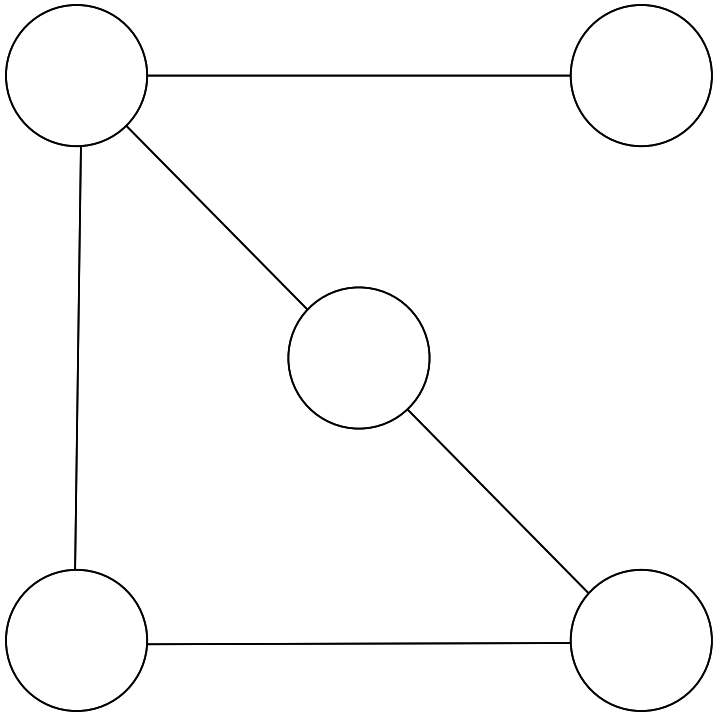
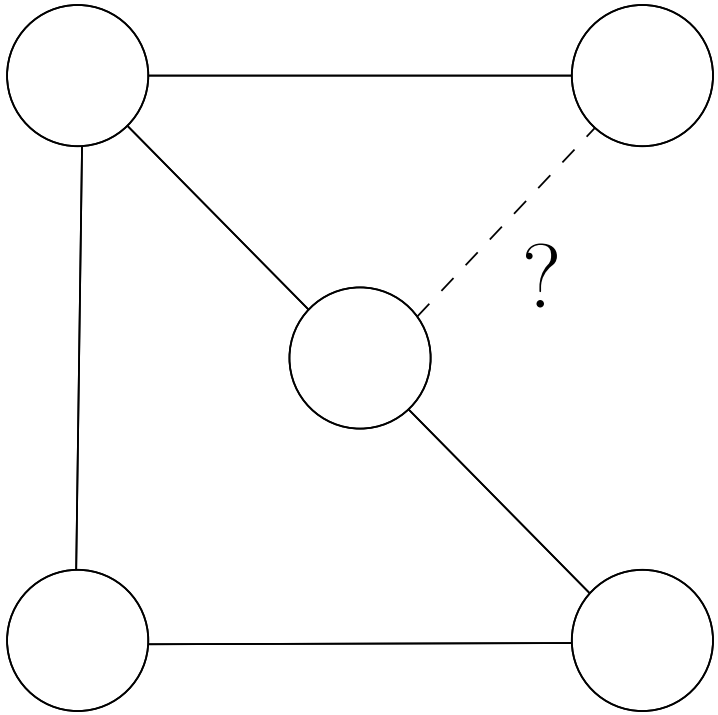
Data Mining
Mining Social Network Data: Link Prediction

Ad Feelders

Universiteit Utrecht

The Link Prediction Problem





Applications

- Biology: protein-protein interaction prediction.
- Recommendation systems, e.g. link recommendation in social networks like Facebook.
- Analysis of criminal/terrorist networks.
- Automatic web hyper-link creation (e.g. discovering missing links in WikiPedia).
- Record linkage/deduplication.
- ...

Example: Predicting Romantic Relationships

The latest offering from Facebooks data-science team teases out who is romantically involved with whom by examining link structures. It turns out that if one of your Facebook friends - lets call him Joe - has mutual friends that touch disparate areas of your life, and those mutual friends are themselves not extensively connected, its a strong clue that Joe is either your romantic partner or one of your closest personal friends.

`http://www.technologyreview.com/view/520771/now-facebook-can-see-inside-your-heart-too/`

Lars Backstrom and Jon Kleinberg: *Romantic Partnerships and the Dispersion of Social Ties: A Network Analysis of Relationship Status on Facebook*, Proc. 17th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW), 2014

The Link Prediction Problem

Given a (social) network $G = (V, E)$ in which an edge $e = (u, v) \in E$ represents some form of interaction between its endpoints at a particular time $t(e)$.

