Advanced Data Mining

M2177.003000, Seoul National University, Fall 2019

These are lecture notes for "Advanced Data Mining" taught by U Kang at Seoul National University during the fall of 2019. These notes are not official and for personal use, and thus have not been proofread by the instructor for the course and may be of limited use to you. These notes live in my lecture notes respository at

https://github.com/qmdnls/lecture-notes.

If you find any errors, please open a bug report describing the error, and label it with the course identifier, or open a pull request so I can correct it.

Contents

Syllabus

Instructor	Kang U, https://datalab.snu.ac.kr/~ukang/
Lecture	MW 3:30–4:45 PM, 301-101
Textbook	None
Midterm	Wednesday, October 23, 2019
Final	TBA

Data mining attracted much interests as an essential tool for big data analysis. Especially, designing and implementing advanced data mining algorithms and analysis platforms play crucial roles in extracting executable knowledges from big data. This course covers advanced data mining techniques, algorithms, and core platforms for big data analysis. This course also covers the techniques to effectively analyze very large data and high-speed data. A very important aspect of this course is the course project. Students will pick an interesting data mining project, and do data mining researches on the topic. At the end of the course, students will learn how to do interesting researches in data mining, and how to write good papers.

Your final grade for the course will be determined by

$$10\% \text{ attendance} + 20\% \text{ homework} + 20\% \text{ midterm} + 20\% \text{ final} + 30\% \text{ project.}$$

1 Graphs

Starting with a basic recap of graphs. I will not go into much detail here as most of these things should be familiar from previous lectures. Lecture did not have any formal definitions so I will leave them out here as well.

> **Definition** (Graph)**.** A graph $G = (V, E)$ is defined by a set of vertices V and a set of edges $E = V \times V$ between these vertices.

Graphs are a way of specifying relationships among a set of items. These relationships are expressed in the forms of edges between vertices (or nodes).

1.1 *Types of graphs and basic graph terms*

There are different types of graphs:

- directed graph
- undirected graph
- weighted graph
- unweighted graph
- simple and attributed graph

Some examples of real-world graphs include:

- social networks
- citation networks
- world wide web
- protein interactions
- document networks
- computer networks (ArpaNet)

Let's define some basic graph terms: Two vertices are *adjacent* if they share a common edge. Two adjacent vertices are *neighbors*. An edge is *incident* with another edge if they share a vertex. An edge is incident with two vertices. The *degree* of a node is the number of its neighbors. In a directed graph we define the *in-degree* and the *out-degree* based on the number of neighbors with incoming or outgoing edges.

1.2 *Small world phenomenon*

The *small world phenomenon* describes the results of a series of experiments that have shown that social networks in the real world have a low average path length. Milgram's experiment for instance has shown that there was an average of six degrees of separation in the US. Another popular number: Kevin Bacon number. Describes the distance from Kevin Bacon in the actor-actor collaboration graph. The average Kevin Bacon number of actors is 2.9 which is surprisingly low.

1.3 *Densification power law*

Describing the growth of real-world graphs has important applications in network simulation and network prediction. But how do these graphs evolve over time? Conventional wisdom was that the number of edges grows linearly with the number of of nodes, so the average degree stays constant, while the diameter slowly grows. However new findings have shown that networks become denser over time, this is called the *densification power law*. Furthermore, the diameter appears to be shrinking as a result of this densification.

Theorem 1.1 (Densification power law)**.** *For the number of edges $E(t)$ and the number of nodes $N(t)$ at time t, we have the following relation: $E(t) \propto N(t)^a$ with $a \in [1,2]$ where $a = 1$ yields the traditionally assumed constant out-degree (linear growth), while $a = 2$ gives quadratic growth.*

In real world graphs this densification exponent differs and depends on the type of underlying real-world relationships but is typically $a > 1$ which means the average degree is increasing. Further analysis of real-world graphs shows that this densification leads to shorter distances between nodes and thus the diameter shrinks over time.

