RECOMMENDER SYSTEMS AND GRAPH PROCESSING

CKME 134 – BIG DATA ANALYTICS TOOLS
RYERSON UNIVERSITY
WINTER 2015

Instructor: Shaohua Zhang

Course Outline

1. Intro to Big Data
2. Distributed Computing and MapReduce
3. Hadoop Ecosystem
4. Programming Hive
5. Advanced Hive
6. Mid-Term Review
7. Programming Pig
8. Advanced Pig
9. Hadoop Use Cases
10. Building Data Product & Next-Gen Hadoop (Spark)
11. Beyond Hadoop: Graph Analytics and Recommender Systems

Exam and Preparation

- Final Exam Preparation
  - Course Notes – Session 1~Session 9
  - Hive Intro Lab – Session 4 Lab
  - Pig Intro Lab 1 – Session 7 Lab
- Hangout Session 2
  - Engineering Building (lower level)
  - Saturday, April 11
  - 1pm ~ 4pm

Lecture 11 - Outline

- Recommender Systems 101
- Graph Processing
  - Spark GraphX
  - GraphLab
  - Giraph

PageRank Algorithm

- PageRank gives web pages a ranking score on links from other pages
  - Links from many pages ➔ high rank
  - Link from a high-rank page ➔ high rank

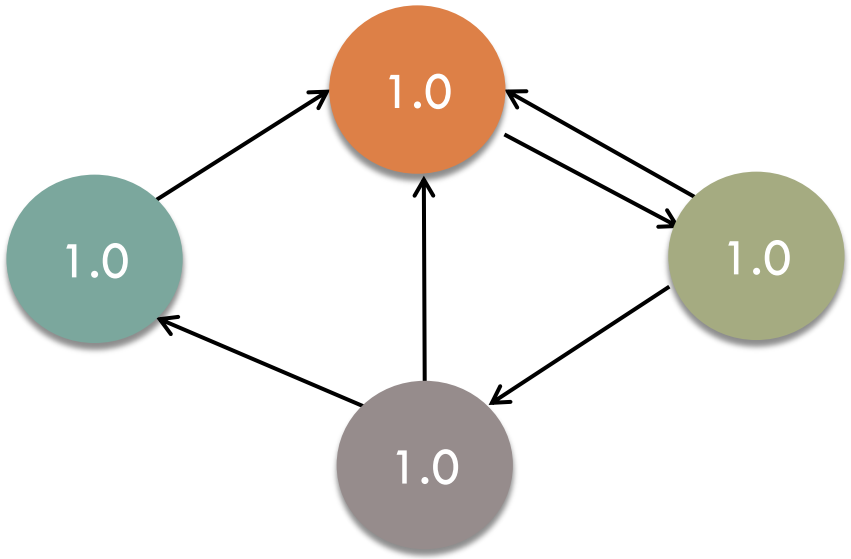
Graph (directed)

PageRank Explained – Initial State

1. Start each page at a rank of 1

```
# set initial page ranks to 1.0
ranks=links.map(lambda (page,neighbors): (page,1.0))
```

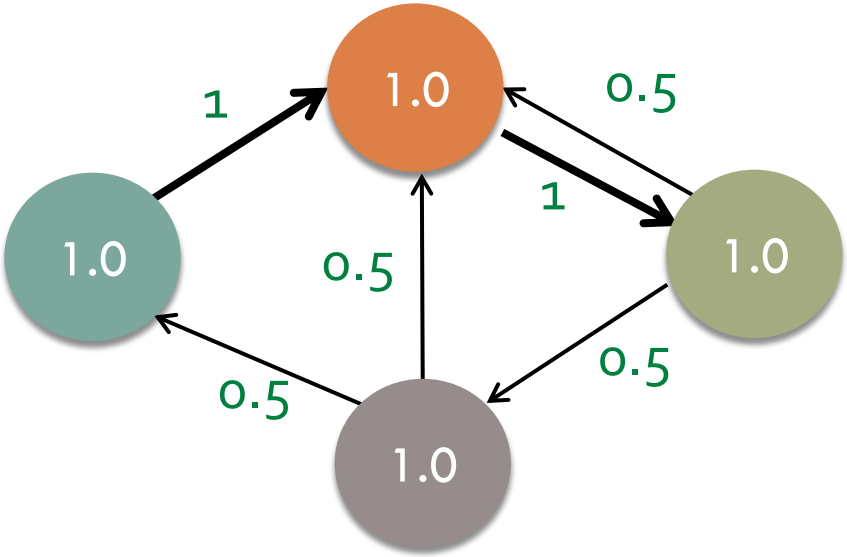
```
links = sc.textFile(linkfile).map(lambda line: line.split())\
    .map(lambda pages: (pages[0],pages[1]))\
    .distinct()\
    .groupByKey()\
    .cache()
```