Multiple interactions between the same pair of nodes can be recorded by parallel edges or using multiple time-stamps for an edge.

The Link Prediction Problem

For time $t \leq t'$, let $G[t, t']$ denote the subgraph of G restricted to the edges with time-stamps between t and t'.

Supervised learning problem: choose a training interval $[t_0, t_0']$ and a test interval $[t_1, t_1']$ where $t_0' < t_1$.

The link prediction problem is to output a list of edges not present in $G[t_0, t_0']$, but are predicted to appear in the network $G[t_1, t_1']$.

Approaches to link prediction

1. Construct (similarity-) based features for pairs of nodes, and use binary classification algorithms to predict the existence/emergence of links.
2. Use node embeddings. Similar nodes should get similar embedding vectors.
3. Probabilistic models. Fit a statistical model based on the structure of the network.
4. ...

We will focus on the first approach.

Node Neighborhood Based Features

Let $\Gamma(x)$ denote the neighborhood of node x, that is, the set of nodes directly connected to x.

For two nodes x and y, we define:

1. Number of shared neighbors:

$$\text{Common-Neighbors}\,(x,y) = |\Gamma(x) \cap \Gamma(y)|$$

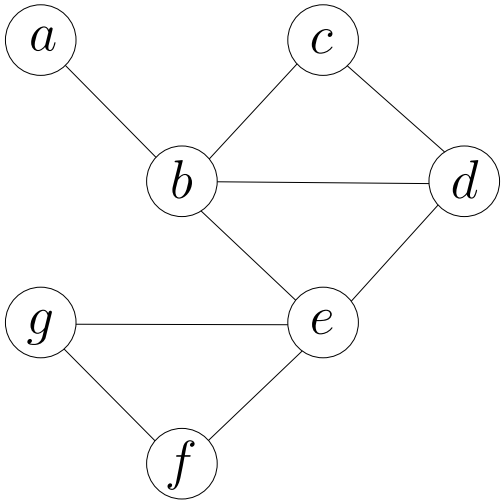
2.

$$\text{Jaccard-Coefficient}\,(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

3. A shared neighbor that is itself not heavily connected gets higher weight:

$$\text{Adamic-Adar}\,(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|}$$

Examples



- $\Gamma(c) = \{b, d\} \quad \Gamma(e) = \{b, d, f, g\}$
- Common-Neighbors $(c, e) = |\Gamma(c) \cap \Gamma(e)| = |\{b, d\}| = 2$
- Jaccard-Coefficient $(c, e) = \frac{|\Gamma(c) \cap \Gamma(e)|}{|\Gamma(c) \cup \Gamma(e)|} = \frac{|\{b,d\}|}{|\{b,d,f,g\}|} = \frac{2}{4}$

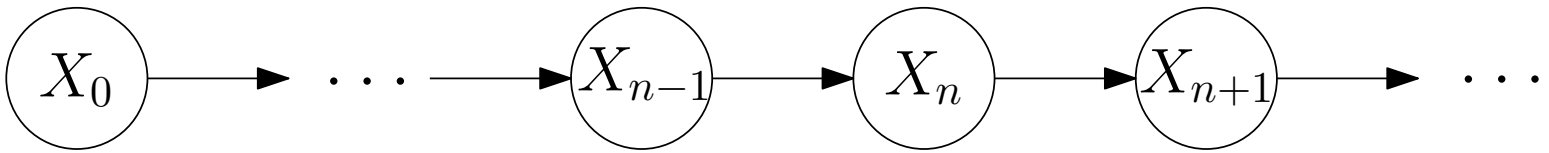
Intermezzo: Markov Chain

Let P be a $k \times k$ matrix with elements P_{ij}.

A stochastic process $(X_0, X_1, \ldots)$ with state space $S = \{s_1, \ldots, s_k\}$ is said to be a Markov chain with transition matrix P if for all $i, j \in \{1, \ldots, k\}$ we have

$$\begin{aligned} P(X_{n+1} = s_j \mid X_n = s_i, X_{n-1}, \ldots, X_0) &= P(X_{n+1} = s_j \mid X_n = s_i) \\ &= P_{ij} \end{aligned}$$

Slogan: the future is independent of the past given the present.



