple partially aligned social networks*. To be general, we can define the target network structure as an *information network*.

Definition 1 (Information Network): Formally, the *information network* involving information entities $\mathcal{V}$ and known links $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ can be represented as a graph $G = (\mathcal{V}, \mathcal{E})$. The links in information network G are subject to certain *cardinality constraint* depending on the problem setting.

Information network is a general representation about the structured data studied in different disciplines. Specifically, when studying the anchor link prediction problem, the entity set $\mathcal{V}$ will be decomposed into two disjoint subsets $\mathcal{V}_1 \cup \mathcal{V}_2$, and the anchor links merely exist between the entities in $\mathcal{V}_1$ and $\mathcal{V}_2$ (i.e., $\mathcal{E} \subset \mathcal{V}_1 \times \mathcal{V}_2$). To denote the *cardinality constraints* on links in information networks, we will provide its formal definition as follows.

Definition 2 (Cardinality Constraint): Formally, the *cardinality constraint* on links specifies the the number of links that information entities can be associated with. Formally, in this paper, given the information network $G = (\mathcal{V}, \mathcal{E})$, the *cardinality constraint* on link $\mathcal{E}$ can be represented as $M : N$, where M and N denote the *maximum* (which can also be the *mandatory* or *minimum*) *cardinality* of links going out from/into entities in set $\mathcal{V}$.

The above definition provides a general representation of the *cardinality constraint* on links, which can be applied to all the three different kinds of *cardinality constraints* introduced in Section 1.

- *One-to-One Constraint*: $1:1$ on *anchor links* is a *maximum-maximum constraint*, where users can either be connected by anchor links or stay unconnected if they have no accounts in the other network.

- *One-to-Many Constraint*: $N:1$ on *supervision links* is a *maximum-mandatory constraint*, where managers can simultaneously supervise multiple subordinates while employees will be supervised by exactly one manager.

- *Many-to-Many Constraint*: $N:N$ on *co-author links* is also a *maximum-maximum constraint*, where researchers usually work with many other researchers at the same time and N usually equals to $|\mathcal{V}| - 1$ if no specific value is provided in the constraint definition.

2.2 Problem Formulation

Problem Definition: Given the information network $G = (\mathcal{V}, \mathcal{E})$ with the known link set $\mathcal{E}$, we can represent all the potential links

among the information entities as $\mathcal{L} = \mathcal{V} \times \mathcal{V} \setminus \{(u,u)\}_{u \in \mathcal{V}}$. Meanwhile, the set of the unknown links in graph G can be denoted as set $\mathcal{L} \setminus \mathcal{E}$. For links in $\mathcal{L}$, we propose to assign them with labels from set $\mathcal{Y} = \{0, 1\}$ according to the existing works [7], where the links will be formed are assigned with label $+1$ (i.e., positive instances), while those will never be formed are assigned with label 0 (i.e., negative instances). Since we have no idea about the formation likelihood of links in the unknown link set $\mathcal{L} \setminus \mathcal{E}$, which actually involves both positive and negative instances (i.e., links to be formed and those will never be formed), the unknown links should be unlabeled as introduced in [28]. In this paper, we will treat the **C**LP as a PU (positive and unlabeled) learning problem. Based on the set of positive links $\mathcal{E}$ and unknown links $\mathcal{L} \setminus \mathcal{E}$ as well as the information about these links available in the network G, we aim at building a mapping $f : \mathcal{L} \setminus \mathcal{E} \to \{0, 1\}$ to project the links $\mathcal{L} \setminus \mathcal{E}$ to their corresponding labels in this paper.

3. PROPOSED METHOD

In this section, we will first introduce the notations used in this paper. After that, in Section 3.2, we will talk about the loss function for the link prediction problem. The link *cardinality constraint* will be talked about in Section 3.3, and we will introduce the optimization objective function of **C**LP as well as the ITERCLIPS framework in Section 3.4. Detailed analysis about the **C**LP problem and framework ITERCLIPS will be available in Section 3.5. Finally, we will introduce the distributed implementation of the ITERCLIPS framework in Section 3.6.

3.1 Notations

In the sequel, we will use lower case letters (e.g., x) to denote scalars, lower case bold letters (e.g., $\mathbf{x}$) to denote column vectors, bold-face upper case letters (e.g., $\mathbf{X}$) to denote matrices, and upper case calligraphic letters (e.g., $\mathcal{X}$) to denote sets. Given a matrix $\mathbf{X}$, we denote $\mathbf{X}(i,:)$ (and $\mathbf{X}(:,j)$) as the i_{th} row (and the j_{th} column) of $\mathbf{X}$, and the (i_{th}, j_{th}) entry of matrix $\mathbf{X}$ can be denoted as $X(i,j)$ or $X_{i,j}$ (which are interchangeable). We use $\mathbf{X}^\top$ (and $\mathbf{x}^\top$) to denote the transpose of matrix $\mathbf{X}$ (and vector $\mathbf{x}$). For vector $\mathbf{x}$, we denote its L_p-norm as $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$, and the L_p-norm of matrix $\mathbf{X}$ can be represented as $\|\mathbf{X}\|_p = (\sum_{i,j} |X(i,j)|^p)^{\frac{1}{p}}$.

3.2 Link Prediction via Loss Minimization

Based on the whole link set $\mathcal{L}$, a set of features can be extracted for these links with the information available in the information network G, which can be represented as set $\mathcal{X} = \{\mathbf{x}_l\}_{l \in \mathcal{L}}$ ($\mathbf{x}_l \in \mathbb{R}^m, \forall l \in \mathcal{L}$). Given the link existence label set $\mathcal{Y} = \{0, 1\}$, as introduced in Section 2, the objective of the **C**LP problem studied in this paper is to achieve a general link inference function $f : \mathcal{X} \to \mathcal{Y}$ to map the link feature vectors to their corresponding labels. Depending on the specific application setting and information available in the networks, the feature vectors extracted for links in $\mathcal{L}$ can be very diverse. We will not focus on the specific extracted features here, which will be briefly talked about in Section 4 when introducing the detailed experimental settings.

Formally, the loss introduced in the mapping $f(\cdot)$ can be represented as function $L : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ over the link feature vector/label pairs. Meanwhile, for one certain input feature vector $\mathbf{x}_l$ for link $l \in \mathcal{L}$, we can denote its inferred label introducing the minimum loss as $\hat{y}_l$:

$$\hat{y}_l = \arg \min_{y_l \in \mathcal{Y}, \mathbf{w}} L(\mathbf{x}_l, y_l; \mathbf{w}),$$

where vector $\mathbf{w}$ denotes the parameters involved in the mapping function $f(\cdot)$.