PageRank Explained – Iteration 1

1. Start each page at a rank of 1
2. On each iteration, have page p contribute $contrib_p = rank_p / neighbors_p$ to its neighbors

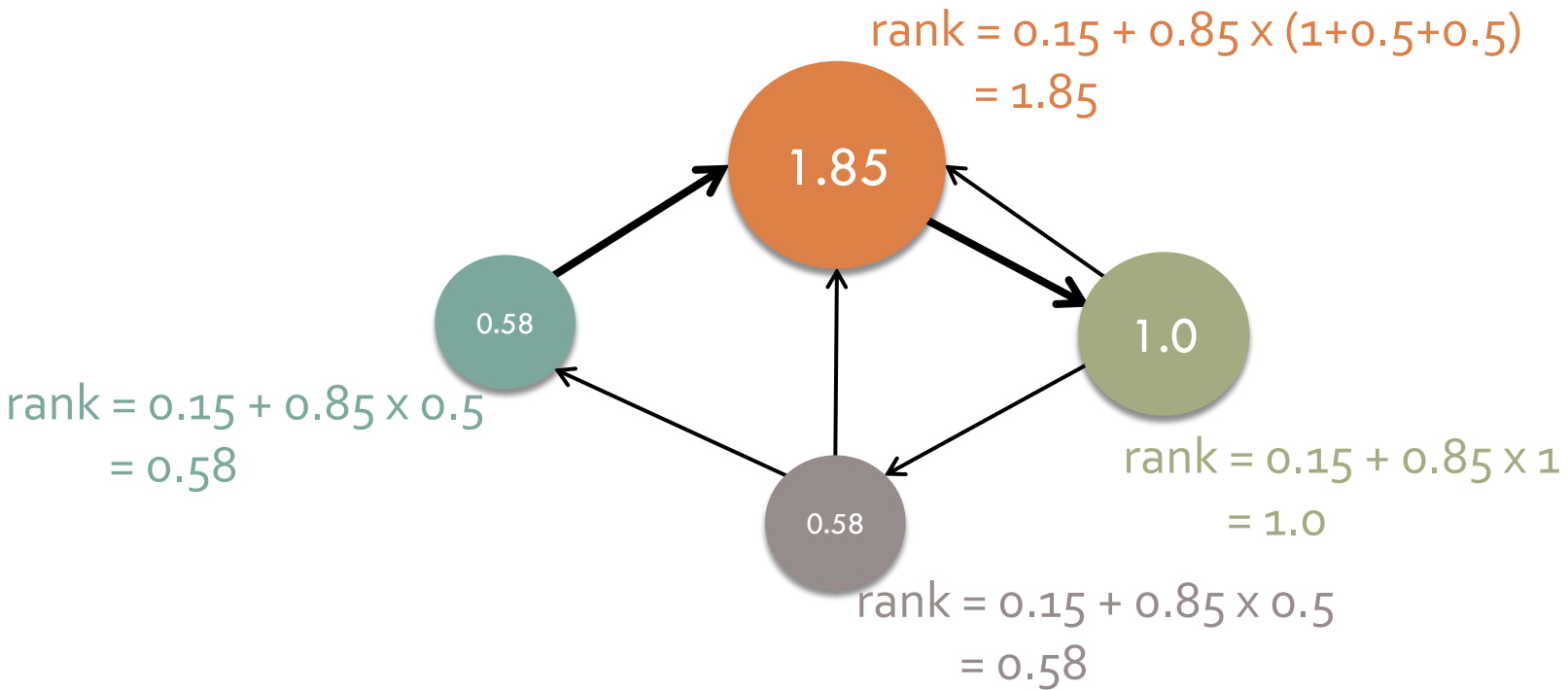
```
def computeContribs(neighbors, rank):
    for neighbor in neighbors:
      yield(neighbor, rank/len(neighbors))
```