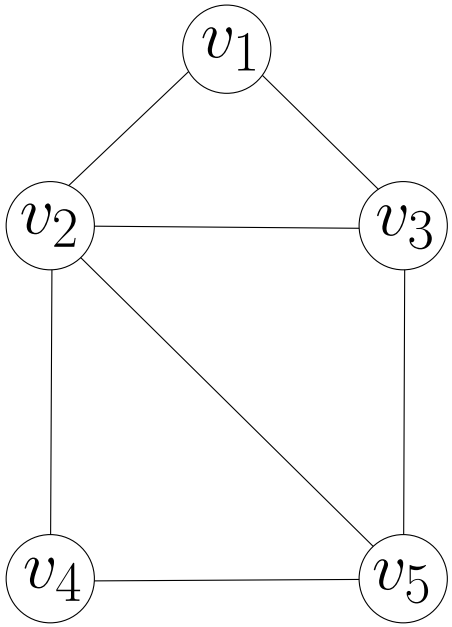
Random Walk on Graph

A random walk on a graph $G = (V, E)$ is a Markov chain with state space $V = \{v_1, v_2, \ldots, v_k\}$.

If the random walker stands at vertex v_i at time n, then it moves at time $n + 1$ to one of the neighbors of v_i chosen at random, with equal probability for each of the neighbors. More formally:

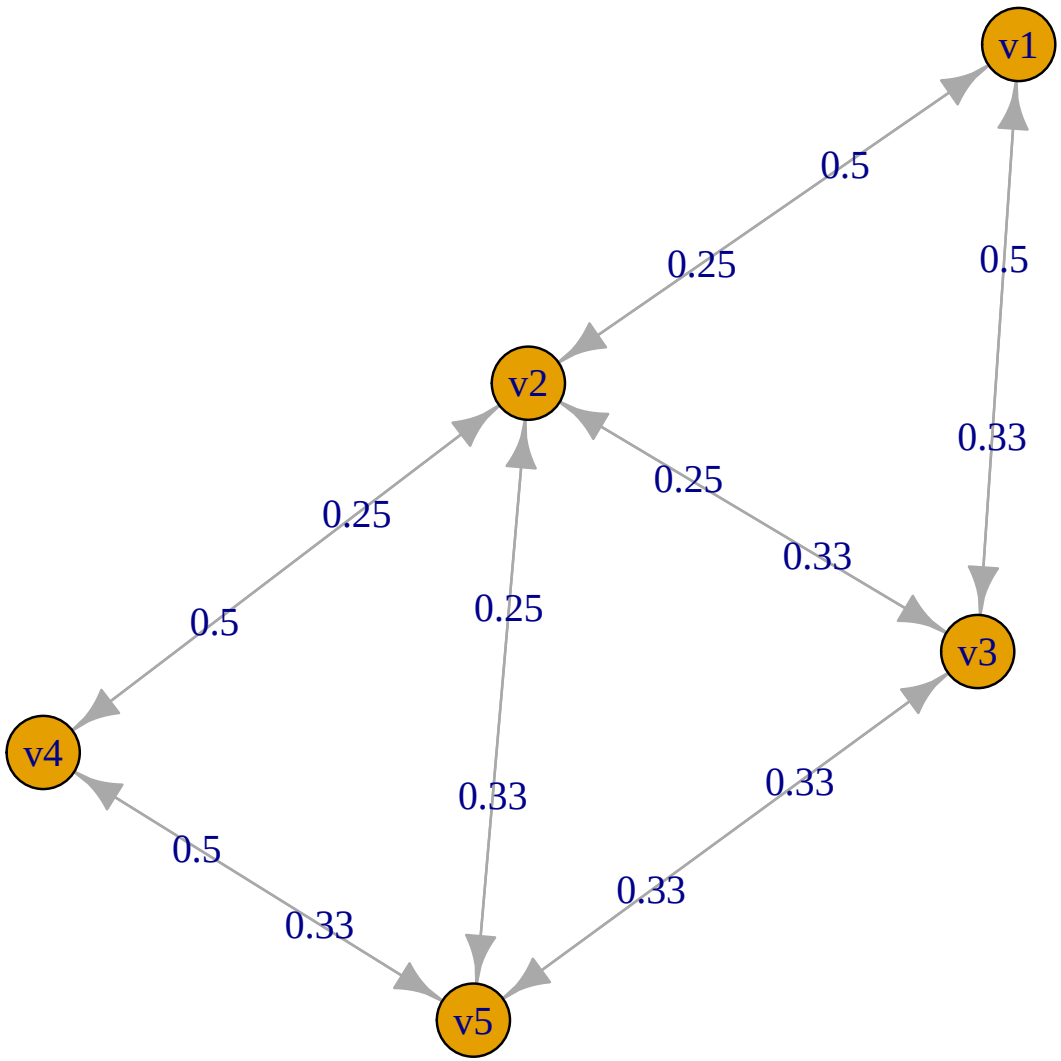
$$P_{ij} = \begin{cases} \frac{1}{|\Gamma(v_i)|} & \text{if } v_j \in \Gamma(v_i) \\ 0 & \text{otherwise} \end{cases}$$