Therefore, given the pre-defined loss function $L(\cdot)$, the general form of the objective mapping $f : \mathcal{X} \to \mathcal{Y}$ parameterized by vector $\mathbf{w}$ can be represented as:

$$f(\mathbf{x}; \mathbf{w}) = \arg \min_{y_l \in \mathcal{Y}} L(\mathbf{x}, y; \mathbf{w}).$$

In many cases (e.g., when the links are not linearly separable), the feature vector $\mathbf{x}_l$ of link l needs to be transformed as $g(\mathbf{x}_l) \in \mathbb{R}^k$ (k is the transformed feature number) and the transformation function $g(\cdot)$ can be different *kernel projections* depending on the separability of instances. Throughout this paper, we assume loss function $L(\cdot)$ to be linear in some combined representation of the transformed link feature vector $g(\mathbf{x}_l)^\top$ and label y_l, i.e.,

$$L(\mathbf{x}_l, y_l; \mathbf{w}) = (\langle \mathbf{w}, g(\mathbf{x}_l) \rangle - y_l)^2 = (\mathbf{w}^\top g(\mathbf{x}_l) - y_l)^2.$$

Furthermore, based on all the links in the network $\mathcal{L}$, we can represent the extracted feature vectors for these links to be matrix $\mathbf{X} = [g(\mathbf{x}_{l_1}), g(\mathbf{x}_{l_2}), \cdots, g(\mathbf{x}_{l_{|\mathcal{L}|}})]^\top \in \mathbb{R}^{|\mathcal{L}| \times k}$ (for simplicity, linear kernel projection is used in this paper, and $g(\mathbf{x}_l) = \mathbf{x}_l$). Meanwhile, their existence labels can be represented as vector $\mathbf{y} = [y_{l_1}, y_{l_2}, \cdots, y_{l_{|\mathcal{L}|}}]^\top$, where $y_l \in \{0, 1\}, \forall l \in \mathcal{L}$. Specifically, for the existing links in $\mathcal{E}$, we know their labels to be positive in advance, i.e., $y_l = 1, \forall l \in \mathcal{E}$. According to the above loss function definition, based on $\mathbf{X}$ and $\mathbf{y}$, we can represent the loss introduced by all links in $\mathcal{L}$ to be

$$L(\mathbf{X}, \mathbf{y}; \mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2.$$

To learn the parameter vector $\mathbf{w}$ and infer the potential label vector $\mathbf{y}$, we propose to minimize the loss term introduced by all the links in $\mathcal{L}$. Meanwhile, to avoid overfitting the training set, besides minimizing the loss function $L(\mathbf{X}, \mathbf{y}; \mathbf{w})$, a regularization term $\|\mathbf{w}\|_2^2$ about the parameter vector $\mathbf{w}$ is added to the objective function:

$$\min_{\mathbf{w}, \mathbf{y}} \frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{c}{2} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2,$$
$$s.t.\ \mathbf{y} \in \{0, 1\}^{|\mathcal{L}| \times 1}, \text{ and } y_l = 1, \forall l \in \mathcal{E},$$

where constant c denotes the weight of the loss term in the function.

3.3 Link Cardinality Constraints

The *cardinality constraints* define both the limit on link cardinality and the limit on node degrees that those links are incident to. To be general, the links studied in this paper can be either uni-directed or bi-directed, where undirected links are treated as bi-directed. For each node $u \in \mathcal{V}$ in the network, we can represent the potential links going-out from u as set $\Gamma^{out}(u) = \{l | l \in \mathcal{L}, \exists v \in \mathcal{V}, l = (u, v)\}$, and those going-into u as set $\Gamma^{in}(u) = \{l | l \in \mathcal{L}, \exists v \in \mathcal{V}, l = (v, u)\}$. Furthermore, with the link label variables $\{y_l\}_{l \in \mathcal{L}}$, we can represent the out-degree and in-degree of node $u \in \mathcal{V}$ as $degree^{out}(u) = \sum_{l \in \Gamma^{out}(u)} y_l$ and $degree^{in}(u) = \sum_{l \in \Gamma^{in}(u)} y_l$ respectively. Considering that the node degrees cannot be negative, besides the upper bounds introduced by the *cardinality constraints*, a lower bound "≥ 0" is also added to guarantee validity of node degrees by default.

One-to-One Cardinality Constraint

For the bi-directed anchor links with $1:1$ *cardinality constraint*, the nodes in the information networks can be attached with at most one such kind of link. In other words, for all the nodes (e.g., $u \in \mathcal{V}$) in the network, its in-degree and out-degree can not exceed 1, i.e.,

$$0 \leq \sum_{l \in \Gamma^{out}(u)} y_l \leq 1, \forall u \in \mathcal{V}, \text{ and } 0 \leq \sum_{l \in \Gamma^{in}(u)} y_l \leq 1, \forall u \in \mathcal{V}.$$

Algorithm 1 Greedy Link Selection (GREEDYLS)

Input: link estimate result $\hat{\mathbf{y}}$, parameter k
Output: link label vector $\mathbf{y}$

```
initialize link label vector y = 0
for l ∈ E do
    y_l = 1
end for
for l ∈ L \ E and ŷ_l < 0.5 do
    y_l = 0
end for
Let L̃ = {l | l ∈ L \ E, ŷ_l ≥ 0.5}
while L̃ ≠ ∅ do
    select l ∈ L̃ with the highest estimation score
    if add l as positive instance violates the cardinality constraint or more than k links have been selected then
        y_l = 0
    else
        y_l = 1
    end if
end while
return y
```

One-to-Many Cardinality Constraint

Meanwhile, for the uni-directed supervision links with the $N:1$ *cardinality constraint*, the manager nodes can have multiple (N) links going out from them while the subordinate nodes should have exactly one link going into them (except the CEO). In other words, for all the nodes (e.g., $u \in \mathcal{V}$) in the network, its *out-degree* cannot exceed N and the *in-degree* should be exactly 1, i.e.,

$$0 \leq \sum_{l \in \Gamma^{out}(u)} y_l \leq N, \forall u \in \mathcal{V}, \text{ and } 1 \leq \sum_{l \in \Gamma^{in}(u)} y_l \leq 1, \forall u \in \mathcal{V}.$$

Many-to-Many Cardinality Constraint

In many cases, there usually exist no specific *cardinality constraints* on links, and nodes can be connected with each other freely. Simply, we can assume the node *in-degrees* and *out-degrees* to be limited by the maximum degree parameter $N = |\mathcal{V}| - 1$, i.e.,

$$0 \leq \sum_{l \in \Gamma^{out}(u)} y_l \leq N, \forall u \in \mathcal{V}, \text{ and } 0 \leq \sum_{l \in \Gamma^{in}(u)} y_l \leq N, \forall u \in \mathcal{V}.$$

