Networks

David Puelz

Outline

Networks and data from networks

Summarizing networks

Generating networks – association rules

Networks

Network data is everywhere. In most cases network relationships are assumed to not exist, but they are actually important and meaningful.

Examples:

- Spatial networks, countries and trade networks
- City streets / NYC subway
- Social media: Meta (Facebook), Instagram, Linkedin
- Spotify
- Classrooms / households

Networks to graphs

Graphical models provide a language for networks:

- 0/1 connections between people/sites/covariates

Think of graphs like a binary version of correlation

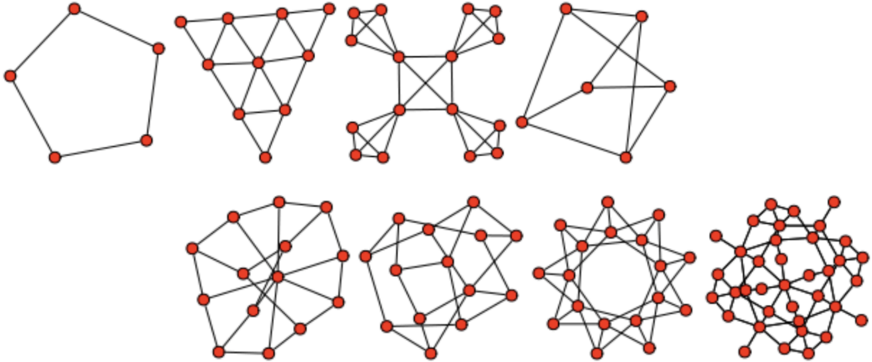
Graph Structure (how to describe a network?):

- Summarization: nodes and edges, direction
- Measuring connectivity and betweenness

Association and networks

- Market Basket Analysis

What comprises a network?



The network has **nodes** (vertices), such as a website or worker, and **edges** are the (directed or undirected) links between nodes.

Network data is connected

A network consists of variables and connections between them. A connection is discrete: it's either there or it's not.

Data living on a network:

- → Word usage in text and language (what words follow?)
- → Organization charts and employment (who's boss?)
- → Business credit, supply, and competition networks
- → Genes
- → Everything on the internet!

Sometimes the network is given, other times we just get a glimpse of traffic on the network.

Network structure

Start with a given network: you see all connections.

We'll reduce dimension + summarize important properties.
In particular, we'll focus on measures of network connectivity.

Each node has connectivity statistics

Degree: How many other nodes are you connected to?
Betweenness: How many node-to-node paths go through you?

You can also make a lot of cool illustrations for graphs. These tend to be more pretty than informative, but that doesn't mean they aren't useful.

Networks and R

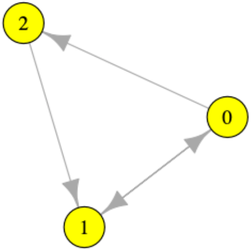
[igraph](#) is a great toolbox for visualizing and summarizing graphs. It has front-ends for `R` and `Python`.

Unlike most `R` packages, `igraph` is well documented. Type `help(igraph)` to get started.

For most applications, you'll read graphs from an edgelist:

```
0 1
0 2
1 0
2 1
```

$\Rightarrow$



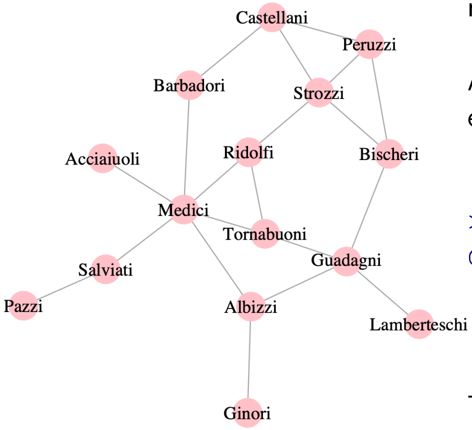
Example: Marriage and power

Early Renaissance Florence was ruled by an oligarchy of powerful families.

By the 15th century, the Medicis emerged supreme, & Medici Bank became the largest in Europe.

Political ties were established via marriage. How did Medici win?

Marriage in Florence: 1250-1450



Network links can be used to measure "social capital"

A node's degree is its number of edges

```
> sort(degree(marriage))
Ginori ... Strozzi Medici
     1           4      6
```

The Medicis are connected!

Betweenness – a deeper measure of network structure

An alternative to degree, betweenness measures the proportion of shortest paths containing a given node

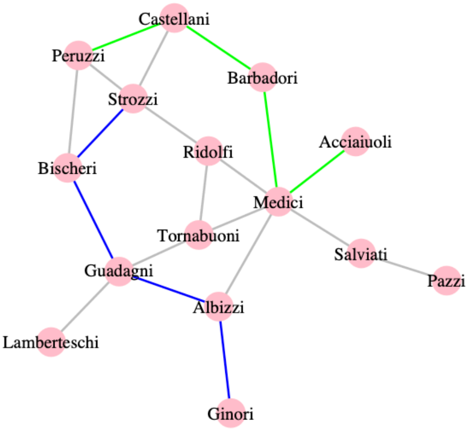
Shortest path: fewest steps from i to j (direction matters)

Say $s_k(i,j)$ is the proportion of shortest paths from i to j containing node k.

$$\text{betweenness}(k) = \sum_{i,j:i\neq j,k\notin\{i,j\}} s_k(i,j)$$

Intuitively, this measure how much influence a node has over connections between others!

Betweenness versus degree



Medicis have the highest degree, but only by a factor of 1.5 over the Strozzis.

```
> sort(betweenness(marriage))
 Ginori  ...  Strozzi  Medici
    0.0           9.3    47.5
```

But their betweenness is 5 times higher!

