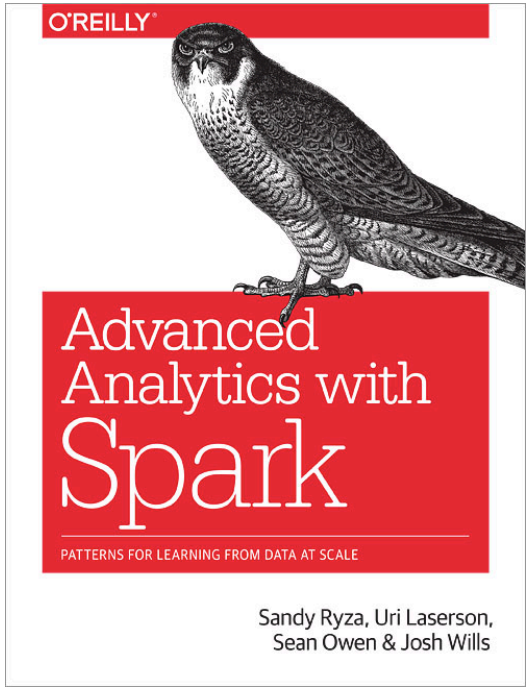
Analyzing Co-occurrence Networks with GraphX

CIT-652 Advanced Analytics

Sameh El-Ansary, PhD
Nile University



Beyond Basic Spark & MLLib

- First, we started by seeing the power of Spark:
  - General distribution of computations
  - As an alternative of ETL and SQL solutions at scale
- Second, we have seen so far the power of MLLib including:
  - Sparse Arrays and Matrices
  - Recommender Systems
  - Supervised Learning
  - Unsupervised Learning (in projects)
  - Text Analytics (in projects)
- Today, we see another point of strength in Spark, namely:
  - Graph analytics using the Graphx library

Network Science

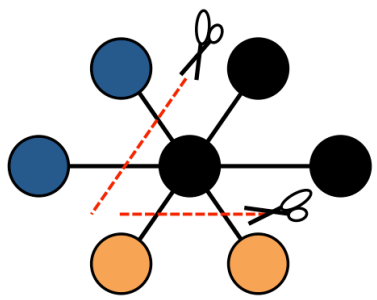
- In many fields, understanding *relationships* between entities is essential
- These entities could be: neurons, individuals, countries, etc.
- It is becoming an important skill of data scientists to work on graphs representing relationships
- Graph theory is a well-established theoretical field for analyzing graphs
- However, with the scale of data found nowadays, new tools are in need

Graph-centric companies

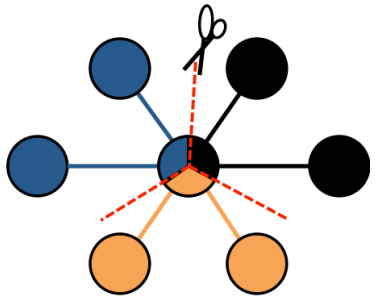
- Pagerank, the main search algorithm of Google is based on an ieterative graph algorithm
- Facebook, LinkedIn and all social media companies, operate on a large graph of friends and graphs are used in many services, like finding new friends, selecting relevant posts, etc..

GraphX on Spark, Why?

