Supervised Link Prediction by Geographic and Social Attributes

Wonhong Lee WONHONG@STANFORD.EDU
Department of Computer Science
Stanford University

S. Jun Yu SJYU@STANFORD.EDU
Institute for Computational & Mathematical Engineering
Stanford University

Abstract

In this paper, we employ learning algorithms to develop an efficient link prediction model based on geographic and social attributes.

1. Introduction

Link prediction in complex network is an active area of research in network analysis. This task is complicated by the fact that shape dynamics of the network is constantly changing, and it is difficult to define which inherent factors drive this change. In this paper, we will complement an existing algorithm by considering social, geographic, and demographic features to enhance the performance of link predictions.

1.1. Related Work

Backstrom and Leskovec [1] devised a link prediction model based on features involving personal attributes and network topology. This algorithm, however, may be infeasible in many cases due to limitations in gathering personal information. With a heightened awareness towards privacy issues, it has become more difficult to collect these data.

The model introduced by Liben-Nowell and Keinberg [3] predicts possible connections between nodes in a social network by using graph theoretic features. Although this algorithm is effective in making predictions for existing nodes, we do not have the same assurance for newly formed nodes since they do not hold any network topological information.

The main idea in the prediction model proposed by Scellato, Noulas, and Mascolo [7] is to incorporate geographic features such as physical distance and check-in data. This is a promising approach which demonstrates how features other than network topology and social attributes can be relevant in link prediction. We believe that there are new ways to render such geographic information to improve prediction results.

1.2. Problem Formulation

We want to develop a robust algorithm that can make accurate predictions for both existing and newly formed nodes. Some graph theoretic features, such as the Adamic-Adar score, play a crucial role in link prediction, and they often yield outstanding results for existing nodes. We will certainly incorporate these features in building a new prediction model.

As mentioned above, however, it is difficult to make predictions for newly formed nodes by solely analyzing the network topological properties. We run into similar obstacles when predicting possible links between a pair of nodes with distance greater than 3. It is easy to see how notions of topology defined for immediate neighbors might be insufficient for a meaningful prediction. To address such limitations, we complement topological features by employing new features of social, geographic, and demographic flavor.

1.3. Heuristic Overview

We consider three types of data sets: (1) $\mathcal{S}_2$, topological features are meaningful; (2) $\mathcal{S}_3$, topological features are unmeaningful; and (3) $\mathcal{S}_\infty$, newly formed nodes. Then we define features which are classified into three cateogires: (1) τ, topological features; (2) γ, geometric features; and (3) σ, social features. The basic idea is to train the prediction model on τ, $\tau \cup \gamma$, and $\tau \cup \gamma \cup \sigma$ separately for each data set, and see how performance is improved with the addition of each feature class.

Keywords: link prediction, supervised learning.
This project was carried out in collaboration with Jeongjin Ku from the Department of Computer Science.

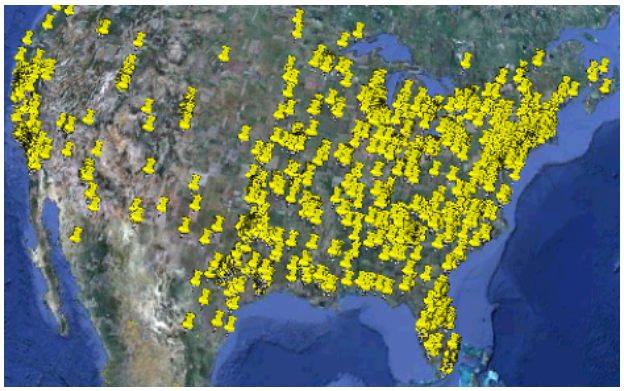
2. Data Rendering

We consider two types of data sets: (1) social network data obtained from Gowalla, an online location-based social network owned by Facebook; and (2) demographic data based on the 2000 United States Census which consists of demographics for each ZIP code area.