Example Graph and Transition Matrix



$$P = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix}$$

Transition Diagram



Stationary Distribution

Let $(X_0, X_1, \ldots)$ be a Markov chain with state space $S = \{s_1, \ldots, s_k\}$ and transition matrix P. A row vector $\pi = (\pi_1, \ldots, \pi_k)$ is said to be a stationary distribution for the Markov chain, if it satisfies:

1. $\pi_i \geq 0$ for $i = 1, \ldots, k$ and $\sum_{i=1}^{k} \pi_i = 1$, and
2. $\pi P = \pi$, meaning that

$$\sum_{i=1}^{k} \pi_i P_{ij} = \pi_j, \text{ for } j = 1, \ldots, k$$

The second property implies that if the initial distribution $\mu^{(0)} \equiv P(X_0)$ equals π, then the distribution $\mu^{(1)}$ of the chain at time 1 satisfies:

$$\mu^{(1)} = \mu^{(0)} P = \pi P = \pi,$$

and by iterating we see that $\mu^{(n)} = \pi$ for every n.

Stationary Distribution for Random Walk on Graph

The stationary distribution is:

$$\pi = \left(\frac{2}{14}, \frac{4}{14}, \frac{3}{14}, \frac{2}{14}, \frac{3}{14}\right)$$

Check

$$\left(\frac{2}{14}, \frac{4}{14}, \frac{3}{14}, \frac{2}{14}, \frac{3}{14}\right) \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix} = \left(\frac{2}{14}, \frac{4}{14}, \frac{3}{14}, \frac{2}{14}, \frac{3}{14}\right)$$

For example: in the long run, the random walker will find itself in node v_2 for $\frac{4}{14} \approx 29\%$ of the time.

Existence and Convergence

1. Existence: every irreducible and aperiodic Markov chain has a unique stationary distribution.
2. Convergence: If we run the Markov chain for a sufficiently long time n, then regardless of what the initial distribution $\mu^{(0)}$ was, the distribution $\mu^{(n)}$ at time n will be close to the stationary distribution π. This is often referred to as the Markov chain approaching equilibrium as $n \to \infty$.

Irreducible and Aperiodic

A Markov chain is *irreducible* if every state is reachable from every other state.

The period $d(s_i)$ of a state $s_i \in S$ is defined as

$$d(s_i) = \gcd\{n \geq 1 : (P^n)_{i,i} > 0\}$$

The period of s_i is the greatest common divisor of the set of times for which the chain can return (i.e. has positive probability of returning) to s_i, given that we start with $X_0 = s_i$.

If $d(s_i) = 1$, then we say that the state s_i is *aperiodic*.

A Markov chain is said to be *aperiodic* if all its states are aperiodic. Otherwise, the chain is said to be periodic.

Path Based Features

For two nodes x and y:

- Shortest path distance between x and y.
- Katz:

$$\text{Katz}\,(x,y) = \sum_{\ell=1}^{\infty} \beta^{\ell} |\text{paths}_{x,y}^{\langle \ell \rangle}|$$

  where $|\text{paths}_{x,y}^{\langle \ell \rangle}|$ is the number of length-ℓ paths from x to y. A very small β yields predictions much like common neighbors, since paths of length three or more contribute very little to the summation.
- Hitting time $H(x,y)$ between nodes x and node y is the expected number of steps before node y is visited for the first time, in a random walk starting from node x .
- Commute time: $C(x,y) = H(x,y) + H(y,x)$.

Path Based Features

- Rooted Pagerank(x, y): stationary probability of y in a random walk starting at x that returns to x with probability α at each step, and moves to a random neighbor with probability $1 - \alpha$.