PageRank Explained – Iteration 1

1. Start each page at a rank of 1
2. On each iteration, have page p contribute $contrib_p = rank_p / neighbors_p$ to its neighbors
3. Set each page's rank to 0.15 + 0.85 × contribs

```
for x in xrange(n):
  contribs=links\
    .join(ranks)\
    .flatMap(lambda (page,(neighbors,rank)): \
        computeContribs(neighbors,rank))
  ranks=contribs\
    .reduceByKey(lambda v1,v2: v1+v2)\
    .map(lambda (page,contrib): \
        (page,contrib * 0.85 + 0.15))
  print "Iteration ",x
  for pair in ranks.take(10): print pair
```

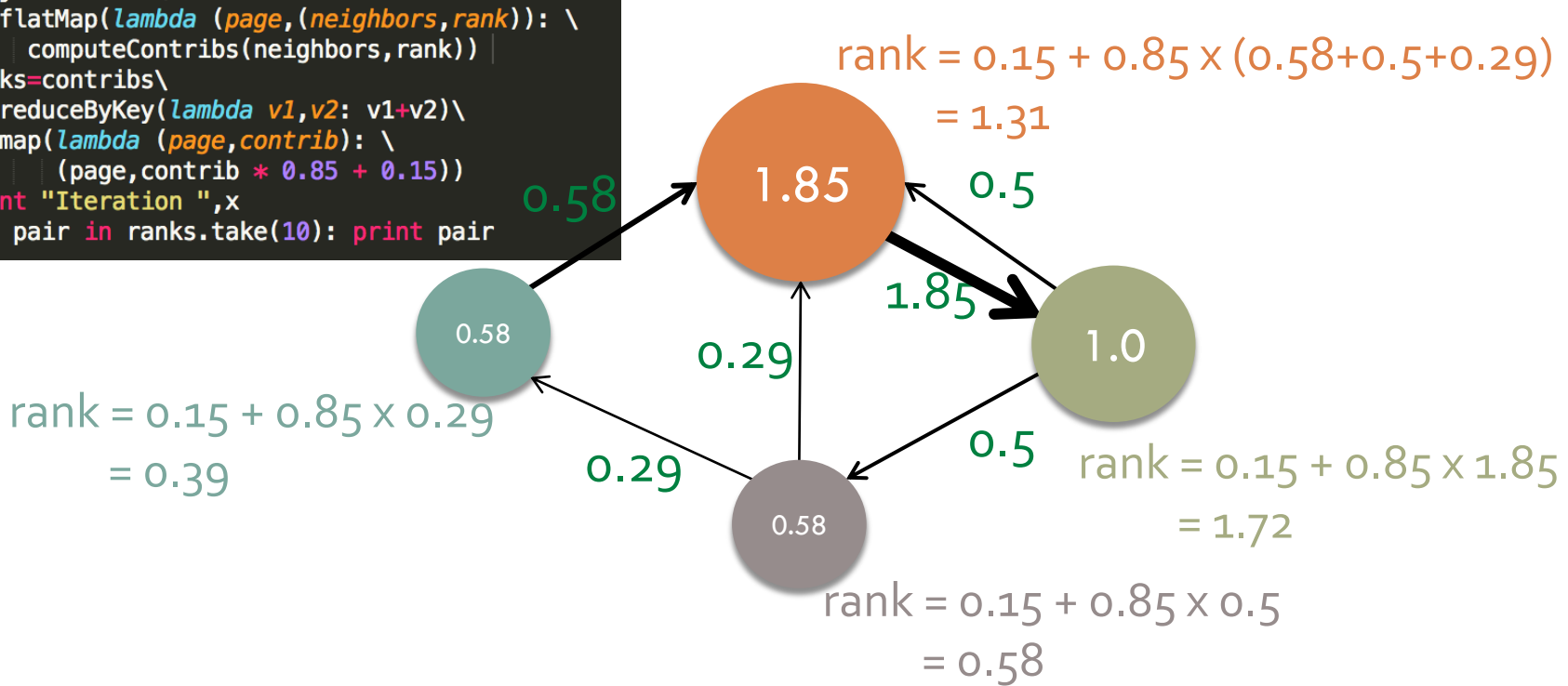


PageRank Explained – Iteration 2

1. Start each page at a rank of 1
2. On each iteration, have page p contribute $contrib_p = rank_p / neighbors_p$ to its neighbors
3. Set each page's rank to 0.15 + 0.85 × contribs