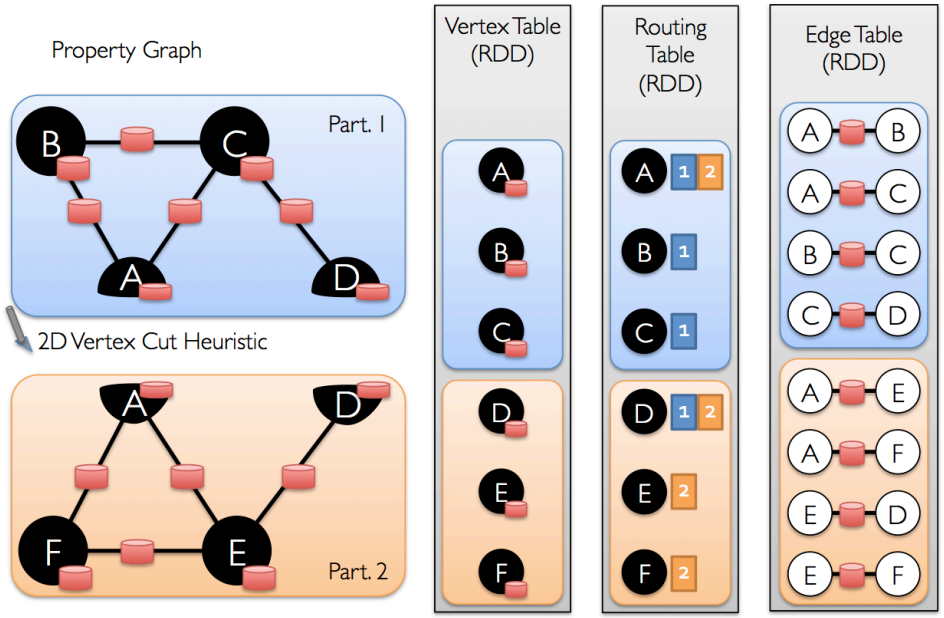
- A library that extends spark to support many graph-parallel algorithms.
- At it is the case in e.g. Sparse matrices, GraphX innovation includes novel ways of representing graphs in a distributed manner
- Internal implementation of graphs in GraphX is out of the scope of this course



Edge Cut



Vertex Cut



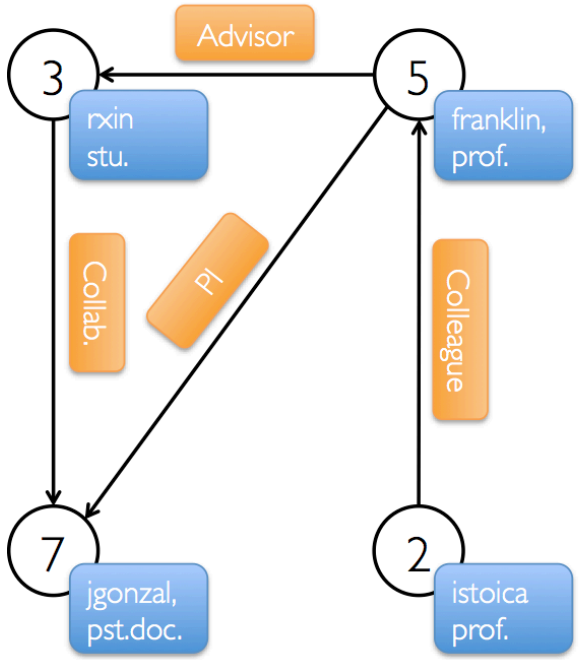
GraphX on Spark -RDDs

- GraphX is designed to help us analyze various kinds of networks using the language and tools of graph theory.
- GraphX builds on top of Spark, it inherits all of Spark's scalability properties, which means that it is capable of carrying out analyses on extremely large graphs that are distributed across multiple machines
- GraphX is based on two specialized RDD implementations that are optimized for graphs:
  - Vertex RDD
  - Edge RDD

VertexRDD & EdgeRDD

- The **`VertexRDD[VD]`**
  - Specialized implementation of **`RDD[(VertexId, VD)]`**
  - **`VertexID`** type is an instance of **`Long`** and is required for every vertex
  - **`VD`** can be any other type of data that is associated with the vertex, and is called the *vertex attribute*
- **`EdgeRDD[ED]`**
  - Specialized implementation of **`RDD[Edge[ED]]`**
  - Edge is a case class that contains two **`VertexId`** values and an *edge attribute* of type **`ED`**.
- Both the **`VertexRDD`** and the **`EdgeRDD`** have internal indices within each partition of the data that is designed to facilitate fast joins and attribute

Property Graph



Vertex Table

Id	Property (V)
3	(rxin, student)
7	(jgonzal, postdoc)
5	(franklin, professor)
2	(istoica, professor)

Edge Table

SrcId	DstId	Property (E)
3	7	Collaborator
5	3	Advisor
2	5	Colleague
5	7	PI

The MEDLINE Citation Index

- MEDLINE:
  - Medical Literature Analysis and Retrieval System Online
  - Database of academic papers published in journals covering the life sciences and medicine sicne 1879
  - More than 20 million articles
- MeSH
  - Medical Subject Headings
  - Set of semantic tags
  - Meaningful framework that can be used to explore relationships between documents
- Limited analysis preivously
- We will analyze this dataset Large-Scale Structure of a Network of Co-Occurring MeSH Terms based on the recent anlysis in:
  - “Statistical Analysis of Macroscopic Properties,” by Kastrin et al. (2014)
  - The analysis includes:
    - Topic co-coccurence
    - Connected graph components
    - Degree distribution
    - Clustering coefficient
    - Average path length