Vertex Feature Aggregation

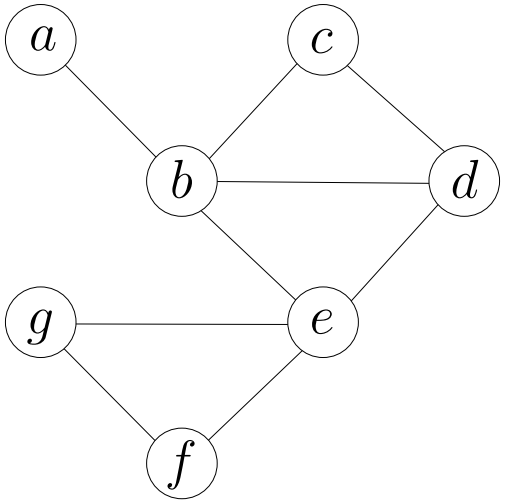
1. Preferential Attachment Score between x and y: $|\Gamma(x)| \cdot |\Gamma(y)|$.
2. Sum of Neighbors of x and y: $|\Gamma(x)| + |\Gamma(y)|$.
3. Clustering coefficient:

$$\text{clustering-coefficient}\,(u) = \frac{2\,|\{(v,w) \in E : v, w \in \Gamma(u)\}|}{|\Gamma(u)|(|\Gamma(u)| - 1)}$$

This is the number of neighbor pairs of u that are neighbors of each other, divided by the total number of neighbor pairs of u. For example, if edges represent collaborations between people (nodes), it's the fraction of pairs of a persons collaborators who have also collaborated with one another.

For a pair of nodes, we can use the sum or product of their clustering coefficients as a feature.

Examples



- Shortest path distance between c and e is 2.
- clustering-coefficient $(e) = \frac{2\times 2}{4\times 3} = \frac{1}{3}$