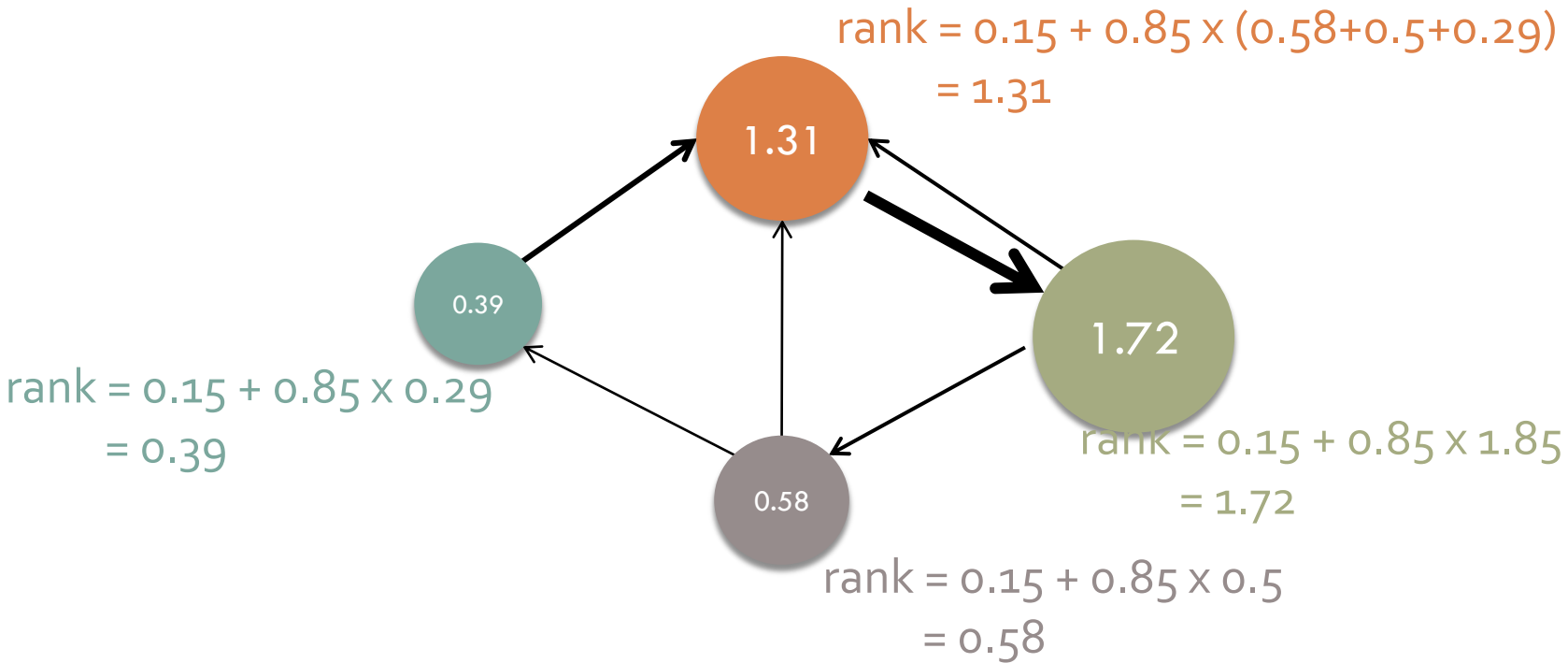
```
def computeContribs(neighbors, rank):
    for neighbor in neighbors:
      yield(neighbor, rank/len(neighbors))
```

```
for x in xrange(n):
  contribs=links\
    .join(ranks)\
    .flatMap(lambda (page,(neighbors,rank)): \
       computeContribs(neighbors,rank))
  ranks=contribs\
    .reduceByKey(lambda v1,v2: v1+v2)\
    .map(lambda (page,contrib): \
        (page,contrib * 0.85 + 0.15))
  print "Iteration ",x
  for pair in ranks.take(10): print pair
```



