Lecture 27: Learning from relational data

STATS 202: Data mining and analysis

Sergio Bacallado

December 4, 2013

Announcements

- Kaggle deadline is this Wednesday (Dec 4) at 4pm. If you haven't already, make a submission to earn credit.
- In order to earn credit for the Kaggle competition, submit "Homework 8" on the website by this Friday. It will ask for your user name, your team's name, and a brief description of what you did.
- My office hours this week have been rescheduled to Friday afternoon (see website).
- You will get Homeworks 6 and 7 back by the end of the week.

Announcements

Final exam:

- Monday, December 9, 8:30-11:30 am at Cubberley Auditorium.
- SCPD students: The exam will be sent out early Monday morning, and you must return it by Tuesday December 10th at noon PST. Local students may take the exam at Stanford with the rest of the class.
- Closed book, closed notes. No calculators or computers. Equation sheet provided.
- Practice problems on the website.
- At least one problem in the exam will be from the practice problems.

Announcements

Course evaluations are available on Axess. If you complete them by Dec 16, you will be able to see your grades as soon as we post them. If you don't complete the evaluations, you will have to wait until Dec 19.

Relational data

The observations have the form of a graph.

Examples.

- Links between websites.
- Relationships between accounts in social networks.
- Transmission networks for contagious diseases.

The links can be **directed** or **undirected**.

There can be different types of link (friend, follower, followed).

We can observe the graph in time (how do social networks grow?).

Each vertex can have additional features or **metadata**.

PageRank algorithm

- Invented by Sergei Brin and Larry Page of Google.
- Uses a graph of links between websites to rank websites by “importance”.
- **Motivation:**
  - Consider the problem of searching the web using the query "birth control".
  - There are millions of pages containing the term.
  - Analyzing the content of each website semantically to infer which one is more likely to satisfy the user is very expensive.
  - We need a way to rank websites, to filter out all those that are rarely visited. This information is given by links.

PageRank algorithm

Consider a hypothetical **surfer** who is jumping from website to website by clicking on random links. Intuitively, the websites that are visited more frequently can be considered more important in the network of links.

Will the surfer visit every website eventually? No. It is possible to get stuck in a website with no outgoing links, or to be stuck in a loop between two websites, for example.

To avoid this problem, we modify the random walk, such that at every step, with probability $1 - q$, we pick a website at random, and with probability q we go through one of the links in the current website at random.

PageRank algorithm

- The **surfer**'s random walk is a Markov chain on the set of websites.
- It is a fact that the frequency with which the surfer visits any website converges to some limit.
- The PageRank of a website is this limiting frequency.

PageRank algorithm

Let P_{ij} be the probability of jumping from website i to website j, then

$$P_{ij} = (1-q)\frac{1}{n} + q\left[\frac{\text{# of links from } i \text{ to } j}{\text{# of links out of } i}\right]$$

The limiting frequency of website j, π_j, must satisfy

$$\pi_j = \sum_{i=1}^{n} \pi_i P_{ij}$$

or in matrix notation $\pi = \pi P$. That is, π is an eigenvector of the transition probability matrix P with eigenvalue 1.

Finding the limiting frequencies π

In principle, finding the limiting frequencies could require solving the eigendecomposition of a matrix P which is $n \times n$, and this has a complexity which grows as n^3.

However, it is possible to compute π by starting with the approximation $\pi^{(0)} = (1/n, \ldots, 1/n)$, and iterating:

$$\pi^{(t)} = \pi^{(t-1)} P.$$

The number of iterations necessary for convergence is typically small.

The matrix-vector multiplication in each iteration can be sped up using sparse matrix techniques.

How can PageRank be used in web search?

One idea:

1. Find all websites that contain all query terms.
2. Display them in order of their PageRank.

A more likely approach:

1. Use PageRank to select the 10,000 most important pages which contain the query terms.
2. Rank these 10,000 pages by analyzing their content, integrating information about the user, etc.

How can PageRank be used in web search?