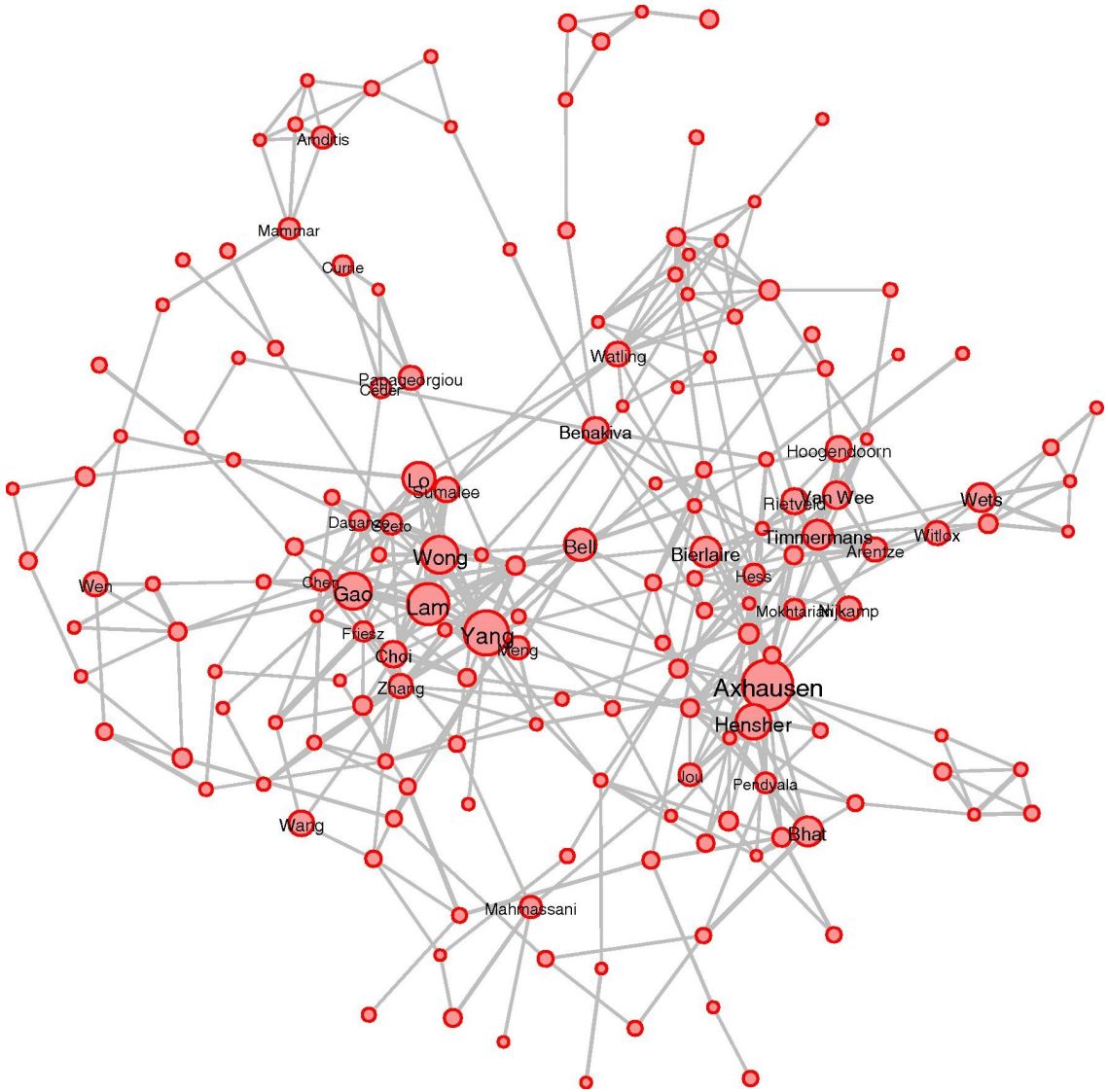
Required Literature

David Liben-Nowell, Jon Kleinberg: *The Link Prediction Problem for Social Networks*, Proceedings of the Twelfth Annual ACM International Conference on Information and Knowledge Management (CIKM03), November 2003, pp. 556–559.

Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki: *Link Prediction Using Supervised Learning*, SDM Workshop on Link Analysis, 2006.

The remaining slides are about the second paper.

Co-authorship Network



Data Sets

Dataset	Number of Papers	Number of Authors
BIOBASE	831,478	156,561
DBLP	540,459	1,564,617

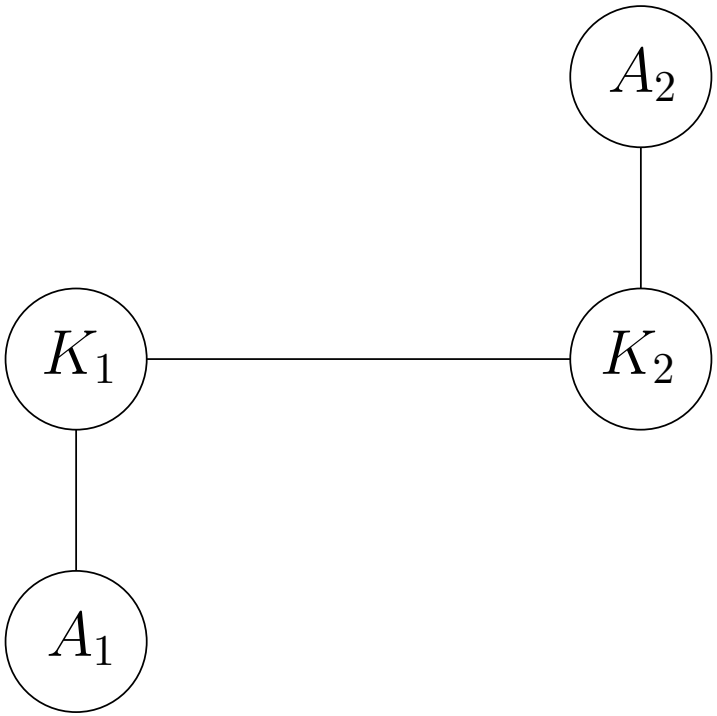
Consider the pairs of nodes not linked in $G[t_0, t_0']$, and give them class label 1 (positive) if they are linked in $G[t_1, t_1']$, and class label 0 (negative) otherwise.

- BIOBASE: 5 years of data from 1998 to 2002 (first 4 for training).
- DBLP: 15 years of data, from 1990 to 2004 (first 11 for training).
- Positive/Negative pairs chosen randomly in equal proportion from pairs that qualify.
- Construct feature vector for each pair of authors.

Some Additional Features Used

- Keyword match count (x, y): the number of shared keywords of papers written by x and papers written by y.
- Sum of keyword count: researchers that have a wide range of interests or those who work on interdisciplinary research usually use more keywords. In this sense they have better chance to collaborate with new researchers.
- Shortest distance in author-keyword graph: the author-keyword graph extends the co-authorship graph with nodes that correspond to keywords. Each keyword node is connected to an author node, if that keyword is used by the author in any of his papers. Moreover, two keywords that appear together in any paper are also connected by an edge.

Author-Keyword Graph



- Author 1 wrote a paper with Keyword 1.
- Author 2 wrote a paper with Keyword 2.
- Keyword 1 and Keyword 2 appeared together in some paper.

Experiments

- Seven different classification algorithms, among which classification (decision) trees and naive Bayes.
- Used 5-fold cross-validation to evaluate performance.
- Separate validation set for hyperparameter tuning.
- Measure accuracy, precision, recall, F_1 score.