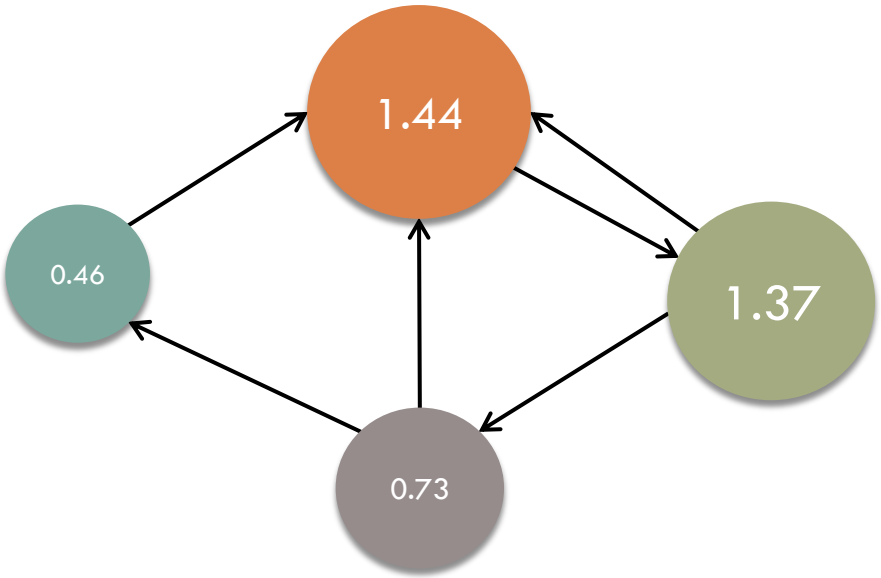
PageRank Explained – Iteration 2

1. Start each page at a rank of 1
2. On each iteration, have page p contribute $contrib_p = rank_p / neighbors_p$ to its neighbors
3. Set each page's rank to 0.15 + 0.85 × contribs



PageRank Explained – Final State

1. Start each page at a rank of 1
2. On each iteration, have page p contribute $contrib_p = rank_p / neighbors_p$ to its neighbors
3. Set each page's rank to $0.15 + 0.85 \times contribs$



Spark Demo

- PageRank Implementation in Spark
  - Download and install Cloudera 5.3 Virtual Machine (Lecture 10 Lab)
  - In Virtualbox, create a new VM
  - Name "Cloudera 5.3 VM with Spark"
  - Type "Linux"
  - Version "Other Linux (64-bit)
  - Set memory size to 4096
  - Choose "Use an existing virtual hard drive file" and select in the directory "Cloudera-QuickStart-VM-5.3.0-0-virtualbox-disk1.vmdk"
  - Create and start the VM
  - Once you're in the Cloudera 5.3 VM, download the pagelinks.txt dataset and PageRank.py script into "/home/cloudera/Downloads" folder
  - Go to this folder, and run the PageRank.py script
  - $ pyspark PageRank.py
  - After several iterations, you should be able to see the pagerank results printed

Introduction to Recommender Systems

Recommender Systems

Examples

- Movie
  - Netflix
- Music
  - Spotify/Pandora/Songza
- Social
  - LinkedIn/Facebook/Twitter
- e-Commerce
  - Amazon/eBay/Taobao
- LBS
  - Foursquare/Yelp!
- News
  - Yahoo!/Flipboard/FB News Feeds
- Events
  - Eventbrite
- Fashion
  - Pose (fashion style graph)
- Dating
  - eHarmoney/OkCupid

Recommender Systems

Why Recommendations?