General Cardinality Constraint Representation

The *cardinality constraint* on links can be generally represented with the linear algebra equations. The relationship between nodes $\mathcal{V}$ and links $\mathcal{L}$ can actually be represented as matrices $\mathbf{T}^{out} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{L}|}$ and $\mathbf{T}^{in} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{L}|}$, where entry $\mathbf{T}^{out}(u,l) = 1$ iff $l \in \Gamma^{out}(u)$ and $\mathbf{T}^{in}(u,l) = 1$ iff $l \in \Gamma^{in}(u)$. Based on the link label vector $\mathbf{y}$, we can formally represent the node out-degrees and in-degrees as vectors $\mathbf{T}^{out}\cdot\mathbf{y}$ and $\mathbf{T}^{in}\cdot\mathbf{y}$ respectively. The general representation of the *cardinality constraints* introduced above can be rewritten as follows:

$$\underline{\mathbf{b}}^{out} \leq \mathbf{T}^{out}\cdot\mathbf{y} \leq \overline{\mathbf{b}}^{out}, \text{ and } \underline{\mathbf{b}}^{in} \leq \mathbf{T}^{in}\cdot\mathbf{y} \leq \overline{\mathbf{b}}^{in},$$

where vectors $\underline{\mathbf{b}}^{out}$, $\overline{\mathbf{b}}^{out}$, $\underline{\mathbf{b}}^{in}$ and $\overline{\mathbf{b}}^{in}$ can take different values depending on the cardinality constraint on the links (e.g., for the $1:1$ constraint, we have $\underline{\mathbf{b}}^{out} = \underline{\mathbf{b}}^{in} = \mathbf{0}$ and $\overline{\mathbf{b}}^{out} = \overline{\mathbf{b}}^{in} = \mathbf{1}$).

3.4 Joint Constrained Link Prediction Optimization Function

Based on the above remarks, the constrained optimization objective function of the CLP problem can be represented as

$$\min_{\mathbf{w},\mathbf{y}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{c}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2,$$
$$s.t.\ \mathbf{y} \in \{0,1\}^{|\mathcal{L}|\times 1}, y_l = 1, \forall l \in \mathcal{E},$$
$$\underline{\mathbf{b}}^{out} \leq \mathbf{T}^{out}\cdot\mathbf{y} \leq \overline{\mathbf{b}}^{out}, \underline{\mathbf{b}}^{in} \leq \mathbf{T}^{in}\cdot\mathbf{y} \leq \overline{\mathbf{b}}^{in}.$$

The above objective function involves variables $\mathbf{w}$ and $\mathbf{y}$ at the same time, which is actually not jointly convex and can be very challenging to solve. In this paper, we propose to solve the function with an alternative updating framework ITERCLIPS by fixing one variable and updating the other one iteratively. The framework ITERCLIPS involves two steps:

Step 1: Fix $\mathbf{y}$ and Update $\mathbf{w}$

By fixing $\mathbf{y}$ (i.e., treating $\mathbf{y}$ as a constant vector), the objective function about $\mathbf{w}$ can be simplified as

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{c}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2.$$

Let $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{c}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$. By taking the derivative of the function $h(\mathbf{w})$ regarding $\mathbf{w}$ we can have

$$\frac{\mathrm{d}h(\mathbf{w})}{\mathrm{d}\mathbf{w}} = \mathbf{w} + c\mathbf{X}\mathbf{w}\mathbf{X}^\top - c\mathbf{y}\mathbf{X}^\top.$$

By making the derivation to be zero, we can obtain the optimal vector $\mathbf{w}$ to be

$$\mathbf{w} = c(\mathbf{I} + c\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y},$$

and the minimum value of the function will be $\frac{c}{2}\mathbf{y}^\top\mathbf{y} - \frac{c^2}{2}\mathbf{y}^\top\mathbf{X}(\mathbf{I} + c\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$.

Step 2: Fix $\mathbf{w}$ and Update $\mathbf{y}$

When fixing $\mathbf{w}$ and treating it as a constant vector, the objective function about $\mathbf{y}$ can be represented as

$$\min_{\mathbf{y}} \frac{c}{2}\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2,$$
$$s.t.\ \mathbf{y} \in \{0,1\}^{|\mathcal{L}|\times 1}, y_l = 1, \forall l \in \mathcal{E},$$
$$\underline{\mathbf{b}}^{out} \leq \mathbf{T}^{out}\cdot\mathbf{y} \leq \overline{\mathbf{b}}^{out}, \underline{\mathbf{b}}^{in} \leq \mathbf{T}^{in}\cdot\mathbf{y} \leq \overline{\mathbf{b}}^{in},$$

where $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$ denotes the inference results of the links in $\mathcal{L}$ with the updated parameter vector $\mathbf{w}$ from Step 1. The objective function is an constrained non-linear integer programming problem about variable $\mathbf{y}$. Formally, the above optimization sub-problem is named as the CLS (Cardinality Constrained Link Selection) problem. The CLS problem is shown to be NP-hard (we will analyze it in the next subsection), and achieving the optimal solution to it is very time consuming. To preserve the *cardinality constraints* on the variables and minimize the loss term, one brute-force way to achieve the optimal solution $\mathbf{y}$ is to enumerate all the feasible combination of links candidates to be selected as the positive instances, which will lead to very high time complexity. In this paper, we propose a greedy link selection algorithm to resolve the problem, and the pseudo-code of the greedy link selection method GREEDYLS is available in Algorithm 1. Meanwhile, the framework ITERCLIPS is illustrated with the pseudo-code available in Algorithm 2. ITERCLIPS updates vectors $\mathbf{w}$ and $\mathbf{y}$ alternatively until both of them converge.

Detailed analysis about the CLS problem and the greedy algorithm GREEDYLS will be provided in the next subsection.

3.5 Problem and Algorithm Analysis

In this part, we will show the CLS problem with $M:N$ *cardinality constraints* can be reduced to the *k-maximum weight matching problem*, which is NP-hard and not solvable in polynomial time.