Results

Classification model	Accuracy	Precision	Recall	F-value	Squared Error
Decision Tree	90.01	91.60	89.10	90.40	0.1306
SVM(Linear Kernel)	87.78	92.80	83.18	86.82	0.1221
SVM(RBF Kernel)	90.56	92.43	88.66	90.51	0.0945
K_Nearest Neighbors	88.17	92.26	83.63	87.73	0.1826
Multilayer Perceptron	89.78	93.00	87.10	90.00	0.1387
RBF Network	83.31	94.90	72.10	81.90	0.2542
Naive Bayes	83.32	95.10	71.90	81.90	0.1665
Bagging	90.87	92.5	90.00	91.23	0.1288

Table 2: Performance of different classification algorithms for BIOBASE database

Classification model	Accuracy	Precision	Recall	F-value	Squared Error
Decision Tree	82.56	87.70	79.5	83.40	0.3569
SVM(Linear Kernel)	83.04	85.88	82.92	84.37	0.1818
SVM(RBF Kernel)	83.18	87.66	80.93	84.16	0.1760
K_Nearest Neighbors	82.42	85.10	82.52	83.79	0.2354
Multilayer Perceptron	82.73	87.70	80.20	83.70	0.3481
RBF Network	78.49	78.90	83.40	81.10	0.4041
Naive Bayes	81.24	87.60	76.90	81.90	0.4073
Bagging	82.13	86.70	80.00	83.22	0.3509

Table 3: Performance of different classification algorithms for DBLP dataset

Feature Ranking

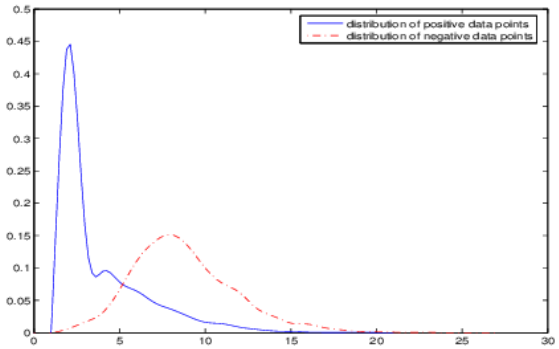
Attribute name	Information gain	Gain Ratio	Chi-Square Attribute Eval.	SVM feature evaluator	Avg. Rank
Sum of Papers	3	4	3	4	3
Sum of Neighbors	1	3	1	2	2
Sum of KW count	6	6	6	3	5
Sum of Classification count	5	5	5	6	5
KW match count	2	1	2	1	1
Sum of log of Sec. Neighbor. count	7	7	7	8	7
Shortest distance	4	2	4	5	4
Clustering Index	9	9	9	7	8
Shortest dist. in KW-Author graph	8	8	8	9	8

Table 4: Rank of different Attributes using different algorithms for BIOBASE dataset

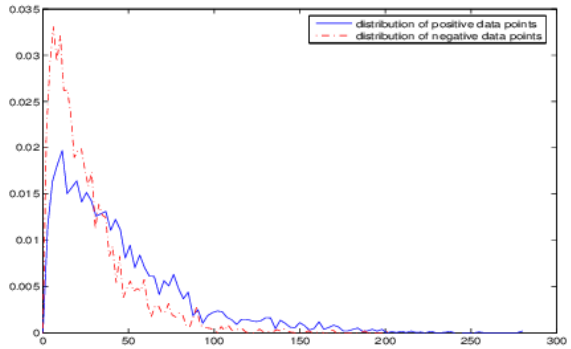
Attribute name	Information gain	Gain Ratio	Chi-Square Attribute Eval.	SVM feature evaluator	Avg. Rank
Sum of Papers	4	4	4	2	4
Sum of Neighbors	3	3	3	4	3
Shortest distance	1	1	1	1	1
Second shortest distance	2	2	2	3	2

Table 5: Rank of different Attributes using different algorithms for DBLP dataset