Outline of Analysis

- **Parsing XML Documents**
- **Basic stats:**
  - Topic popularity
  - Topic co-occurrence
- **Construct a co-occurrence graph**
- **Compute Graph stats:**
  - Connected Components
  - Degree Distribution
- Filtering Out Noisy Edges
- Small-world network analysis

Parsing XML

- We use a custom input format on fo xml
- Each record is an xml element and not a line

```
▼<MedlineCitationSet>
 ▼<MedlineCitation Owner="PIP" Status="MEDLINE">
    <PMID Version="1">12255379</PMID>
  ▶<DateCreated>...</DateCreated>
  ▶<DateCompleted>...</DateCompleted>
  ▶<DateRevised>...</DateRevised>
  ▼<Article PubModel="Print">
    ▶<Journal>...</Journal>
    ▼<ArticleTitle>
       Association of maternal and fetal factors with development of mental deficiency.
     </ArticleTitle>
    ▶<Pagination>...</Pagination>
    ▶<AuthorList CompleteYN="Y">...</AuthorList>
     <Language>eng</Language>
    ▶<PublicationTypeList>...</PublicationTypeList>
   </Article>
  ▶<MedlineJournalInfo>...</MedlineJournalInfo>
   <CitationSubset>J</CitationSubset>
  ▶<MeshHeadingList>...</MeshHeadingList>
   <OtherID Source="PIP">550018</OtherID>
   <OtherID Source="POP">00020417</OtherID>
  ▶<OtherAbstract Type="PIP" Language="eng">...</OtherAbstract>
  ▼<KeywordList Owner="PIP">
     <Keyword MajorTopicYN="N">Behavior</Keyword>
     <Keyword MajorTopicYN="Y">Comparative Studies</Keyword>
     <Keyword MajorTopicYN="N">Congenital Abnormalities</Keyword>
     <Keyword MajorTopicYN="N">Diseases</Keyword>
     <Keyword MajorTopicYN="N">Handicapped</Keyword>
     <Keyword MajorTopicYN="N">Intelligence</Keyword>
     <Keyword MajorTopicYN="Y">Maternal-fetal Exchange</Keyword>
     <Keyword MajorTopicYN="Y">Mental Retardation</Keyword>
     <Keyword MajorTopicYN="N">Personality</Keyword>
     <Keyword MajorTopicYN="N">Pregnancy</Keyword>
     <Keyword MajorTopicYN="Y">Pregnancy Complications</Keyword>
     <Keyword MajorTopicYN="N">Psychological Factors</Keyword>
     <Keyword MajorTopicYN="N">Reproduction</Keyword>
     <Keyword MajorTopicYN="N">Research Methodology</Keyword>
     <Keyword MajorTopicYN="N">Studies</Keyword>
   </KeywordList>
 </MedlineCitation>
</MedlineCitationSet>
```

Topic Popularity

- We count, the number of articles having a certain topic as a tag. **240,000 articles**
- The most frequently occurring major topics are, the most *general ones*, like “Research,” “Toxicology,” “Pharmacology,” and “Pathology.”
- Also includes references to various **patient populations**, like “Child,” “Infant,” “Rats,” or “Adolescent.”
- The most frequently occurring major topic only occurs in a small fraction of all the documents (5,591/240,000 ~ 2.3%), so we will probably get a **long tail popularity distribution**

```
(Research,5591)
(Child,2235)
(Infant,1388)
(Toxicology,1251)
(Pharmacology,1242)
(Rats,1067)
(Adolescent,1025)
(Surgical Procedures, Operative,1011)
(Pregnancy,996)
(Pathology,967)
```

DataSet actual number might slightly differ!!