Algorithm 2 Framework ITERCLIPS

Input: link feature vector $\mathbf{X}$
weight parameter c
Output: parameter vector $\mathbf{w}$, link label vector $\mathbf{y}$

Initialize label vector $\mathbf{y} = \frac{1}{2} \cdot \mathbf{1}$
For links in $\mathcal{E}$, assign their label as 1
Initialize parameter vector $\mathbf{w} = \mathbf{0}$
Initialize convergence-tag = False
while convergence-tag == False **do**
  Update vector $\mathbf{w}$ with equation $\mathbf{w} = c(\mathbf{I} + c\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$
  Calculate link estimation result $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$
  Update vector $\mathbf{y}$ with Algorithm GREEDYLS($\hat{\mathbf{y}}$)
  if $\mathbf{w}$ and $\mathbf{y}$ both congerge **then**
    convergence-tag = True
  end if
end while

In addition, we will also prove that the GREEDYLS method can actually achieve $\frac{1}{2}$-approximation of the optimal result of the **C**LS problem.

In the **C**LS problem, for all the existing links in $\mathcal{E}$, we know their label should be 1 in advance. For all the links in $\mathcal{L} \setminus \mathcal{E}$ with estimation score (i.e., $\hat{y}$) lower than 0.5, assigning their label with value 0 will introduce less loss and has no impact on the cardinality constraints. Therefore, in Algorithm 1, these links are handled in advance to simplify the problem. For the remaining links, we need to select those with high scores to assign with label 1 (so as to minimize the loss term), and preserve the *cardinality constraints* at the same time. For the links selection of which violates the cardinality constraints, they will be assigned with label 0 instead.

Formally, we can represent the unlabeled links with confidence scores greater than 0.5 as set $\tilde{\mathcal{L}} = \{l | l \in \mathcal{L} \setminus \mathcal{E}, \hat{y}_l > 0.5\}$. For all the links in set $\tilde{\mathcal{L}}$, we can represent the introduced loss term as

$$\sum_{l \in \tilde{\mathcal{L}}} (\hat{y}_l - y_l)^2 = \sum_{l \in \tilde{\mathcal{L}}} \hat{y}_l^2 + \sum_{l \in \tilde{\mathcal{L}}} y_l^2 - \sum_{l \in \tilde{\mathcal{L}}} 2\hat{y}_l \cdot y_l,$$

where term $\sum_{l \in \tilde{\mathcal{L}}} \hat{y}_l^2$ is a constant, term $\sum_{l \in \tilde{\mathcal{L}}} y_l^2$ denotes the number of selected links, and $\sum_{l \in \tilde{\mathcal{L}}} 2\hat{y}_l \cdot y_l$ represents the confidence scores of the selected links. Let's assume k links are selected finally, i.e., $\sum_{l \in \tilde{\mathcal{L}}} y_l^2 = k$, the optimal k links which can minimize the loss term can be achieved by maximizing the confidence scores of the selected links:

$$\max \sum_{l \in \tilde{\mathcal{L}}} \hat{y}_l y_l$$
$$s.t.\ \ y_l \in \{0,1\}, \forall l \in \tilde{\mathcal{L}}, \sum_{l \in \tilde{\mathcal{L}}} y_l = k,$$
$$\underline{\mathbf{b}}^{out} \leq \mathbf{T}^{out} \cdot \mathbf{y} \leq \overline{\mathbf{b}}^{out}, \underline{\mathbf{b}}^{in} \leq \mathbf{T}^{in} \cdot \mathbf{y} \leq \overline{\mathbf{b}}^{in}.$$

By enumerating different k values in range $[1, |\tilde{\mathcal{L}}|]$, we can identify the optimal link set for the **C**LS problem.

Theorem 1: The k-maximum weighted matching problem can be reduced to the above optimization problem with general $M : N$ *cardinality constraints*.

Proof: The above optimization problem with $1 : 1$ *cardinality constraint* is actually identical to the *k-maximum weighted matching problem* studied in the existing works [5], and the reduction is trivial. Meanwhile, for the above optimization problem with $N : 1$ *cardinality constraints* on the links, we can have vectors $\overline{\mathbf{b}}^{in} = \underline{\mathbf{b}}^{in} = [1, 1, \cdots, 1]^\top$, $\overline{\mathbf{b}}^{out} = [N, N, \cdots, N]^\top$, and $\underline{\mathbf{b}}^{out} = \mathbf{0}$. Given the information network G with $1 : N$ *cardinality constraints*, we propose to construct N dummy nodes for each the nodes with out-going links. The constructed dummy nodes are connected to the original nodes to indicate the belonging relationships. For each original link, e.g., (u, v), we will add a dummy directed link connecting the dummy node created for u with node v, whose weight is identical to the weight of the original link (u, v). Given the *k-maximum weighted matching* result on the constructed dummy network, we can obtain the optimal solution to the above optimization problem on network G by replacing all the created dummy nodes with the original nodes corresponding to them. Meanwhile, for any solution to the above optimization problem on network G, we can also obtain the solution to the *k-maximum weighted matching* problem on the constructed dummy network. In other words, the *k-maximum weighted matching problem* can be reduced to the above optimization problem with $N : 1$ *cardinality constraints* via the constructed dummy network. Meanwhile, for the networks with general $M : N$ constraint, dummy nodes can be created for both nodes with out-going links and in-coming links at the same time, whose reduction to the *k-maximum weighted matching problem* is not provided due to the limited space. In addition, in the $M : N$ case, to avoid the case that solutions pick links connecting the more than one link connecting the dummy nodes corresponding the common node pairs (e.g., both (u', v') and (u'', v'') are selected, where u', u'' and v', v'' are the dummy nodes of u and v respectively), more constraints will be added to the objective function of the *k-maximum weighted matching problem*.

According to the existing works [6], the *k-maximum weighted matching problem* is actually NP-hard. To address the problem efficiently, a greedy link selection method called GREEDYLS is applied as introduced in the previous subsection. As shown in Algorithm 1, among all the remaining links in $\tilde{\mathcal{L}}$, GREEDYLS picks the links with the highest confidence scores $\hat{y}_l$. If the selection of a link doesn't violate the *cardinality constraint*, GREEDYLS will add it to

Algorithm 3 Distributed GREEDYLS (DISTGREEDYLS) with $1 : 1$ Cardinality Constraint

Input: node u, neighbor set $\Gamma(u)$

Initialize neighborhood set $N = \Gamma(u)$
Initialize matching candidate set $C = \emptyset$
Select candidate $c = candidate(u, N)$
if $c \neq null$ **then**
  Send $\langle invite \rangle$ message to c
end if
while $N \neq \emptyset$ **do**
  Receive a message m from neighbor v
  if $m == \langle invite \rangle$ **then**
    $C = C \cup \{v\}$
  end if
  if $m == \langle remove \rangle$ **then**
    $N = N \setminus \{v\}$
    $C = C \setminus \{v\}$
    if $v == c$ **then**
      Select new candidate $c = candidate(u, N)$
      if $c \neq null$ **then**
        Send $\langle invite \rangle$ message to c
      end if
    end if
  end if
  if $c \neq null \wedge c \in C$ **then**
    for $w \in C \setminus \{c\}$ **do**
      Send $\langle remove \rangle$ message to w
    end for
    $C = \emptyset$
  end if
end while

the final result. We will show that GREEDYLS can actually achieve $\frac{1}{2}$-approximation of the optimal result.