1.4 *Forest fire model*

To describe these properties of real-world graphs we introduce a new graph generation model that follows the densification power law and that has a shrinking diameter over time. In the *Forest Fire* model a node randomly chooses an "ambassador" when it arrives. It then starts burning adjacent nodes with probability p and adds a link to burned nodes. This "fire" then spreads recursively. This is very similar to the process of friendship networks in the real world as people will often become friends with their friend's friends. This forest fire model generates graphs that densify and have a shrinking diameter.

1.5 *Erdős–Rényi random graph model and preferential attachment*

The *Erdős–Rényi random graph model* (ER model) is a graph generation model. Given the desired number of vertices $n \in \mathbb{N}$ and a probability $p \in [0,1]$ we create n vertices and include an edge between any two vertices with probability p. The degree distribution in this model follows a Poission distribution which means it can be described by $P(k) \sim \frac{e^{-\lambda}\lambda^k}{k!}$ where k is the degree.

However, real graphs show different behavior. We see that the degree distribution typically follows a power law: $P(k) \sim k^{-r}$. To better describe this real-world graph behavior we introduce the *preferential attachment* model. In this model networks continuously expand by the addition of new vertices. These new vertices attach preferentially to other vertices that are already well connected. Specifically, we define the probability that a new vertex is connected to an existing vertex i by

$$P_i = \frac{d(i)}{\sum_j k_k}$$

where $d(i)$ is the degree of vertex i. This then leads to a power law distribution in the resulting graph.

1.6 Graph robustness

How robust are real-world graphs? What happens when we remove nodes from them? We find that preferential attachment is very robust against random failure (low probability of high-degree nodes failing), however it is vulnerable to attacks (attack only high-degree hub nodes). In general, scale-free networks have a high-degree of tolerance against random failures that exponential networks do not have. This observation could explain why many complex systems in the real world manage to function and have high error tolerance.

1.7 Community connection model

A *rebel* is a vertex in a graph that does not belong to its largest connected component (GCC). The rebel probability describes a probability of a given node being a rebel. Models introduced thus far (preferential attachment, forest fire) only have a single connected component. To generate graphs that have many connected components we introduce the *Community connection model*. The model has two parameters: p_{host} and p_{step}. When a new node joins the network in this model, it connects to a new host with p_{host} and then begins a random walk with p_{step} until there are no more "steps". This is repeated for all hosts. We find that this model describes the rebel probability found in real-world graphs very well.

1.8 Power laws in the internet

A few questions that motivate us: What does the internet look like? Are there any topological properties of the internet that remain constant over time? How will it develop in the future? How can we generate internet-like graphs?

There are three main power laws in the internet.

Theorem 1.2 (Power law 1 (rank exponent))**.** *The out-degree d_v of a node v is proportional to the rank of a node r_v to the power of a constant $\mathcal{R}$:*

$$d_v \propto r_v^{\mathcal{R}}$$

Theorem 1.3 (Power law 2 (out-degree exponent))**.** *The frequency f_d of an out-degree d is proportional to the out-degree to the power of a constant $\mathcal{O}$:*

$$f_d \propto d^{\mathcal{O}}$$

Theorem 1.4 (Power law 3 (eigen exponent))**.** *The eigenvalues λ_i of a graph are proportional to the order i to the power of a constant $\mathcal{E}$:*

$$\lambda_i \propto i^{\mathcal{E}}$$

1.9 Generation of power law distributions

Power laws can be used to describe graphs more accurately: the average often falsely implies a uniform or Gaussian distribution when in reality distributions can be skewed. For example, 85% of the nodes in the Int12-98 dataset have an out-degree that is smaller than the average. Besides graph descriptions, power laws can be useful for graph generation, for prediction and extrapolation and to evaluate protocol performance (in the internet for example). Finding power laws: power law distributions are linear in a log-log scale. Thus, we can discover them by plotting data in log-log (but beware: not all distributions that look like power laws at first glance follow power laws).