2.1. Social Network Data

For the first data set, we have friendship snapshots taken at July of 2010 and October of 2010, and public check-in history on February of 2009 and October of 2010 for users worldwide.

Not only is the size of the first data set massive, but the demographic information from the second data set is restricted to the United States.



We therefore extract information on the set of users with at least one check-in point in the United States. The following table shows the size of reduced data set.

Time of snapshot	Number of nodes	Number of edges
July of 2010	26,989	115,495
October of 2010	36,231	178,791

As for the check-in history, there are 3,742,003 locations for the two time frames combined.

For simplicity, let t_1 and t_2 denote the time, in chronological order, at which friendship snapshots were taken. We will also refer to users as nodes. Each node is uniquely assigned to a nonnegative integer.

Since the adjacency matrix for the reduced data set is extremely sparse, we only consider the set $\mathcal{A}$ of nodes incident to an edge that is present at t_2 but not at t_1. In other words, $\mathcal{A}$ consists of active nodes. We further define $\mathcal{A}_1$ as the subset of $\mathcal{A}$ with nodes present only at t_1, and set $\mathcal{A}_2 = \mathcal{A} \setminus \mathcal{A}_1$. Hence, $\mathcal{A}_2$ is the set of newly formed nodes. Note also that $\mathcal{A}_1$ and $\mathcal{A}_2$ partition $\mathcal{A}$.

Let u, v, and w be nodes in $\mathcal{A}$. The degree of u is denoted $\deg u$. We write $w \sim \{u, v\}$ when w is adjacent to both u and v. The distance between u and v is written as $d(u, v)$.

The set of all check-in locations for both time frames will be denoted by Λ. Hence, we do not distinguish between check-in points visited in different time frames. We also define $\lambda(u)$ to be the set of all check-in locations of $u \in \mathcal{A}$. Note that Λ is the disjoint union of $\lambda(u)$ for all $u \in \mathcal{A}$.

Furthermore, we select subsets of $\mathcal{A} \times \mathcal{A}$ from which we plan to build the training examples. First, let

$$\mathcal{S}_2 = \{(u, v) \in \mathcal{A}_1 \times \mathcal{A}_1 : d(u, v) = 2 \text{ with } u > v\}.$$

Many graph theoretic features are meaningful for this data set. Similarly, we define

$$\mathcal{S}_3 = \{(u, v) \in \mathcal{A}_1 \times \mathcal{A}_1 : d(u, v) = 3 \text{ with } u > v\}.$$

Observe that topological features defined for immediate neighbors are not useful in this case. Finally, the data set for newly formed nodes is given by

$$\mathcal{S}_\infty = \{(u, v) \in \mathcal{A}_1 \times \mathcal{A}_2 : u \sim v\}.$$

By testing the prediction model on these classes of training examples, we hope to demonstrate social, geographic, and demographic features significantly improve the accuracy of link predictions.

2.2. Demographic Data

The second data set consists of 19 fields representing various demographical attributes. The following table lists some of these features relevant to link prediction.

ZIP code	Area code
Population	Total population
Population density	Population per unit area
Geographic area	Urban, suburban, farm, non-farm
Race	White, Black, Asian, Indian, Hawaiian, other
Age	Age groups
Education	Education level of population over 18
Household income	Median household income
Per capita income	Median income per person
House value	Average value of homes
Housing density	Number of houses per unit area

For consistency, the check-in locations in the first data set are converted into an area code by using the Geopostal Service provided by Nuestar.

In order to take full advantage of the second data set, we must first deal with the missing values. Let $\mathcal{D}$ denote the entire demographic data set, where each element is a row vector r_i for the ith area code. We also write r_i' to denote the truncation of r_i, where the entry with the missing values are simply deleted. We can also form a column vectors c_j for the jth field, and define its trucation c_j' in a similar fashion. Note that the size of r_i' may vary for each i, and the same is true

of c'_j. Furthermore, we let μ_i and μ'_i denote the sample of the entries of r_i and r'_i, respectively. As for the sample covariance matrix, let Σ correspond to rows and columns of the observed entries, while Σ' corresponds to the rows of missing entries and the columns of the observed entries. We employ the expectation-maximiation algorithm to estimate the missing values.