Theorem 2: The GREEDYLS method can achieve $\frac{1}{2}$-approximation of the optimal solution to the **C**LS problem.

Proof: Formally, let $\mathcal{C}$ be the set of links selected by GREEDYLS to assign with label $+1$, while the optimal solution to the **C**LS problem can be represented as OPT. Every time, when GREEDYLS selects the links with the highest confidence score (e.g., $l = (u, v)$) to add to $\mathcal{C}$, the degrees of nodes u and v will get increased by 1 and some other links incident to u, v will no longer get added to $\mathcal{C}$ due to the degree limit (introduced by the *cardinality constraint*). At most two links incident to u and v can get removed due to the selection of (u, v), since (u, v) occupies the degree space of u and v by one respectively. Formally, we can represent the set of links incident to l as set $\Gamma(l) = \Gamma^{out}(u) \cup \Gamma^{in}(v)$. Depending on whether link $l \in \mathcal{C}$ is in OPT or not and the number of links in the optimal solution but are removed in $\mathcal{C}$ due to the selection of l (i.e., links in $\Gamma(l) \cap (OPT \setminus \mathcal{C})$), there exist 3 cases:

1. $l \in OPT$: Link l also belongs to the optimal result, and adding l into $\mathcal{C}$ will not affect the selection of other links.
2. $l \notin OPT$ and $\Gamma(l) \cap (OPT \setminus \mathcal{C}) = \{l_1\}$: Link l is not in the optimal solution, and adding l to the result $\mathcal{C}$ will occupy the degree space and make the optimal link $l_1 \in OPT$ (incident to either u or v) fail to be selected. Meanwhile, since l is the link with the highest score at selection, if l_1 is not selected ahead of l, it is easy to show that $\hat{y}_l > \hat{y}_{l_1} > \frac{1}{2}\hat{y}_{l_1}$.
3. $l \notin OPT$ and $\Gamma(l) \cap (OPT \setminus \mathcal{C}) = \{l_1, l_2\}$: Link l is not in the optimal solution, and adding of $l = (u, v)$ will occupy the degrees of nodes u and v by 1 and make links $l_1, l_2 \in OPT$ incident to u and v respectively to be removed. Since l has the highest score, if links l_1 and l_2 are not selected ahead of l, it is easy to show that $\hat{y}_l > \hat{y}_{l_1}$ and $\hat{y}_l > \hat{y}_{l_2}$. Therefore, we have $\hat{y}_l > \frac{1}{2}(\hat{y}_{l_1} + \hat{y}_{l_2})$.

Based on the above remarks, for all the selected links in $\mathcal{C}$, we have

$$\begin{aligned}\hat{y}(\mathcal{C}) &= \hat{y}\left((\mathcal{C} \cap OPT) \cup (\mathcal{C} \setminus OPT)\right)\\ &= \hat{y}(\mathcal{C} \cap OPT) + \hat{y}(\mathcal{C} \setminus OPT)\\ &= \hat{y}(\mathcal{C} \cap OPT) + \sum_{l \in \mathcal{C} \setminus OPT} \hat{y}_l\\ &> \frac{1}{2}\hat{y}(OPT \cap \mathcal{C}) + \frac{1}{2}\sum_{l \in OPT \setminus \mathcal{C}} \hat{y}_l\\ &= \frac{1}{2}\hat{y}(OPT).\end{aligned}$$

where $\hat{y}(\mathcal{C}) = \sum_{l \in \mathcal{C}} \hat{y}_l$ denotes the score sum of the links in $\mathcal{C}$.

Therefore, the GREEDYLS algorithm can achieve $\frac{1}{2}$-approximation of the optimal solution for the **C**LS problem with $M : N$ link *cardinality constraint*, and the time complexity of the GREEDYLS method is $O(|\tilde{\mathcal{L}}|)$.

3.6 Distributed Framework

Meanwhile, for large-scale networks involving billions of nodes and links, the complete network data can hardly be stored in one single machine and framework ITERCLIPS may suffer from the high computational cost problem a lot. In this section, we will introduce a scalable version of framework ITERCLIPS based on distributed computational platforms. The framework ITERCLIPS involves two iterative steps actually. In the first step, we need to update vector $\mathbf{w}$ to calculate the confidence vector $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} = c\mathbf{X}(\mathbf{I} + c\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$, where matrix $c\mathbf{X}(\mathbf{I} + c\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ can actually be pre-computed and divided into blocks to be stored in different slaves (i.e., worker nodes in a cluster). For instance, in Spark, the matrix can be divided into rows, where each row can be saved as a RDD (resilient distributed dataset) in one slave, and each entry in vector $\hat{\mathbf{y}}$ can be updated independently in different slaves simultaneously. The updated values in $\hat{\mathbf{y}}$ can be exchange among the slaves with very low communication costs. Meanwhile, for the second step in framework ITERCLIPS, how to generalize GREEDYLS to the distributed version is not very straightforward, which will be the focus in the following part of this subsection.

According to Theorem 1, the the *k-maximum weighted matching problem* can be reduced to the objective function of k-**C**LS with general $M : N$ cardinality constraint in polynomial time. Therefore, next we will propose the distributed version of GREEDYLS for the **C**LS with $1 : 1$ constraint specifically (which can be applied for the general $M : N$ cardinality constraint as well). Before diving into details of the DISTGREEDYLS (Distributed GREEDYLS) algorithm, we first provide some intuitive ideas about how DISTGREEDYLS works. In the distributed weighted link selection, each process representing one node in the graph knows its neighbor nodes as well as their inferred confidence scores $\hat{y}$. These processes can also communicate with each other by sending and receiving messages. Via the communication among processes, links with locally highest confidence scores can be identified concurrently. Intuitively, by running the algorithm on all the nodes simultaneously, the same matching result can be obtained as Algorithm GREEDYLS based on the stand-alone mode.