We can describe a power law distribution in the following way:

$$p(x) = Cx^{-a} \; \forall \; x \geq x_{\text{min}}$$

for a constant C and an exponent a which gives us a way of directly computing $p(x)$ for any x. The exponent a can often be estimated from data like so:

$$a = 1 + n \left(\sum_{i=1}^{n} ln \left(\frac{x_i}{x_{\text{min}}} \right) \right)^{-1}$$

There are three types of power laws: PDF (*probability distribution function*, also frequency-count plot), *Zipf plot* (rank-frequency) and *complementary cumulative distribution function* (CCDF, also true for NCDF $= 1 -$ CDF). We find that, if any one of the three follows a power law, so do the other two.

In short we can say that any distribution which follows a "the rich get richer" principle, typically follows a power law. Examples of generative mechanisms for power law distributions are:

- Chinese restaurant process (Yule distribution)
- Combination of exponentials
- Inverses of quantities
- Random walks

Chinese restaurant process. One such way of generating power law distributions is given by the *Chinese Restaurant Process* (also Yule distribution): any newcomer to a restaurant sits down at an existing table. In his choice the newcomer prefers large groups. She only chooses to start a new table with probability $\frac{1}{m}$. This is also referred to as a *Yule process.*

Combination of exponentials. Combination of exponentials is a very simple technique to generate power law distributions. One example is radioactive decay with half-life $-a$ and $p(y) = e^{ay}$. Another example is Russian roulette where the participants' capital x increases every time they survive with $x \sim e^{by}$. Then their final capital follows a power law distribution. Lastly, imagine a monkey typing on a typewriter. If each letter is equiprobable and the space bar has probability q_s then the frequency of the x-th most frequent word is x^{-a}, i.e. follows a power law.

Random walks. For random walks the number of steps required to arrive at the same position, the *inter-arrival time*, follows a power law.

Random multiplication. Starting with C dollars and a random interest rate $s(t)$ for each year t, we get $C(t) = C(t-1)(1+s(t))$. We have $\log C(t) = \log C + \log.. + \log..$ which is a Gaussian distribution and thus $C(t) = \exp(\text{Gaussian})$ which is a lognormal distribution. The lognormal distribution looks like a power law in its tail distribution (but strictly is not a power law).

Fragmentation. Starting with a stick of length 1, break it at a random point $0 \leq x \leq 1$. Then recursively break the resulting pieces at random. The resulting length distribution is lognormal (analogous to random multiplication).

1.10 Radius plots to describe large networks

Real-world networks often have a giant connected component (GCC). This component is seen to be growing over the years (example: Wikipedia, LinkedIn) while the tail slope remains constant in the distribution (see slides for graphs). But what does the structure of these large networks look like? Which nodes are the most central and how does the structure change over time?

To better describe the structure of a graph we can use its radius: we plot the count of nodes over the radius. Different structures yield different *radius plots*: in a clique all nodes have radius 1, in a star one node has radius 1 and all other nodes have radius 2, similarly a chain and a graph with a near-bipartite core have distinct radius plots (see slides). As a measure of node centrality we can use its radius: to choose a node to advertise in a graph, simply choose the node with the smallest radius for fastest propagation in the network. We can study the change of the structure of networks over time by looking at the radius plot over time.

In real graphs, the radius plot is multi-modal (again, see slides). The LinkedIn and US patent radius plots are bi-modal. This is due to the fact that there is outsiders which have a very small radius, then there is the GCC in which some very-connected nodes have a smaller radius but almost all nodes have a slightly larger radius and then there is the whiskers, which are further outside the core, have a longer path to the core and thus have a larger radius. In other datasets (Yahoo Web) with a multi-modal radius plot there is more than core. Over time, real-world radius plots expand and contract (*expansion-contraction*).

Also consider *radius-degree plots*: describes the degree of nodes of a given radius. We see that high-degree hubs have a lower radius while lower-degree nodes mostly have a larger radius.

2 Spectral analysis