Algorithm 1. (Expectation-Maximiation)

Initialization

(a) Set $z_{ij} = \mu_j$ and $\ell'(\Theta) = 0$, and choose $\epsilon > 0$.
(b) Set $i = 1$, $\ell_0(\Theta) = \ell(\Theta)$, and $\ell(\Theta) = 0$.

E-step

(c) Set $\mu'_i = \mu'_i + \Sigma'\Sigma^{-1}(r_i - \mu_i)$.
(d) Set $\ell(\Theta) = \ell(\Theta) - \frac{1}{2}\{(r_i - \mu_i)^T\Sigma^{-1}(r_i - \mu_i) - \log|\Sigma|\}$.
(e) Set $i = i + 1$, and if $i < n$, then go to (b).

M-step

(f) Set $\Theta = \arg\max_\Theta \ell(\Theta)$.
(g) If $|\ell(\Theta) - \ell_0(\Theta)| \geq \epsilon$, then go to (b), otherwise, break.

Note that, to get faster convergnece, we initialize the missing values by the sample mean of the observed entries in the corresponding column vectors.

3. Filter Feature Selection

We define and classify various features, and then run a feature selection algorithm to eliminate the insignificant ones.

3.1. Topological Features

The topological features are by far the most important features as they retain information on the graph theoretical properties of the network. The most natural topological feature is the number of common nodes, denoted τ_n. That is, given $u \in \mathcal{A}$ and $v \in \mathcal{A}$,

$$\tau_n(u, v) = \sum_{w \notin \{u,v\}} \mathbf{1}(w \sim \{u, v\}).$$

Observe that this feature does not take into account the fact that users corresponding to nodes with higher degree are more likely to be friends with a larger group of users.

The cosine similarity τ_c of $u, v \in \mathcal{A}$ is defined as

$$\tau_c(u, v) = \frac{\tau_n(u, v)}{\deg u \cdot \deg v}.$$

By incorporating this feature into the prediction model, we lend less significance to a pair of nodes with higher degree since users corresponding to these two nodes are more likely to have many common friends.

The Adamic-Adar score τ_a of $u, v \in \mathcal{A}$ is given by

$$\tau_a(u, v) = \sum_{w \sim \{u,v\}} \frac{1}{\log(\deg w)}.$$

We employ this feature to downgrade the effect of common nodes with higher degree since users corresponding to these nodes are more likely to be friends of a large group of users.

We also define the preferential attachment τ_p of $u \in \mathcal{A}$ and $v \in \mathcal{A}$ as

$$\tau_p(u, v) = \deg u \cdot \deg v.$$

This feature captures more active users corresponding to nodes with higher degree.

3.2. Geograhpic Features

We now define a set of geographic features based on the check-in history. Each check-in point is a physical location which can be written in the geographic coordinate system, that is, for $x \in \lambda(u)$ for some $u \in \mathcal{A}$,

$$\gamma_p(x) = (\theta, \phi),$$

where θ and ϕ are the latitude and longditude of $x(u)$, respectively.

The mode γ_m of $u \in \mathcal{A}$ is given by

$$\gamma_m(u) = \arg\max_{x \in \lambda(u)} \mathbf{P}(x(u)),$$

that is, the check-in location of u that occurs most frequently.

Similarly, the sample mean γ_s of $u \in \mathcal{A}$ is defined in the usual way as

$$\gamma_s(u) = \frac{\sum_{x \in \lambda(u)} x}{\sum_{x \in \Lambda} \mathbf{1}(x \in \lambda(u))},$$

that is, the arithmetic mean of the check-in locations of u.