- Consumer Perspective
  - Information Overload
    - Too many products/items
  - Efficiency and relevancy
  - Personalized experience
- Company Perspective
  - Product discoveries → Cross-sell
  - Engagement → Ads
  - Customer satisfaction

Recommender Systems

Types of Recommendations

- Editorial and hand curated
  - Newspaper
- Simple aggregates
  - Top 10, Most Popular
- Personalized – tailored to users
  - Amazon, Netflix
  - Facebook News Feeds

Recommender Systems

Types of Data

- Star Ratings – Explicit
  - Netflix Prize
- Beyond Ratings
  - Implicit Cues
    - Clicks, Likes, Comments
    - Download/Install
    - Time Spent on page
  - User Profiles
    - Demographics → Gender, Age, Marital Status
    - Interests → Sailing, Music, Karate
  - Social Signals
    - Facebook graph
    - Tweets
  - Contextual
    - Location
    - Weather
    - Mode of Transportation
    - etc.

Recommender Systems

Types of Algorithms

- Content-based
- Collaborative Filtering
  - Memory-based
    - User-User
    - Item-Iterm
  - Model-based
- Hybrid

Recommender Systems

Content-based Recommenders

- Main Idea: Recommend items to customer **x** similar to previous items rated highly by **x**
- Examples
  - Movie Recommendation
    - Recommend movies with same directors, actors, genre
  - News Recommendation
    - Recommend news with similar content/topics

Recommender Systems

Content-based Recommenders – Item Profiles

- For each item, create an item profile
- Profile is a set of features
  - Movies: author, title, actors, director, …
  - News: Set of important keywords in document

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

$$IDF_i = \log \frac{N}{n_i}$$

$$w_{ij} = TF_{ij} \times IDF_i$$

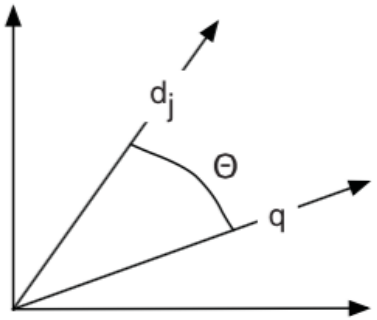
	Director		Title, Metadata					Actor					Genre				
	Steven	Woody	Speed	War	Love	Future	Home	Brad Pitt	Jennifer Laurence	Bradley Cooper	Tom Hanks	Jennifer Aniston	Horror	Action	Love	Cartoon	International
Movie 1	1				1		1	1	1			1	1		1		
Movie 2		1		1					1	1	1			1			
Movie 3	1				1	1		1				1			1		
Movie 4	1		1				1		1		1		1				

Recommender Systems

Content-based Recommenders – User Profiles & Predictions

	Director		Title, Metadata					Actor					Genre				
	Steven	Woody	Speed	War	Love	Future	Home	Brad Pitt	Jennifer Laurence	Bradley Cooper	Tom Hanks	Jennifer Aniston	Horror	Action	Love	Cartoon	International
Tom	0.87		0.3	0.5				0.9	0.77		0.34		0.67			0.33	0.54

- User Profiles
  - Weighted average of rated item profiles
  - Taxonomy/Ontology based
  - etc.
- Prediction
  - Cosine similarity between item profiles and user profiles



$d_j = \langle w_{1,j}, w_{2,j}, \ldots, w_{n,j} \rangle$

$q = \langle w_{1,q}, w_{2,q}, \ldots, w_{n,q} \rangle$

w = weight assigned to term

	Director		Title, Metadata					Actor					Genre				
	Steven	Woody	Speed	War	Love	Future	Home	Brad Pitt	Jennifer Laurence	Bradley Cooper	Tom Hanks	Jennifer Aniston	Horror	Action	Love	Cartoon	International
Movie 1	1				1		1	1	1			1	1		1		
Movie 2		1		1					1	1	1			1			
Movie 3	1				1	1		1				1			1		
Movie 4	1		1				1		1		1		1				
Tom	0.87		0.3	0.5				0.9	0.77		0.34		0.67			0.33	0.54

Recommender Systems

Content-based Recommenders – Pros and Cons

- No cold start or sparsity problems
  - No dependency on other items/users
- Able to recommend to users with unique tastes
- Able to recommend new and unpopular items
- Easy to explain to users

