

Amazon Recommendation System

Dhwanit Agarwal, Natasa Dragovic

University of Texas at Austin

dhwanit@ices.utexas.edu, ndragovic@math.utexas.edu

December 4, 2017

Overview

- 1 Problem
- 2 Data available
- 3 Methods
- 4 Preliminary Results
- 5 To do

Motivation

- Its hard for a person to choose from all the products on the internet, so it is helpful to have tailored options
- Companies want to increase their revenues by giving suggestions based on costumers tastes
- Using mathematical models to predict what people want to buy



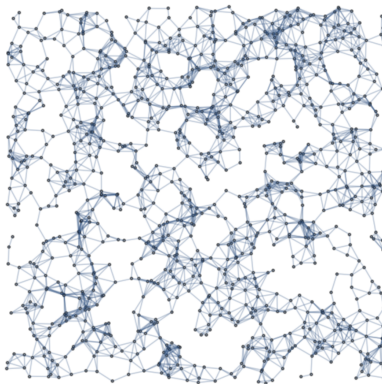
Figure: Books bought together [2]

Dataset

We are given a list of pairs of products purchased at some point in time. Each product is given a number and is a node on the graph. Two nodes are connected if they were bought together at some point.

Number of nodes=number of distinct products ≈ 400000

Number of edges ≈ 2 million



Problem statement

- Essentially, the problem at hand is a **link prediction problem**
- **Input:** Undirected sparse Graph $G = (V, E)$ V : set of vertices, E : set of edges
- **Output:** For a fixed node x output all the $score(x, y)$ for all $y \in V$, where $score(x, y)$ denotes the likelihood of connection of x to y
- **Requirements:** Reasonable precision. Feasible to run in real time and hence should run in a few seconds

Methods implemented [1]

- **Based on node neighborhoods:** Jaccard's coefficient, Common neighbors(CN), Adamic/Adar, Preferential Attachment(PA)
- **Based on ensemble of all paths:** Katz Index, Random Walk with Restarts (RWR)

Based on node neighborhoods: score calculation

Let $\Gamma(x)$ denotes the set of neighbors of x .

Common Neighbors (CN):

$$\text{score}(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

Calculates the number of common neighbors of x and y .

Jaccard's coefficient:

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

Fraction of common neighbors

Adamic/Adar:

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

Less connected neighbors have more weights

Preferential attachment (PA):

$$\text{score}(x, y) = |\Gamma(x)| \cdot |\Gamma(y)|$$

Uses the assumption that the likelihood that x has an additional edge is proportional to the number of links it already has.

Time complexity for calculating all these measures for fixed x is $O(|E|)$

Based on ensemble of all paths: score calculation

Katz _{β}

$$score(x, y) = \sum_{\ell=1}^{\infty} \beta^{\ell} |paths_{x,y} < \ell >|$$

where

$$paths_{x,y} < \ell > := \{\text{paths of length exactly } \ell \text{ from } x \text{ to } y\}$$

Measure that directly sums over a collection of paths, exponentially damped by the length to count short paths more heavily.

Fast Implementation

$$score(x, \cdot) = Se_x = \sum_{i=1}^n \beta A^i e_x$$

where β is smaller than the largest eigenvalue of A and n is a large number. e_x is the x^{th} canonical basis vector. A is the adjacency matrix of the graph which is sparse. Hence the **running time** is just of the order of matrix vector multiplication $O(|V|)$.

Based on ensemble of all paths: score calculation

Random walk with Restarts(RWR):

- Take a node x and create a random walk in the following way: with probability c it moves to a random neighbor and returns to x with probability $1 - c$.
- q_{xy} denotes the probability that this random walk locates at node y in the steady state
- $q_x = (1 - c)(I - cP^T)^{-1}e_x$
- P is a transition matrix with $P_{xy} = \frac{1}{|\Gamma(x)|}$ if x and y are connected, and $P_{xy} = 0$ otherwise. Sparse but non symmetric.
- Score is defined as $score(x, y) = q_{xy} + q_{yx}$.
- **Fast implementation:** Calculating the $score(x, y)$ for all y is not feasible in real time, since each of y requires a linear system to be solved using GMRES. Using the prediction from other methods, we reduce the sample of y and then apply this method.

Matrix Forest Index

Matrix Forest Index defined as

$$S = (I + D - A)^{-1}$$

A is the **adjacency matrix**. $D_{xy} = \delta_{xy}|\Gamma(x)|$, S output is the matrix of similarity score. Similarity between x and y can be understood as the ratio of the number of spanning rooted forests of the network.

Fast Implementation Instead of calculating S , just calculate Se_x . Time complexity is of **conjugate gradient** solved on sparse symmetric $(I+D-A)$. Typically the CG converges in two or three iterations, 10^{-3} accuracy.

Results

- **Precision** = Number of correct predictions/ Total number of predictions
- **Testing the algorithms:** We remove a certain number of edges randomly. Then we run different algorithms that predicts some number of edges. Finally we calculate how precise each algorithm is and its running time

Method	Katz	RWR	Jaccard	CN	PA	Adamic	MFI
AvgPrec	0.3	0.26	0.48	0.41	0.0	0.41	0.24
MaxPrec	1	1	1	1	0.25	1	0.75
MinPrec	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Time(sec)	0.2s	8s	4s	3s	1.5s	2s	0.2s

Further steps

- Implement the Maximum Likelihood Method
- Exploring a way to combine these metrics together to get better predictions
- Try expensive matrix completion methods

Maximum Likelihood Method: Hierarchical Structure Model

- Some real networks have hierarchy and so can be structured into groups of groups that could help classify and find missing links
- Drawback: algorithm runs at least $O(|V|^2)$ so we can only apply for small subset of nodes

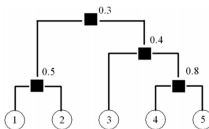


Fig. 2. Illustration of a dendrogram of a network with 5 nodes. Accordingly, the connecting probability of nodes 1 and 2 is 0.5, of nodes 1 and 3 is 0.3, of nodes 3 and 4 is 0.4.

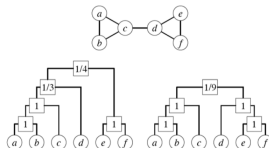





Fig. 3. The likelihood of two possible dendrograms for an example network consisting of 6 nodes. The interval nodes are labeled with the maximum likelihood probability obtained by Eq. (32). The likelihoods are $\mathcal{L}(D_1) \approx 0.00165$ (left dendrogram) and $\mathcal{L}(D_2) \approx 0.0433$ (right dendrogram). Copyright is held by Nature Publishing Group.

References

-  Linyuan Lu, Tao Zhou *Link predictions in complex networks: A survey* Physica A 390, (2011)
-  <https://carlispina.files.wordpress.com/2012/03/yasiv-for-amazon1.png>
-  David Liben-Nowel, Jon Kleinberg *The link prediction problem for social networks* Proceedings of the 12th annual ACM International Conference on Information and Knowledge Management(2004)

Thank you!