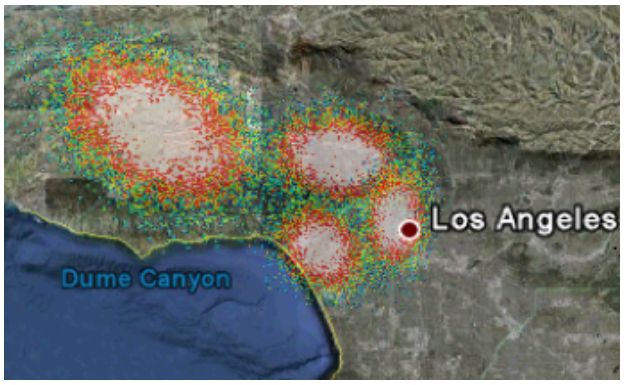
We would also like to define a feature that captures the intuition of communities within a network. To do this, we repeatedly apply k-means clustering to form a binary decision tree. We proceed as follows.

Algorithm 2. (Modified k-Means Clustering)

(a) Set $k = 2$ and choose $\sigma^2 > 0$.
(b) For $u \in \mathcal{A}$, set $\ell_1 = \{\lambda(u)\}$ and $\ell_2 = \ell(u) = \emptyset$.
(c) For $\lambda \in \ell_1$, run k-means clustering to get λ_1 and λ_2.
(d) If $\mathrm{var}(\lambda_1) \geq \sigma^2$, then $\lambda_1 \in \ell_2$; otherwise, $\lambda_1 \in \ell(u)$.
(e) If $\mathrm{var}(\lambda_2) \geq \sigma^2$, then $\lambda_2 \in \ell_2$; otherwise, $\lambda_2 \in \ell(u)$.

(f) Set $\ell_1 = \ell_1 \backslash \{\lambda\}$, and if $\ell_1 \neq \emptyset$, then go to (c).
(g) Set $\ell_1 = \ell_2$ and $\ell_2 = \emptyset$, and go to (c).

The elements of $\ell(u)$ for $u \in \mathcal{A}$ in Algorithm 2 are precisely the leaves of the binary decision tree for u. We now let $\mu(u)$ denote the mean of the cluster in $\ell(u)$ with the largest number of check-in points.



The figure above shows the result of applying this algorithm to a user living in California. Now the clustering distance between $u \in \mathcal{A}$ and $v \in \mathcal{A}$ is defined as

$$\gamma_c(u, v) = \|\mu(u) - \mu(v)\|_2,$$

that is, the Euclidean distance between the mean of the largest clusters in $\ell(u)$ and $\ell(v)$.

3.3. Social Features

Although 27 social features are considered in the prediction model, we only discuss a few of the important ones as the rest are defined similarly. We write $N(u)$ to denote the total population of the area code for $u \in \mathcal{A}$.

We define the housing density σ_h of $u \in \mathcal{A}$ as

$$\sigma_h(u) = \frac{H(u)}{A(u)},$$

where $H(u)$ is the number of houses in the area code for u.

The density of white population σ_w for $u \in \mathcal{A}$ is

$$\sigma_w(u) = \frac{W(u)}{N(u)},$$

where $W(u)$ is the white population of the area code for u.

We write the per capita income σ_p of $u \in \mathcal{A}$ as

$$\sigma_p(u) = \frac{I(u)}{N(u)},$$

where $I(u)$ is the net income of residents in the area code for u.

Finally, the urban population density σ_u of $u \in \mathcal{A}$ is defined as

$$\sigma_u(u) = \frac{U(u)}{N(u)},$$

where $U(u)$ is the population living in urban areas within the area code for u.

3.4. Mutual Information

We now compute the Kullback-Leibler divergence for each feature, and eliminate features with low scores.

Feature	MI Score	Feature	MI Score	Feature	MI Score
τ_n	0.0435	γ_p	0.0988	σ_h	0.0737
τ_c	0.0205	γ_m	0.1456	σ_w	0.0435
τ_a	0.0850	γ_s	0.1042	σ_p	0.0620
τ_p	0.1670	γ_c	0.1523	σ_u	0.0298

Among the 27 social features, many of which are not listed in the table above, we eliminated the ones with scores below 0.01. For instance, σ_f, the population density of farmers had the lowest score of 9.7×10^{-4}.

