Node similarity and classification

Davide Mottin, Anton Tsitsulin

Hasso Plattner Institute

Graph Mining course Winter Semester 2017
Acknowledgements

- Some part of this lecture is taken from: http://web.eecs.umich.edu/~dkoutra/tut/icdm14.html
- Other adapted content is from Social Network Data Analytics (Springer) Ed. Charu Aggarwal, March 2011
SimRank

Idea: two objects are similar if they are referenced by similar objects.

\[ s(a, b) = \text{Avg similarity between in-neighbors of } \alpha \text{ and in-neighbors of } b \]

- decay factor in [0,1]
- total # of in-neighbors pairs
- similarity of in-neighbors

Glen Jeh and Jennifer Widom. SimRank: a measure of structural-context similarity. SIGKDD, 2002
SimRank: Intuition

Intuition: Computing SimRank is like propagating on the $G^2$ graph of node-node pairs

1. The source of similarity is self-vertices, like (Univ, Univ).
2. Similarity propagates along pair-paths in $G^2$, away from the sources.
SimRank for Bipartite Graphs

\[ s(A, B) = \text{Avg similarity between out-neighbors of } A \text{ and out-neighbors of } B \]

\[ s(c, d) = \text{Avg similarity between in-neighbors of } c \text{ and in-neighbors of } d \]

[Jeh, Widom ’02; Improvements: Antonellis+’08 SimRank++, C. Li, Han+’10, Y. Zhang ’13, P. Li+’14 ...]
Lecture road

Similarity based

Iterative classification

Label propagation
Iterative classification methods

Idea:
Use features that take into account the neighbor nodes and repeat the classification several times until nothing changes.

Suppose for each node we have two features:
1. Number of neighbors with class A
2. Number of neighbors without a class

Learn Labels and apply to nodes

Recompute features on unlabeled

Neville, J. and Jensen, D., 2000. Iterative classification in relational data. AAAI.
Iterative Classification Algorithm (ICA)

- Train classifier using the labeled instances
- Until convergence
  - Apply classifier to the unlabeled nodes
  - Updated the feature vectors for unlabeled nodes
- Return the labels for the labeled nodes

Neville, J. and Jensen, D., 2000. *Iterative classification in relational data*. AAAI.
Extension to multi-labels

- Each node has a distribution over the labels
- To avoid noise keep only the top-k labels for each unlabeled node sorted in descending order.
  
  • Intuition: remove the less confident labels

Lecture road

- Similarity based
- Iterative classification
- Label propagation
Guilt-by-Association Techniques

Given:
• graph and
• few labeled nodes

Find: class \((\text{red}/\text{green})\) for rest nodes

Assuming: network effect (homophily/heterophily)
Personalized Random Walk with Restarts (RWR)

**Idea:** Propagate labels from a set of nodes to the rest of the graph

Google [Brin+ ’98; Haveliwala ’03; Tong+ ’06; Minkov, Cohen ’07]
PageRank: A kind of random walk

The importance of John is high if Laura, Andrew, and Bill are also important.

\[ P_{\text{Rank}}(\text{John}) = \frac{P_{\text{Rank}}(\text{Bill})+P_{\text{Rank}}(\text{Laura})+P_{\text{Rank}}(\text{Andrew})}{\text{Number of in-links}} \]
Random Walk

- In a random walk you do the opposite, you assume that the walker moves randomly and chooses one of the neighbors to visit.

1. Tom chooses Andrew or Laura with probability $1/2$.
2. Once he chooses one it increases the number of times he visited that node.
3. Continue the process until nothing changes anymore (at a probabilistic level).

John will receive many visits since many nodes are connected to him.
Now assume that with probability $c$ you perform another move and with probability $(1-c)$ you jump back to Tom.

Therefore the probability for the walker of being in Tom place will be

$$Tom(t) = c \frac{\text{Prob}(\text{Andrew})(t - 1) + \text{Prob}(\text{Laura})(t - 1)}{3} + (1 - c)$$

Number of Andrew and Laura’s neighbors

Probability of visiting Andrew at time (t-1)

Probability of jumping back to Tom
Comparing two nodes

- Start two random walks from the two nodes you want to compare separately.
- Compare the final scores you obtain for each node in the graph using some vector comparison (e.g., cosine similarity, KL-divergence).