Recommender Systems

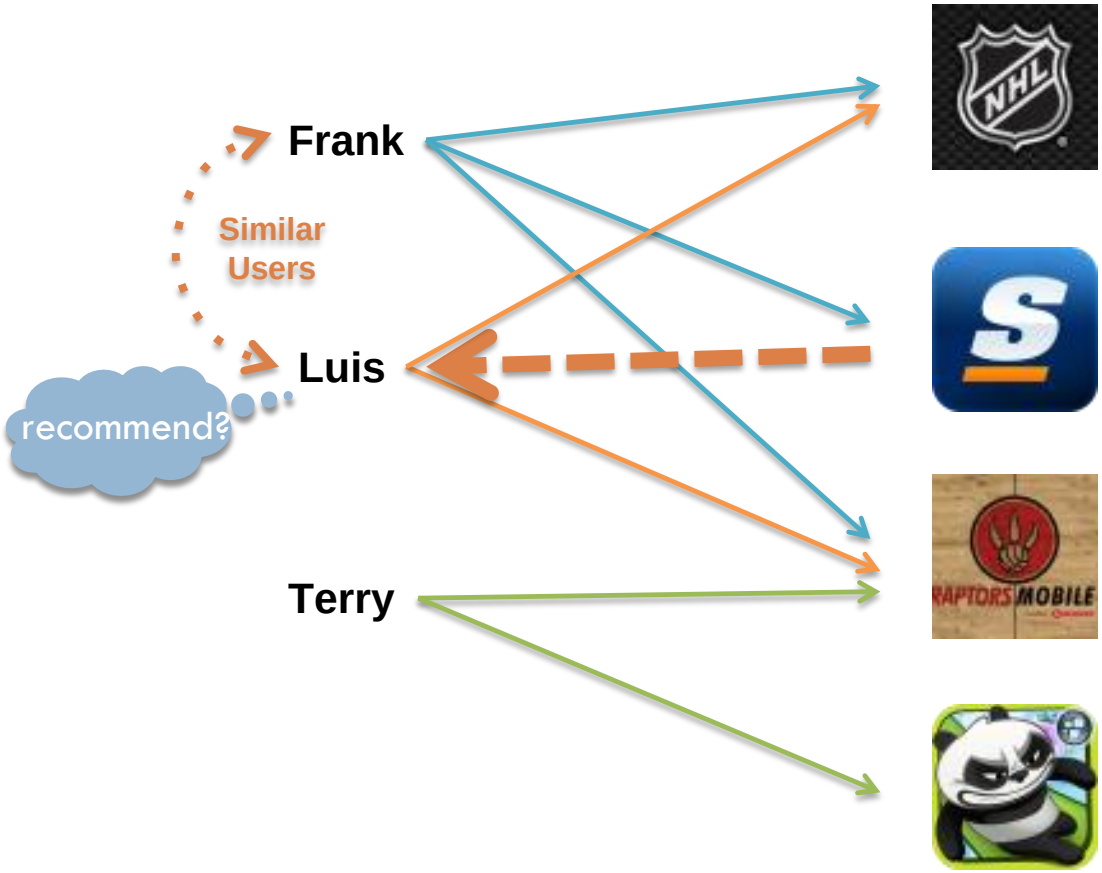
*Collaborative Filtering – **User-based** Recommender*

- Consider user **x**
- Find set of **k** other users whose ratings are "similar" to *x*'s ratings
- Estimate **x**'s ratings based on ratings of users in **k**

Recommender Systems

*Collaborative Filtering – **User-based** Recommender*

- Finding similar users



Recommender Systems

*Collaborative Filtering – **User-based** Recommender*

- Similarity Metrics
  - Jaccard
  - Cosine Similarity
  - Pearson Correlation

Jaccard similarity

$$sim(\vec{x_i}, \vec{x_j}) = \frac{|\vec{x_i} \cap \vec{x_j}|}{|\vec{x_i} \cup \vec{x_j}|}$$