Formally, the pseudo-code of the distributed algorithm DISTGREEDYLS is available in Algorithm 3. According to the algorithm, for each node u, we can initialize its neighbor set N as $\Gamma(u)$ (N will change dynamically in the algorithm). Function $c = candidate(u, N)$ returns the candidate of u, whose link with u is of the highest confidence score, i.e.,

$$c = candidate(u, N) = \arg_{v \in N} \max \hat{y}_{(u,v)}.$$

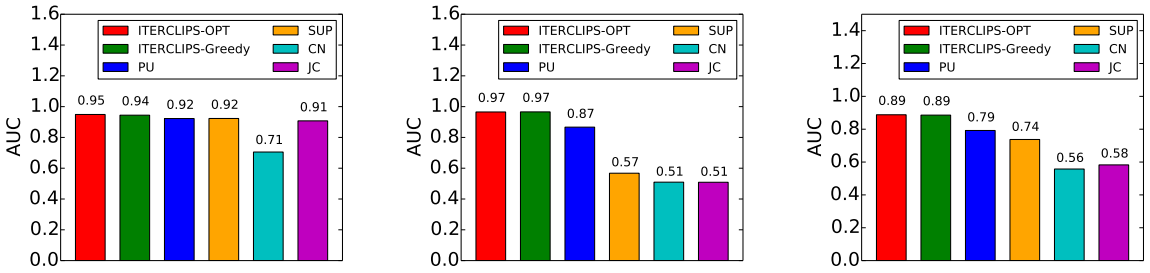
Initially, node u will send the *invite* message to the candidate c, and receive messages from all the neighbors in set N. If the message received from neighbor v is also an *invitation* (i.e., u is the most promising candidate of v), u will add v to its *matching candidate set* C. Meanwhile, if the message is *remove*, it denotes v has already found its partner and link between them has already been selected. Node v will be removed from u's *neighbor set* and *matching candidate set*. What's more, if v happens to be candidate c, node u will retrieve the next most promising candidate c and send the new *invite* message again. Finally, if candidate c invites u and u also invites c, the link between whom will be of the highest score and selected finally.

Lemma 1: In the DISTGREEDYLS algorithm, each process (node) sends out at most one message over each incident edge.

Proof: In the algorithm, for each node u, u sends *invite* message to the first candidate as well as other candidate if the previous candidates send a *remove* message to u. Therefore, for each potential candidate c obtained via function $candidate(u, N)$, u sends at most one *invite* message to c. Meanwhile, the *remove* messages are merely sent to the other neighbors in N (excluding candidates c) only once. In other words, all the neighbors in $\Gamma(u)$ only receive exactly one message (either *invite* or *remove*) from u in the whole process.

According to the analysis in Lemma 1, we can prove the time complexity of the DISTGREEDYLS Algorithm to be $O(|\tilde{\mathcal{L}}|)$.

4. EXPERIMENTS



(a) Collaboration Network (b) Organizational Chart (c) Foursquare-Twitter

Figure 1: Performance of Comparison Methods Evaluated by AUC.

To demonstrate the effectiveness of framework ITERCLIPS, extensive experiments will be done on real-world information network datasets in this section. In the following parts, we will first introduce the datasets used in the experiments. After that, we will talk about the experimental setting in detail, and show the experimental results with detailed performance analysis.

4.1 Dataset Description

Three different information network datasets are used in the experiments, links in which are subject to different *cardinality constraints*. The first dataset used in this paper is the collaboration network crawled from Arxiv[1], which involves about $18,772$ researchers and $198,110$ co-author links among the researchers. Links in the collaboration network are subject to the *many-to-many cardinality constraint*. The second dataset used in this paper is about the organizational chart [26, 20, 27] of a famous IT company[2], which contains more than $100k$ employees and the supervision links among the employees as well as employee internal profile information in the company. The supervision links in the organizational chart are subject to the *one-to-many cardinality constraint*. The third dataset used is about multiple aligned social networks, Foursquare and Twitter [28, 11, 23], from which about $5k$ users together with their complete social information are crawled. The anchor links across the networks are subject to the *one-to-one cardinality constraint*.

4.2 Experimental Setting

4.2.1 Experimental Setup

In the experiments, from each dataset, we can obtain the set of existing known links as well as the non-existing links, which are treated as the positive and negative links respectively. Based on the known positive links, we propose to apply the 5-fold cross validation to partition the links into two subsets according to ratio $4:1$, where 4 folds are used as the training set and 1 fold is used as the testing set. The testing set will be combined with the negative link set to form the final unlabeled link set. Based on the training and unlabeled link sets, we aim at building models to uncover these hidden positive links from the unlabeled link set. When building the framework ITERCLIPS, a set of features are extracted for the links based on the information available in the information networks. In the experiment, to be general, the features are extracted based on a set of the meta paths. For more information about the meta path based features defined for the co-author links, supervision links, and anchor links please refer to [18, 26, 21] respectively. Based on the ITERCLIPS framework introduced in this paper, a set of promising links will be selected from the unlabeled link set, which will be labeled as the positive instances. Meanwhile, the estimation scores of these links achieved at the convergence will also be outputted by ITERCLIPS as the confidence scores of these links. The predicted results together with the ground truth will be measures by some traditional metrics to evaluate the performance of different link prediction methods. All the codes are implemented in Python based on Spark 1.5.2. The experiments are done based on a cluster of 10 servers (each with 24 CPUs (based on the x86_64 architecture) and 94 GB memory), and the operating system is Red Hat Enterprise Linux Server release 6.5.

[1]https://snap.stanford.edu/data/ca-AstroPh.html

[2]we are not able to reveal the company name and actual statistical numbers here and throughout the paper

4.2.2 Comparison Methods

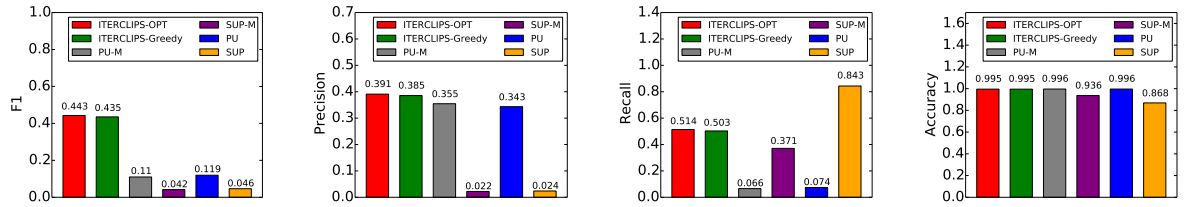
The set of methods to be compared in the experiments can be divided into 2 categories depending on whether the *cardinality constraint* is considered or not in building the model.

Comparison Methods with Cardinality Constraints

- ITERCLIPS: Framework ITERCLIPS is the cardinality constrained link prediction framework proposed in this paper, which applies the greedy method GREEDYLS to select promising links in step 2.
- ITERCLIPS-OPT: Framework ITERCLIPS-OPT is identical to ITERCLIPS except that the Hungarian algorithm is applied in step 2 of ITERCLIPS-OPT to select the optimal links from the network.
- PU-M: Method PU-M is an extension to the PU (positive and unlabeled) link prediction method proposed in [28]. In the experiment, we extend the PU link prediction with an additional network flow based matching step [26] to prune the redundant links and preserve the link cardinality constraint.
- SUP-M: Method SUP-M is a two-phase supervised classification based link prediction method involving a supervised link prediction step [7] (treating all unlabeled links as negative instances) and a network flow based link post-processing step to maintain the link cardinality constraint.