Collaborative filtering: Building a network from data

A common question in data mining: What do one person's choices say about anothers?

As Amazon says: "people who buy this book also bought..."

These types of tasks are referred to as **collaborative filtering**: using *shared choices* to predict preferences.

Collaborative filtering: Building a network from data

It's a big field, with many tools:

- logistic regression of each product on to all other choices
- principal components analysis: Underlying taste factors.

But as an easy start, there are good fast algorithms for discovering low dimensional association rules. Foreshadow, we will build networks with these association rules.

Collaborative filtering: Association rules

Consider two binary variables: x_a & x_b.

If $x_b = 1$ more often when $x_a = 1$, then we say $x_a \implies x_b$ is an **association rule**.

Example: When you buy chips, you need beer to wash them down

Suppose that beer is purchased 10% of the time in general, but 50% of the time when the consumer grabs chips.

$\rightarrow$ The support for beer is 10%
$\rightarrow$ The confidence for beer is 50%
$\rightarrow$ The lift is 5

Collaborative filtering: Association rules

Consider two binary variables: x_a & x_b.

If $x_b = 1$ more often when $x_a = 1$, then we say $x_a \implies x_b$ is an **association rule**.

Example: When you buy chips, you need beer to wash them down

Suppose that beer is purchased 10% of the time in general, but 50% of the time when the consumer grabs chips.

$\rightarrow$ The support for beer is 10% (total probability)
$\rightarrow$ The confidence for beer is 50% (conditional probability)
$\rightarrow$ The lift is 5 (ratio)

Running with this example

Market basket analysis – use purchase coincidence to build association rules

Our example basket: LHS (chips) $\implies$ RHS (beer), where LHS is called the antecedent, and the RHS is called the consequent.

Every event has support: the proportion of times it occurred. This leads to two measures of association rule strength:

- confidence: $\frac{\text{supp(LHS and RHS)}}{\text{supp(LHS)}}$ ... probability of RHS given LHS
- lift: $\frac{\text{supp(LHS and RHS)}}{\text{supp(LHS)supp(RHS)}}$ ... increase in probability of RHS given LHS

Interpretation? Remember Bayes rule! $P(\text{RHS} \mid \text{LHS}) = P(\text{RHS}, \text{LHS})/P(\text{LHS})$

Association, support, and lift

Generally, association rules with high lift are most useful because they tell you something you don't already know.

Low support does not preclude high confidence or high lift

- chips $\implies$ beer is high support, but low lift if everybody always buys beer!
- caviar $\implies$ vodka is low support, but high lift if people only buy vodka for their caviar parties

There's no deep theory around these rules, we often just scan the data for interesting relationships and go from there.

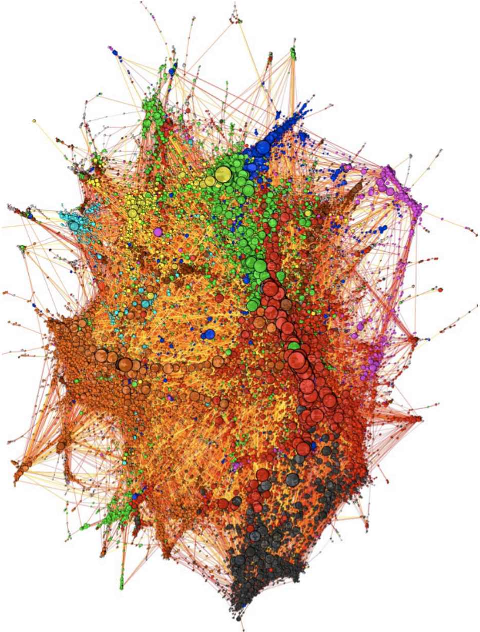
Association rules and R

To find confidence and lift, just count the # of times RHS and LHS happen, and how often they happen together.

$$\text{supp(event)} = \frac{\text{\# of times event occurs}}{\text{\# of observations}}$$

However, counting all combinations can take a long time! The `apriori` function in R is very useful for this task.

Streaming playlists and music preferences



Let's consider a large collection of playlists (like a shopping cart!)

metal, rock, pop, jazz, electronica, hip-hop reggae/ska, classical, folk/country/world.

This "network" shows artists sized by play count, with lines (edges) for shared users.

Association rules for musical taste

lhs		rhs	support	confidence	lift
t.i.	=>	kanye west	0.0104	0.5672	8.8544
pink floyd, the doors	=>	led zeppelin	0.0106	0.5387	6.8020
beyonce	=>	rihanna	0.0139	0.4686	10.8810
morrissey	=>	the smiths	0.0112	0.4655	8.8961
megadeth	=>	iron maiden	0.0132	0.4307	7.2677
jimi hendrix	=>	the doors	0.0120	0.3062	5.3170
nelly furtado	=>	madonna	0.0100	0.2750	5.0374
bright eyes	=>	the shins	0.0102	0.2698	5.4623
elliott smith	=>	modest mouse	0.0109	0.2679	5.1732
britney spears	=>	lady gaga	0.0120	0.2612	7.7292
ramones	=>	the clash	0.0104	0.2586	5.9052
franz ferdinand	=>	kaiser chiefs	0.0132	0.2224	7.1153

Example: Given a new user that listens to a lot of Morrissey, we're 46% positive that they'll also like the Smiths. This is 9 times higher than if we didn't know about Morrissey.

From association rules to networks

Graphs can be a useful way to summarize all sorts of data. We can define a network using any measure of connectivity.

For example, an association network:

- Say there's an edge between LHS and RHS if support and confidence are greater than some <u>thresholds</u>
- If we just look at any shared membership in a playlist, we get our monster graph from the beginning

Let's check out `playlists.R` to see this in action.