Vector for Tom = [0.2, 0.2, 0.1, 0.3, …, 0.01]
Vector for John = [0.05, 0.15, 0.2, 0.2, …, 0.2]

Compare the two vectors (e.g., subtract)
Personalized RWR

Now assume that with probability $c$ you perform another move and with probability $(1-c)$ you jump back to Tom.

$$[I - cD^{-1}A]x = (1 - c)y$$

Graph structure: $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

Relevance vector: $\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$

Starting vector: $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
Personalized RWR

- Personalized RWR is defined as the probability of a random surfer of reaching a node $n$ after ”wandering” in the graph for a ”long time”
- After some time the value for $n$ (and for all the other nodes) does not changes anymore
  - Technically speaking the random walk converges to a stationary distribution
- Let $A$ be the adjacency matrix of graph $G = \langle V, E \rangle$
  
  \[
x = cD^{-1}Ax + (1 - c)y
  \]

  where $D$ is a diagonal matrix with the node degree in the diagonal and $y$ is a vector containing the probability of starting from any of the nodes $y_i$, $x$ is the probability of reaching any node in the graph $\dim(x) = \dim(y) = |V|$

- What is $D^{-1}A$?
  - Recall that the inverse of diagonal matrix is a diagonal matrix containing the reciprocal of the elements in the diagonal
- We need to find $x$
  
  \[
  Ix = cD^{-1}Ax + (1 - c)y
  \]
  \[
  Ix - cD^{-1}Ax = (1 - c)y
  \]
  \[
  (I - cD^{-1}A)x = (1 - c)y
  \]
  \[
  x = (I - cD^{-1}A)^{-1}(1 - c)y
  \]
Another way of looking at the RW

\[ x = cD^{-1}Ax + (1 - c)y \]

- Suppose the surfer starts from the beginning, it will choose one node at random among \( y \)
- In one step he will either choose, with probability \((1-c)\) another starting node or with probability \(c\) to move to one neighbor. Why?
- Think about the process without restart, setting \( W = D^{-1} A \).

\[
\begin{align*}
  x^{(1)} &= Wx^{(0)} \\
  x^{(2)} &= WX^{(1)} = WWx^{(0)} = W^2x^{(0)} \\
  & \vdots \\
  x^{(n)} &= W^nx^{(0)}
\end{align*}
\]

That means that \( W_{i,j}^n \) contains the probability of reaching \( j \) starting from \( i \) in \( n \) steps.
Semi-Supervised Learning (SSL)

**Idea**
If you have few labeled nodes exploit the structure and the homophily (similarity) between nodes

- graph-based
- few labeled nodes
- edges: similarity between nodes
- Inference: exploit neighborhood information

Ji, M., Sun, Y., Danilevsky, M., Han, J. and Gao, J., 2010. *Graph regularized transductive classification on heterogeneous information networks*. ECML/PKDD.
SSL Equation

\[ [I + \alpha (D - A)]x = y \]

- **Laplacian matrix**
- **Homophily strength of neighbors** ~ ”stiffness of spring”
- **Graph structure**
- **Final labels**
- **Known labels**

\[
\begin{pmatrix}
1 & 1 & 1 \\
\end{pmatrix}
\begin{pmatrix}
d1 \\
d2 \\
d3
\end{pmatrix}
\begin{pmatrix}
0 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
? \\
? \\
?
\end{pmatrix}
\begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix}
\]
SSL Equation - What does it compute?

\[ [I + a(D - A)]x = y \]

Hard? Let's unroll it!

\[ x = -a(D - A)x + y \]
\[ = aAx - aDx + y \]

\[ x_i = a \sum_{j=1}^{n} A_{ij}x_j + (y_i - aD_{ii}x_i) \]

Sum the labels from the neighbors

Difference between the learned label in the node and the input label
When is $\text{RWR} = \text{SSL}$?

\[
[I - cD^{-1}A]x = (1 - c)y \quad [I + a(D - A)]x = y
\]

\[
(1 - c)[I - cD^{-1}A]^{-1} = [I + a(D - A)]^{-1}
\]

\[
\left[\frac{1}{1 - c} I - \frac{c}{1 - c} D^{-1} A\right]^{-1} = [I + a(D - A)]^{-1}
\]

\[
\frac{1}{1 - c} I - \frac{c}{1 - c} D^{-1} A = I + a(D - A)
\]

\[
\frac{1}{1 - c} I - \frac{c}{1 - c} D^{-1} A = a(D - A)
\]

We assume the graph is d-regular*.