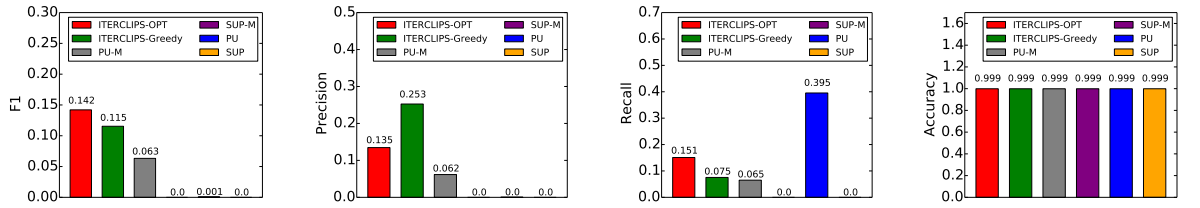
Comparison Methods without Cardinality Constraints

- PU: Method PU is the PU link prediction proposed in [28]. PU is also the first step of method PU-M and it doesn't consider the cardinality constraints



(a) Collaboration-F1 (b) Collaboration-Precision (c) Collaboration-Recall (d) Collaboration-Accuracy

Figure 2: Performance of Comparison Methods on Collaboration Network Evaluated by F1, Precision, Recall and Accuracy.



(a) Chart-F1 (b) Chart-Precision (c) Chart-Recall (d) Chart-Accuracy

Figure 3: Performance of Comparison Methods on Organizational Chart Evaluated by F1, Precision, Recall and Accuracy.

- SUP: Method SUP is the supervised classification based link prediction method proposed in [7]. It is also the first step of method SUP-M and it doesn't consider the cardinality constraints on links.
- *Unsupervised Link Prediction*: To be complete, we also compare the proposed framework ITERCLIPS with a set of traditional unsupervised link prediction methods, including CN (*Common Neighbor*) [13] and JC (*Jaccard's Coefficient*) [13].

4.2.3 *Evaluation Metrics*

Depending on the output of these link prediction methods, various conventional evaluation metrics are applied to measure their performance. For the methods which can output link confidence scores, including ITERCLIPS, ITERCLIPS-OPT, PU, SUP, CN and JC, we will apply the AUC as their evaluation metric. For methods which can output link prediction labels, including ITERCLIPS, ITERCLIPS-OPT, PU-M, SUP-M, PU and SUP, we will use F1, Recall, Precision and Accuracy as the evaluation metrics.

4.3 Experimental Results with Analysis

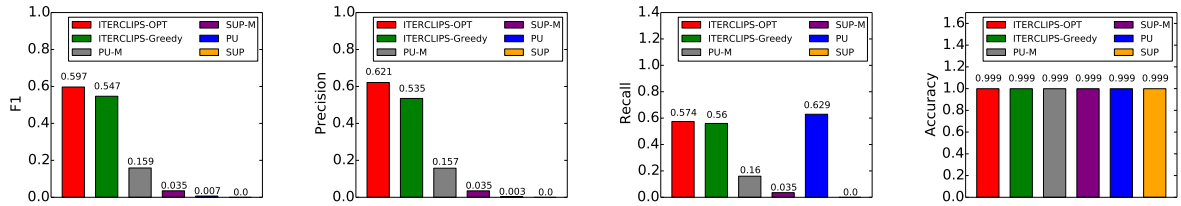
The experimental results achieved by different comparison methods on these three datasets evaluated by AUC, F1, Precision, Recall and F1 are shown in Figures 1-4. From Figure 1, we can observe that ITERCLIPS-OPT and ITERCLIPS can achieve larger AUC scores on these three different network datasets than the other methods. For instance, on the organizational chart dataset, the AUC scores achieved by ITERCLIPS-OPT and ITERCLIPS are both 0.97, which is almost the double of the score obtained by SUP, CN and JC. It demonstrates the claim we make when proposing the **C**LP problem at the very beginning: incorporating the cardinality constraints in the link prediction problem modeling can greatly improve the prediction result. Meanwhile, by comparing the advantages of ITERCLIPS-OPT and ITERCLIPS over the other methods, we observe that the advantage is more significant when inferring the supervision links and anchor links while the advantage is relatively slight when inferring the co-author links. The results shows that the ITERCLIPS framework works very well in inferring different types of links, and its is most suitable for predicting links with challenging *one-to-many* and *one-to-one* cardinality constraints.

In Figures 2-4, we show the performance of the comparison methods on these three different types of network datasets respectively. According to the results, we can observe that (1) for metrics F1 and Precision, frameworks ITERCLIPS and ITERCLIPS-OPT can achieve much better performance, and the scores are almost three times greater than other methods according to the plots 2(a), 3(a), 3(b), 4(a) and 4(b); (2) for some metrics (like recall) method PU and SUP can perform slightly better; and (3) all these methods can achieve very high accuracy scores. The potential explanation for these observations can be (1) framework ITERCLIPS and ITERCLIPS-OPT can identify a reasonable sized link set from the networks, and many of the identified links are correct (i.e., high precision and f1); (2) PU and SUP predicts a large number of links to be positive, which can cover the majority of positive links (i.e., achieve high recall); (3) in the class imbalance case (i.e., non-existing links are far more than the existing links), predicting all the links to be negative can still achieve very high accuracy score, which is not a good metric in such a case actually.

By comparing ITERCLIPS with ITERCLIPS-OPT, their performance is comparable and in some cases the greedy link selection algorithm proposed in this paper can perform slightly better (e.g., in plot 3(b)). However, as shown in Table 1, the time cost of ITERCLIPS-OPT is much larger than the cost of ITERCLIPS, and the difference will be more obvious when the distributed version DISTGREEDYLS is applied. It supports the motivation of applying the greedy link selection to solve the **C**LS problem. In addition, by comparing the scores achieved by all the comparison methods on these three different types of networks, the advantages of ITERCLIPS over the other methods is more prominent when inferring anchor links with *one-to-one* cardinality constraint between Foursquare and Twitter, which indicate the ITERCLIPS can show its potential in more challenging cardinality constraints.