Now among the selected features, we let τ, γ, and σ denote the set of topological, geographical, and social features, respectively. For $\mathcal{S}_2$, by training on sets

$$\mathcal{X}_2 = \mathcal{S}_2 \cup \tau, \ \mathcal{X}_2' = \mathcal{X}_2 \cup \gamma, \ \mathcal{X}_2'' = \mathcal{X}_2' \cup \sigma,$$

we hope to observe how the addition of γ and σ enhance prediction outcomes. As for, $\mathcal{S}_3$, we consider

$$\mathcal{X}_3 = \mathcal{S}_3 \cup \{\tau_p\}, \ \mathcal{X}_3' = \mathcal{X}_3 \cup \gamma, \ \mathcal{X}_3'' = \mathcal{X}_3' \cup \sigma.$$

Note that we only include τ_p from τ as this is the only topological feature relevant to link predition for $\mathcal{S}_3$. Similarly, we define the training sets

$$\mathcal{X}_\infty = \mathcal{S}_\infty \cup \{\tau_p\}, \ \mathcal{X}_\infty' = \mathcal{X}_\infty \cup \gamma, \ \mathcal{X}_\infty'' = \mathcal{X}_\infty' \cup \sigma$$

for $\mathcal{S}_\infty$. We expect γ and σ to play an even bigger role in this case.

4. Supervised Learning Algorithms

We carry out three different learning algorithms on each training set defined in Section 3.4.

4.1. Ridge Logistic Regression

We first implement the ridge regression [2], where we want to find the maximum likelihood estimator $\hat{\theta}$ for

$$\ell(\theta) = \sum_{i=1}^{m} \log h(x^{(i)})^{y^{(i)}} (1 - h(x^{(i)}))^{1-y^{(i)}} + \lambda \theta^T \theta,$$

where h for parameter θ is given by

$$h(x; \theta) = \frac{1}{1 + e^{-\theta^T x}}.$$

Note that we have added the L^2-norm of θ as a penalty term to the log-likelihood function ℓ.

4.2. Naive Bayes Classifier

Given a new example with feature x, we have that

$$\mathbf{P}(x|y) = \prod_{j=1}^{n} \mathbf{P}(x_j|y)$$

by the naive Bayes assumption. We use the parameters

$$\phi_{x|y=0} = \mathbf{P}(x_i = x|y = 0) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-(x-\mu_i)^2/2\sigma_i^2},$$
$$\phi_{x|y=1} = \mathbf{P}(x_i = x|y = 1) = \frac{1}{\sqrt{2\pi\tau_i^2}} e^{-(x-\nu_i)^2/2\tau_i^2},$$

along with

$$\phi_{y=1} = \mathbf{P}(y = 1).$$

to estimate the posterior probability of a new example with features x as

$$\mathbf{P}(y = k|x) = \frac{\mathbf{P}(x|y = k)\mathbf{P}(y = k)}{\mathbf{P}(x)},$$

where $k \in \{0, 1\}$.

4.3. Soft Margin Support Vector Machine

We apply the ν-soft margin classifier [5] with parameter $0 \leq \nu \leq 1$, where we minimize

$$\frac{1}{2} w^T w - \nu\rho + \frac{1}{m} \sum_{i=1}^{m} \xi_i$$

subject to the constraints

$$y_i(w^T x_i + b) \geq \rho - \xi_i,\ \xi_i \geq 0,\ \text{and}\ \rho \geq 0$$

for $1 \leq i \leq n$. The parameter $0 \leq \nu \leq 1$ is a lower bound on the fraction of support vectors. Note that we have an additional variable ρ, where upon setting $\xi_i = 0$ for $1 \leq i \leq n$, the separating hyperplane is given by $2\rho/w^T w$.