*all nodes have degree $d$
Belief Propagation

- Iterative message-based method
- “Propagation matrix”:
  - Homophily

<table>
<thead>
<tr>
<th>Class of Sender</th>
<th>Class of Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

... criterion fulfilled

- Pearl, J., 1982. Reverend Bayes on inference engines: A distributed hierarchical approach. In AAAI.
Belief Propagation

- Iterative message-based method
- “Propagation matrix”:
  - Homophily
  - Heterophily

<table>
<thead>
<tr>
<th>class of sender</th>
<th>class of receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homophily</td>
<td>0.9 0.1</td>
</tr>
<tr>
<td></td>
<td>0.1 0.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>class of sender</th>
<th>class of receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterophily</td>
<td>0.3 0.7</td>
</tr>
<tr>
<td></td>
<td>0.9 0.1</td>
</tr>
</tbody>
</table>
Belief Propagation Equations

message(\(i \rightarrow j\)) \approx \text{belief}(i) \times \text{homophily strength}

\[
\begin{array}{cc}
0.9 & 0.1 \\
0.2 & 0.8 \\
\end{array}
\]
Belief Propagation Equations

\[ b_i(x_i) \propto \phi_i(x_i) \cdot \prod_{j \in N(i)} m_{ij}(x_i) \]

belief of \( i \)

prior belief

messages from neighbors
Fast Belief Propagation

Original [Yedidia+]:

Belief Propagation

\[ m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i) \cdot \psi_{ij}(x_i, x_j) \cdot \prod_{n \in N(i) \setminus j} m_{ni}(x_i) \]

\[ b_i(x_i) = \eta \cdot \phi_i(x_i) \cdot \prod_{j \in N(i)} m_{ij}(x_i) \]

non-linear

FaBP [Koutra+]:

Linearized BP

BP is approximated by

\[ [I + aD - c^T A] b_h = \phi_h \]

prior beliefs

Wait ... How?

\[
m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i) \cdot \psi_{ij}(x_i, x_j) \cdot \prod_{n \in N(i) \setminus j} m_{ni}(x_i)
\]

\[
b_i(x_i) = \eta \cdot \phi_i(x_i) \cdot \prod_{j \in N(i)} m_{ij}(x_i)
\]

\[
[I + aD - c'A] b_h = \phi_h
\]

\[
Ib_h + aDb_h - c'Ab_h = \phi_h
\]

\[
b_h = (aD - c'A)b_h + \phi_h
\]

\[
b_h(i) = aD_{ii}b_h(i) - c' \sum_j A_{ij}b_h(j) + \phi_h(i)
\]

Exchanged messages
## Qualitative Comparison

<table>
<thead>
<tr>
<th>GBA Method</th>
<th>Heterophily</th>
<th>Scalability</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWR</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SSL</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BP</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>FABP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## Correspondence of Methods

\[
\begin{align*}
\text{RWR} & \approx \text{SSL} & \approx \text{BP} \\
\text{Random Walk} & \quad \text{Semi-supervised} & \quad \text{Belief} \\
\text{with Restart} & \quad \text{Learning} & \quad \text{Propagation}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>Matrix</th>
<th>unknown</th>
<th>known</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWR</td>
<td>([I - c \ A D^{-1}])</td>
<td>(x)</td>
<td>((1-c)y)</td>
</tr>
<tr>
<td>SSL</td>
<td>([I + a (D - A)])</td>
<td>(x)</td>
<td>(y)</td>
</tr>
<tr>
<td>FABP</td>
<td>([I + a D - c'A])</td>
<td>(b_h)</td>
<td>(\phi_h)</td>
</tr>
</tbody>
</table>

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 \\
1 \\
1
\end{bmatrix}
\]
Any questions?
References

- Glen Jeh and Jennifer Widom. SimRank: a measure of structural-context similarity. SIGKDD, 2002
- Ji, M., Sun, Y., Danilevsky, M., Han, J. and Gao, J., 2010. Graph regularized transductive classification on heterogeneous information networks. ECML/PKDD.
- Pearl, J., 1982. Reverend Bayes on inference engines: A distributed hierarchical approach. In AAAI.
- Neville, J. and Jensen, D., 2000. Iterative classification in relational data. AAAI.