5. RELATED WORK

Link prediction problems is a traditional research problem stud-



(a) FsqTwt-F1 (b) FsqTwt-Precision (c) FsqTwt-Recall (d) FsqTwt-Accuracy

Figure 4: Performance of Comparison Methods on Foursquare-Twitter Networks Evaluated by F1, Precision, Recall and Accuracy.

Table 1: Running Time of Comparison Methods on Three Different Datasets (time is in minutes).

Methods	Collaboration		Chart		Fsq-Twt	
	update	select	update	select	update	select
ITERCLIPS-OPT	5.76	49.82	46.84	868.81	8.92	62.6
ITERCLIPS	5.75	23.31	45.75	93.76	8.73	24.66
ITERCLIPS-DIST	0.53	2.51	2.13	4.47	0.71	3.14

ied in various areas, which aims at inferring the connections among nodes in the graph. To this context so far, dozens of link prediction works have been published already [14, 4, 15, 19, 2]. Depending on the learning setting utilized, the existing link prediction models for information networks can be divided int several categories. Initially, researchers study the link prediction problem based on an unsupervised learning setting [13], which predicts links by calculating the similarity scores among nodes with the assumption that close nodes are more likely to be connected. Afterwards, to utilize the supervision information and incorporate multiple closeness measures altogether, researchers introduce the supervised classification based link prediction models [7], where the existing and non-existing links are labeled as the positive an negative instances respectively. Recently, researchers point that labeling the non-existing as negative instances is not reasonable, since some of the links will be formed, which should be unlabeled actually [28, 23]. Based on such an intuition, link prediction framework based on PU (Positive and Unlabeled) learning setting is introduced in [28, 23]. To gain a more comprehensive knowledge about existing link prediction works, please refer to the survey paper [22, 16, 8] for more information.

Cardinality constraints are a common concept used in link (relation or projection) modeling, which effectively specifies the population and physical meaning of links (relations and projections). In traditional algebra and set theory studies, researchers use cardinality constraints to represent the property of link (relation or projection), which can be *injective* (i.e., one-to-one), *surjective* and *bijective*. In conventional database studies, the practitioners adopt cardinality constraints to denote the property of relations among entities in the ER (Entity Relation) model [3], which include *one-to-one*, *one-to-many* and *many-to-many*. Meanwhile, in theoretic computer science area, a bunch of traditional research problems can be modeled as optimization problem subject to certain cardinality constraints, like the *stable marriage* problem [10], *assignment problem* [12], *maximum graph matching* problem [5], etc. As shown in this paper, the link selection problem with cardinality constraint can be reduced to the *maximum graph matching* problem [5] actually, which is a traditional research problem studied in graph theory and combinatorial mathematics. After the problem was proposed, several variants of the problem have been introduced, including the *maximum bipartite graph matching* [9], *maximum weighted graph matching* [1], *minimum maximal matching* [17], etc.

6. CONCLUSION

In this paper, we have studied the **C**LP problem, whose objective is to predict the links in information networks subject to the *cardinality constraints*. A new link prediction framework ITERCLIPS is proposed to resolve the problem, which involves two alternative steps: (1) weight vector updating, and (2) cardinality constrained link selection. To overcome the high-time complexity problem encounter when tackling the link selection step, a new greedy link selection algorithm has been proposed in this paper. In addition, to ensure the effectiveness of ITERCLIPS on large-scale networks, a distributed implementation of ITERCLIPS has been further presented as a scalable solution to the **C**LP problem. Extensive experiments have been done on three different information network datasets in this paper, links in which are subject to different cardinality constraints. The experimental results have demonstrated our claim that: by incorporating the cardinality constraints in the link prediction problem modeling, the link prediction results can be improved significantly.

7. ACKNOWLEDGEMENT

This work is supported in part by NSF through grants IIS-1526499, and CNS-1626432. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

8. REFERENCES

[1] D. Avis. Np-completeness of some generalizations of the maximum matching problem. *Networks*, 1983.

[2] B. Cao, N. Liu, and Q. Yang. Transfer learning for collective link prediction in multiple heterogenous domains. In *ICML*, 2010.

[3] P. Chen. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1976.

[4] Y. Dong, J. Tang, S. Wu, J. Tian, N. Chawla, J. Rao, and H. Cao. Link prediction and recommendation across heterogeneous social networks. In *ICDM*, 2012.

[5] J. Edmonds. Maximum matching and a polyhedron with 0, 1 vertices. *Journal of Research the National Bureau of Standards*, 1965.

[6] M. Garey and D. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[7] M. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM*, 2006.

[8] M. Hasan and M. J. Zaki. A survey of link prediction in social networks. In Charu C. Aggarwal, editor, *Social Network Data Analytics*. 2011.

[9] J. Hopcroft and R. Karp. A n5/2 algorithm for maximum matchings in bipartite. In *SWAT*, 1971.

[10] R. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 1994.

[11] X. Kong, J. Zhang, and P. Yu. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*, 2013.

[12] H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*.

[13] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, 2003.

[14] Z. Lu, B. Savas, W. Tang, and I. Dhillon. Supervised link prediction using multiple sources. In *ICDM*, 2010.

[15] G. Qi, C. Aggarwal, and T. Huang. Link prediction across networks by biased cross-network sampling. In *ICDE*, 2013.

[16] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. Yu. A survey of heterogeneous information network analysis. *CoRR*, abs/1511.04854, 2015.

[17] L. Stockmeyer and V. Vazirani. a survey of heuristics for weighted matching problem. *Information Processing Letters*, 1982.

[18] Y. Sun, R. Barber, M. Gupta, C. Aggarwal, and J. Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM*, pages 121–128, 2011.

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

[20] J. Zhang, Y. Lv, and P. Yu. Enterprise social link recommendation. In *CIKM*, 2015.

[21] J. Zhang, W. Shao, S. Wang, X. Kong, and P. S. Yu. Pna: Partial network alignment with generic stable matching. In *IEEE IRI*, 2015.

[22] J. Zhang and P. Yu. Link prediction across heterogeneous social networks: A survey. 2014.

[23] J. Zhang and P. Yu. Integrated anchor and social link predictions across partially aligned social networks. In *IJCAI*, 2015.

[24] J. Zhang and P. Yu. Multiple anonymized social networks alignment. In *ICDM*, 2015.

[25] J. Zhang and P. Yu. Pct: Partial co-alignment of social networks. In *WWW*, 2016.

[26] J. Zhang, P. Yu, and Y. Lv. Organizational chart inference. In *KDD*, 2015.

[27] J. Zhang, P. Yu, Y. Lv, and Q. Zhan. Information diffusion at workplace. In *CIKM*, 2016.

[28] J. Zhang, P. Yu, and Z. Zhou. Meta-path based multi-network collective link prediction. In *KDD*, 2014.