Topic Co-Occurence

- The most frequently occurring co-occurrence pairs are also relatively uninteresting.
- Most of the top pairs, like ("Child," "Infant") and ("Rats," "Research"), are simply the product of two of the most frequently occurring individual topics.
- *There's nothing surprising or informative about the fact that these pairs exist in the data.*

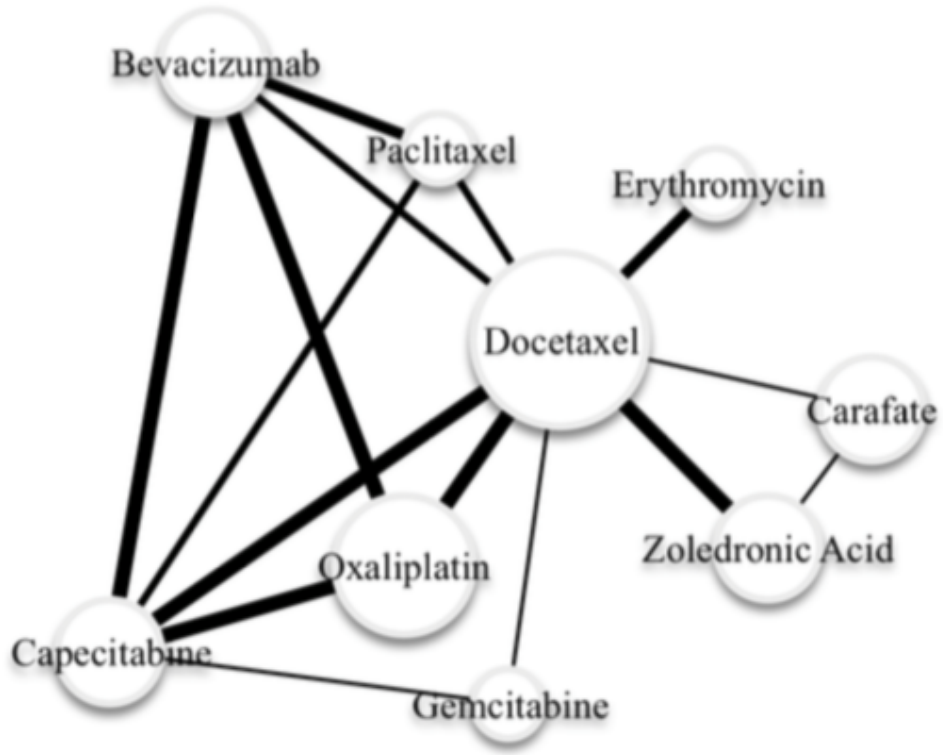
```
(List(Child, Infant),1097)
(List(Rats, Research),995)
(List(Pharmacology, Research),895)
(List(Rabbits, Research),581)
(List(Adolescent, Child),544)
(List(Mice, Research),505)
(List(Dogs, Research),469)
(List(Research, Toxicology),438)
(List(Biography as Topic, History),435)
(List(Metabolism, Research),414)
```

Outline of Analysis

- ~~**Parsing XML Documents**~~
- ~~**Basic stats:**~~
  - ~~Topic popularity~~
  - ~~Topic co-occurrence~~
- **Construct a co-occurrence graph**
- **Compute Graph stats:**
  - Connected Components
  - Degree Distribution
- Filtering Out Noisy Edges
- Small-world network analysis

Co-occurrence Network with GraphX

- Standard tools for summarizing data don't provide us much insight
- Raw counts, don't give us a feel for the overall structure of the relationships in the network
- We want to to think about the topic relationship as "Graph"