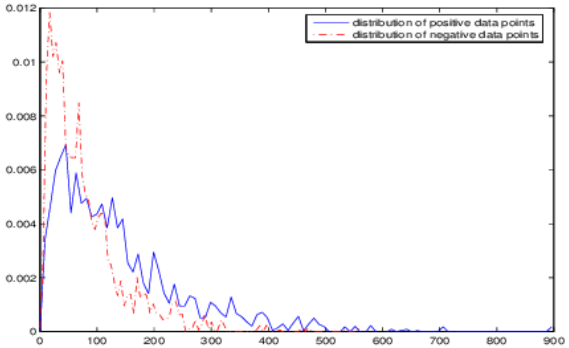
Feature Distributions



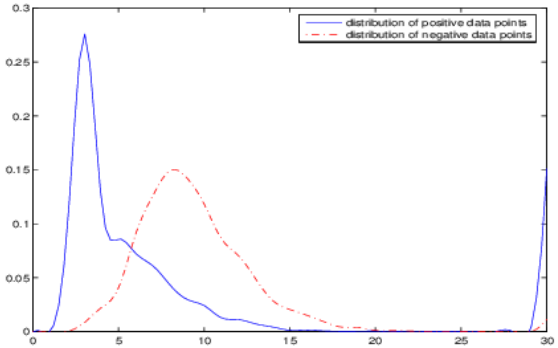
(a) Shortest Distance



(b) Sum of paper count



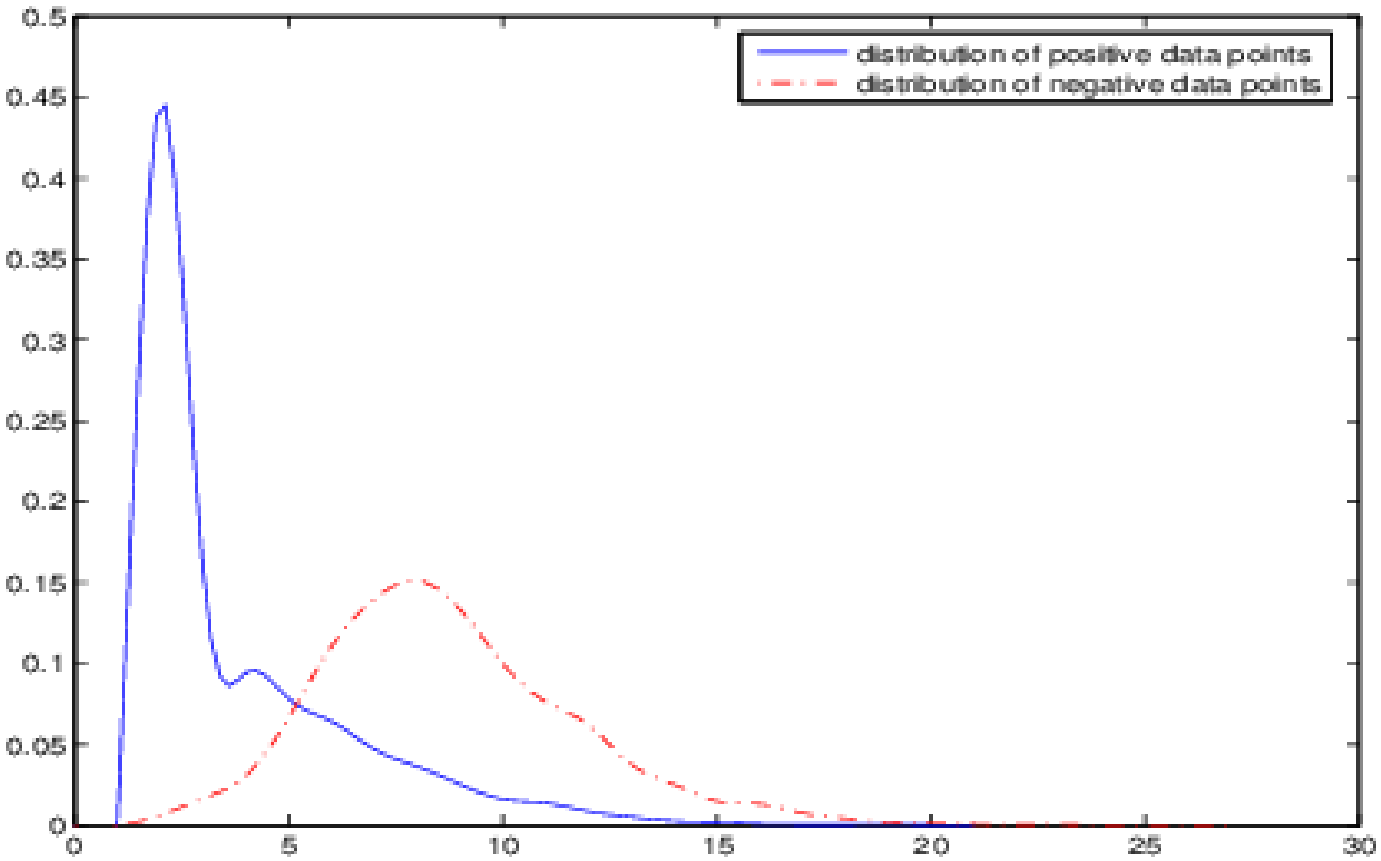
(c) Sum of neighbors count



(d) Second shortest distance

Figure 2: Evaluation of features using class density distribution in DBLP dataset

Feature Distributions



(a) Shortest Distance