4.4. Prediction Results by Cross Validation

For each of the learning algorithms, we carry out the k-fold cross validation for l models as follows.

Algorithm 3. (Cross Validation)

(a) Randomly partion S into $S_1, ..., S_k$; set $i = 1$, $j = 1$.
(b) Train M_i on $\cup_{p \neq j} S_p$ to find hypothesis h_{ij}.
(c) Test h_{ij} on S_j to find ϵ_{ij}; set $j = j + 1$.
(d) If $j < k$, go to (b); else if, go to (e).
(e) Set $\epsilon_i = \frac{1}{k} \Sigma_{1 \leq j \leq k} \epsilon_{ij}$, $i = i + 1$, $j = 1$.
(f) If $i < l$, go to (b); else if, set $i = \arg\min_{1 \leq i \leq l} \epsilon_i$.
(g) Retrain model M_i on S to find hypothesis h.

Here, ϵ_{ij} is the training error. We used $k = 10$ with $l = 3$ corresponding to naive Bayes, logistic regression, and soft margin classifiers.

The following table shows the F1 scores when the classifiers are trained by the set of features built on $\mathcal{S}_2$.

Classifier	$\mathcal{X}_2$	$\mathcal{X}_2'$	$\mathcal{X}_2''$
Naive Bayes	77.1	80.3	81.3
Logistic Regression	76.7	80.1	82.4
Support Vector Machine	76.3	80.0	81.6

Since all three classifiers perform similarly, the k-fold cross validation is not very meaningful in this case. Although τ alone return high F1 scores, it is evident that γ and σ significantly enhance the accuracy of link predictions.

As for the set of features built on $\mathcal{S}_3$, the k-fold cross validation selects logistic regression as it returns the higest F1 scores for each training set.

Classifier	$\mathcal{X}_3$	$\mathcal{X}_3'$	$\mathcal{X}_3''$
Naive Bayes	48.1	65.3	80.1
Logistic Regression	59.2	70.1	87.3
Support Vector Machine	40.3	62.1	79.2

Observe that all three classifiers experience a steep learning curve despite the low F1 scores for $\mathcal{X}_3$. In this case, γ and σ play a crucial role in improving the performance of all three classifiers.

We now look at the F1 scores corresponding to the set of features built on $\mathcal{S}_\infty$.

Classifier	$\mathcal{X}_\infty$	$\mathcal{X}_\infty'$	$\mathcal{X}_\infty''$
Naive Bayes	74.1	80.3	79.1
Logistic Regression	76.7	80.1	78.2
Support Vector Machine	65.3	70.1	77.2

As with the previous cases, we see an overall improvement with the addition of γ and σ into the feature set. The naive Bayes classifier has a slightly higher F1 score for $\mathcal{X}_\infty''$ than the other two.

References

1. L. Backstrom, J. Lescovec, *Supervised Random Walks: Predicting and Recommending Links in Social Networks*, In Proc. ACM WSDM. (2011)
2. S. le Cessie, J. van Houwelingen, *Ridge Estimators in Logistic Regression*, Appl. Statist. **41** No. 1. (1992)
3. D. Liben-Nowell, J. Kleinberg, *The Link Prediction Problem for Social Networks*, In International Conference on Information and Knowledge Management. (2003)
4. D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, A. Tomkins, *Geographic Routing in Social Networks*, PNAS 102(**33**). (2005)
5. P. Chen, C. Lin, B. Schölkopf, *A Tutorial on ν-Support Vector Machines*, Applied Stochastic Models in Business and Industry **21**. (2005)
6. A. Narayanan, V. Shmatikov, *De-anonymizing Social Networks*, In Proc. of IEEE Symposium on Security and Privacy. (2009)
7. S. Scellato, A. Noulas, C. Mascolo, *Exploiting Place Features in Link Prediction on Location-based Social Networks*, In Proceedings of 17th ACM International Conference on Knowledge Discovery and Data Mining. (2011)