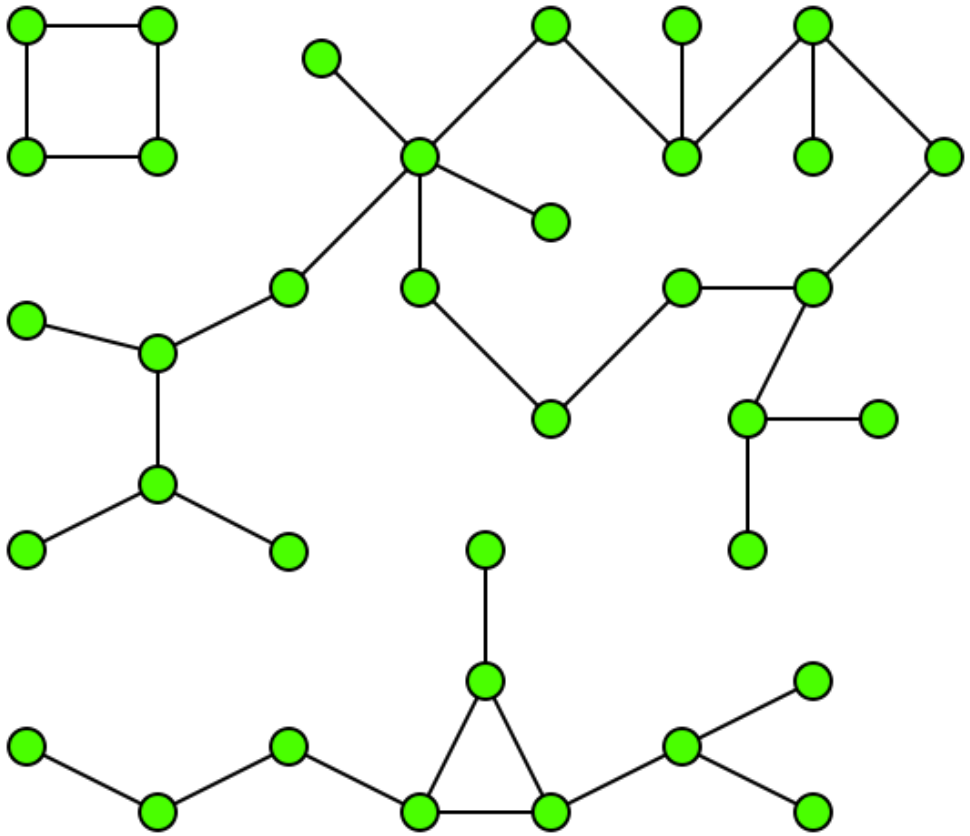
Constructing a GraphX graph

- Vertices:
  - Id: md5 hashes of topic string
  - Attribute: topic string
- Edges:
  - Attributes: count

```
val vertices = topics.map(topic => (hashId(topic), topic))
val edges = cooccurs.map(p => {
 val (topics, cnt) = p
 val ids = topics.map(hashId).sorted
 Edge(ids(0), ids(1), cnt)
})
val topicGraph = Graph(vertices, edges)
```

Graph Components

- A connected graph is one where we can reach any vertex from any other verted
- When a graph is disconnected, it will have multiple ”islands” as subgraphs, each one is called a components



Extracting Graph Components in Graphx

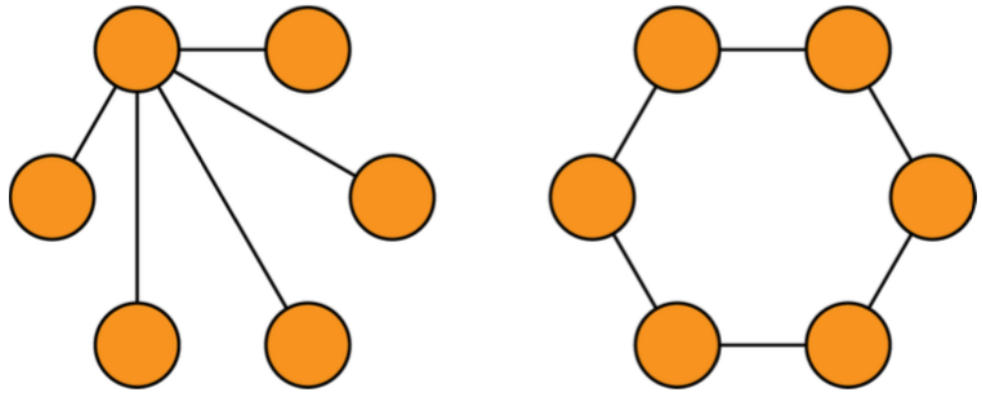
```
val connectedComponentGraph = topicGraph.connectedComponents()
```