One idea:

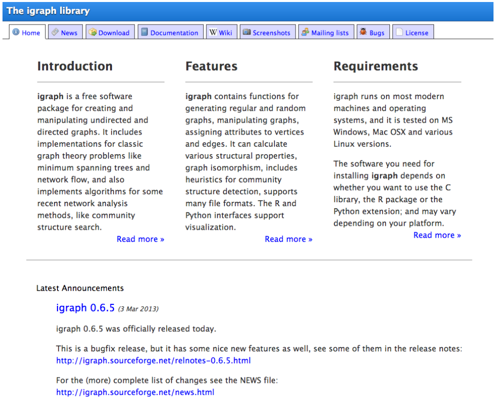
1. Find all websites that contain all query terms.
2. Display them in order of their PageRank.

A more likely approach:

1. Use PageRank to select the 10,000 most important pages which contain the query terms.
2. Rank these 10,000 pages by analyzing their content, integrating information about the user, etc.

PageRank in R

The package igraph in R has a function page.rank, among many other utilities for working with graphs.



Introduction

igraph is a free software package for creating and manipulating undirected and directed graphs. It includes implementations for classic graph theory problems like minimum spanning trees and network flow, and also implements algorithms for some recent network analysis methods, like community structure search.

Read more »

Features

igraph contains functions for generating regular and random graphs, manipulating graphs, assigning attributes to vertices and edges. It can calculate various structural properties, graph isomorphism, includes heuristics for community structure detection, supports many file formats. The R and Python interfaces support visualization.

Read more »

Requirements

igraph runs on most modern machines and operating systems, and it is tested on MS Windows, Mac OSX and various Linux versions.

The software you need for installing **igraph** depends on whether you want to use the C library, the R package or the Python extension; and may vary depending on your platform.

Read more »

Latest Announcements

igraph 0.6.5 *(3 Mar 2013)*

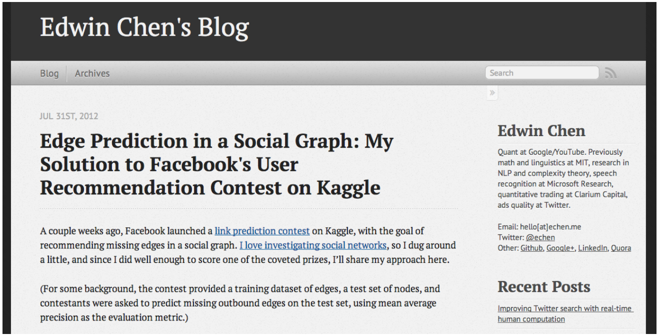
igraph 0.6.5 was officially released today.

This is a bugfix release, but it has some nice new features as well, see some of them in the release notes: http://igraph.sourceforge.net/relnotes-0.6.5.html

For the (more) complete list of changes see the NEWS file: http://igraph.sourceforge.net/news.html

Edge prediction in Facebook's social graph

For the rest of this lecture, we will review Edwin Chen's solution to a Kaggle contest organized by Facebook.


Edwin Chen's Blog

JUL 31ST, 2012

Edge Prediction in a Social Graph: My Solution to Facebook's User Recommendation Contest on Kaggle

A couple weeks ago, Facebook launched a link prediction contest on Kaggle, with the goal of recommending missing edges in a social graph. I love investigating social networks, so I dug around a little, and since I did well enough to score one of the coveted prizes, I'll share my approach here.

(For some background, the contest provided a training dataset of edges, a test set of nodes, and contestants were asked to predict missing outbound edges on the test set, using mean average precision as the evaluation metric.)

Edwin Chen

Quant at Google/YouTube. Previously math and linguistics at MIT, research in NLP and complexity theory, speech recognition at Microsoft Research, quantitative trading at Clarium Capital, ads quality at Twitter.

Email: hello[at]echen.me
Twitter: @echen
Other: Github, Google+, LinkedIn, Quora

Recent Posts

Improving Twitter search with real-time human computation

Link to blog post.