- connectedComponents returns another graph where the list of vertices is the edge with lowest id in each component
- We can print the size of each component
- As we we can there a big huge component with most of the topics and then there are many tiny components
- Upon checking this further it seems from the data that this is a result of mislabelling

```
(-9225947734371556 29,11915)
(-6468702387578666337,4)
(-7038642868304457401,3)
(-7926343550108072887,3)
(-5914927920861094734,3)
(-4899133687675445365,3)
(-9022462685920786023,3)
(-7462290111155674971,3)
(-5504525564549659185,3)
(-7557628715678213859,3)
```

Degree Distribution

- The degree of a node is the number of edges associated with it
- Each vertex has a different degree, so we have a distribution of degrees
- Two different graphs with the same number of vertices can have a different degree distribution
- Left graph distribution:
  - 1 node with degree 5 (high)
  - 5 nodes with degree 1 (low)
- Right graph distribution:
  - 6 nodes with degree 2 (all the same)



- The distribution can be e.g. : **uniform**, which means all nodes are the same or it can be **skewed** meaning that some nodes have much higher degree

Digree Distribution

Highest Degree Topics

(Research,3753)
(Child,2364)
(Toxicology,2019)
(Pharmacology,1891)
(Adolescent,1884)
(Pathology,1781)
(Rats,1573)
(Infant,1568)
(Geriatrics,1546)
(Pregnancy,1431)

Most Popular Topics

(Research,5591)
(Child,2235)
(Infant,1388)
(Toxicology,1251)
(Pharmacology,1242)
(Rats,1067)
(Adolescent,1025)
(Surgical Procedures, Operative,1011)
(Pregnancy,996)
(Pathology,967)

Notice the difference between popularity and high degree, a topic can be popular but alone, not connected to a lot of other things, like e.g. surgical Procedures.

Outline of Analysis

- ~~Parsing XML Documents~~
- ~~Basic stats:~~
  - ~~Topic popularity~~
  - ~~Topic co-occurrence~~
- ~~Construct a co-occurrence graph~~
- ~~Compute Graph stats:~~
  - ~~Connected Components~~
  - ~~Degree Distribution~~
- Filtering Out Noisy Edges
- Small-world network analysis

Small-World Networks

- Data scientists now have rich data sets that describe the structure and formation of real-world networks versus the idealized networks that mathematicians and graph theorists have traditionally studied.
- One of the first papers to describe the properties of these real-world networks, and how they differed from the idealized models, was published in 1998 by Duncan Watts and Steven Strogatz and was titled “Collective dynamics of ‘small-world’ networks”.
- It was a seminal paper that outlined the first mathematical model for how to generate graphs that exhibited the two “small-world” properties that we see in real-world graphs

Small World Networks

- A more sophisticated measure of the “connectedness” of a graph
- A graph is a ”Small-World” network if:
  - Most of the nodes in the network have a small degree and belong to a relatively dense cluster of other nodes; that is, a high fraction of a node’s neighbors are also connected to each other.
  - Despite the small degree and dense clustering of most nodes in the graph, it is possible to reach any node in the network from any other network relatively quickly by traversing a small number of edges.
- There are concrete metrics that could be used to rank graphs based on how strongly they expressed these properties

Cliques

- A graph is *complete* if every vertex is connected to every other vertex by an edge.
- In a given graph, there may be many subsets of vertices that are complete, and we call these complete subgraphs *cliques*.
- The presence of many large cliques in a graph indicates that the graph has the kind of locally dense structure that we see in real small- world networks.
- Unfortunately, finding Cliques is NP-complete, but there are approximation algorithms which could can luckily also be run in parallel.
- GraphX can do that and that shows the first property of a Small-World graph

Shortest Path Length

- The second property of small-world networks is that the length of the shortest path between any two randomly chosen nodes tends to be small. In this section, we'll compute the average path length for nodes contained in the large connected component of our filtered graph.

Where to Go from Here

- At first, small-world networks were a **curiosity**; it was interesting that so many different types of real-world networks, from sociology and political science to neuroscience and cell biology, had such similar and peculiar structural properties.
- More recently, it seems that **deviances** from small-world structure in these networks can be indicative of the potential for **functional problems.**
- Examples:
  - Neurons in the brain exhibits a small-world structure,
  - Deviance from this structure occurs in patients with Alzheimer's disease, schizophrenia, depression
- In general, real-world graphs should exhibit the small-world property; if they do not, that may be evidence of a problem,