Cosine Similarity

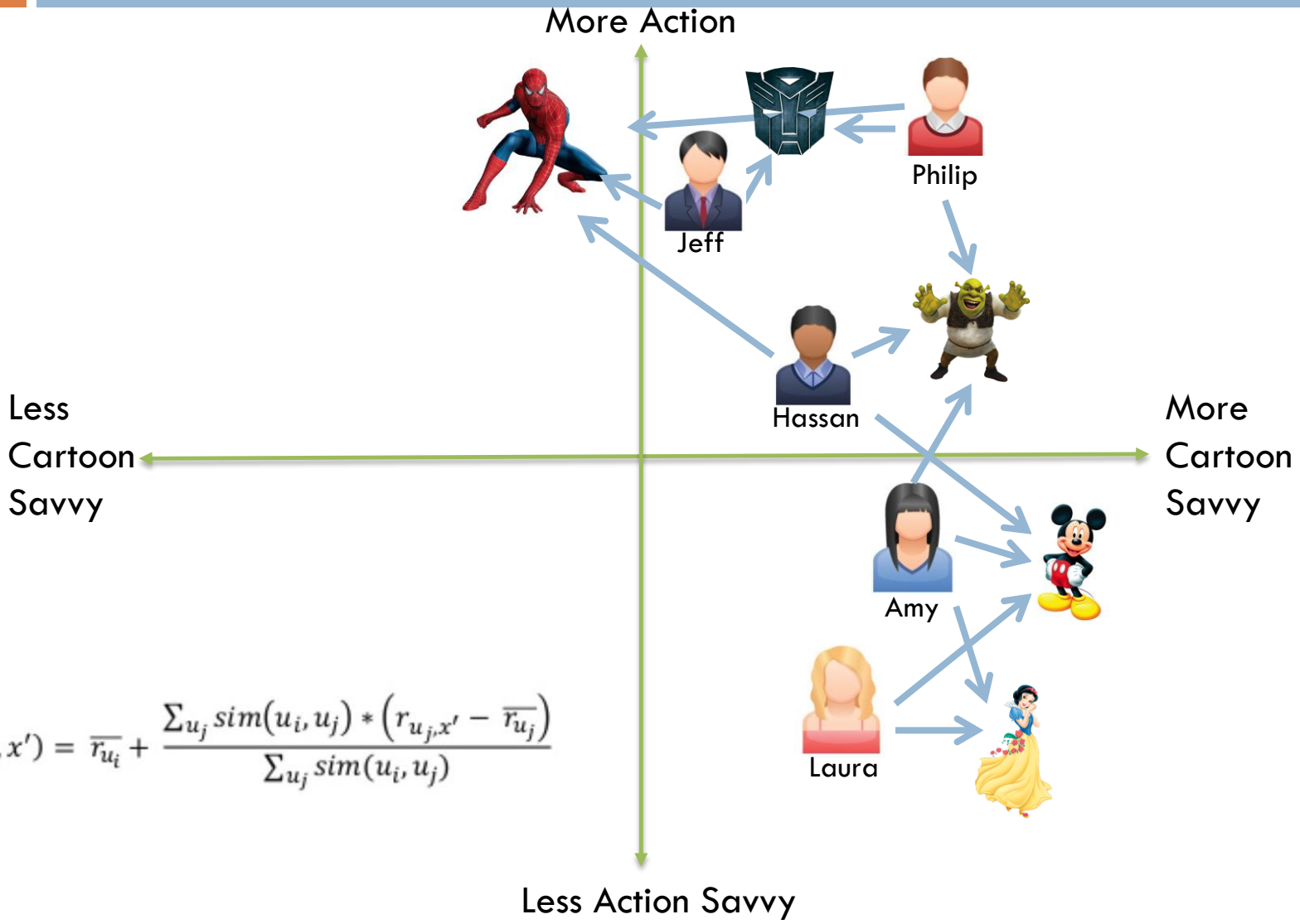
$$sim(\vec{x_i}, \vec{x_j}) = \frac{\vec{x_i} \cdot \vec{x_j}}{|\vec{x_i}| * |\vec{x_j}|} = \frac{\sum_u r_{u,x_i} * r_{u,x_j}}{\sqrt{\sum_u r_{u,x_i}^2} * \sqrt{\sum_u r_{u,x}^2}}$$

Pearson's correlation coefficient

$$sim(u_i, u_j) = \frac{\sum_{x \in X} (r_{u_i,x} - \overline{r_{u_i}}) * (r_{j,x} - \overline{r_{u_j}})}{\sqrt{\sum_{x \in X} (r_{u_i,x} - \overline{r_{u_i}})^2} * \sqrt{\sum_{x \in X} (r_{u_j,x} - \overline{r_{u_j}})^2}}$$

Recommender Systems

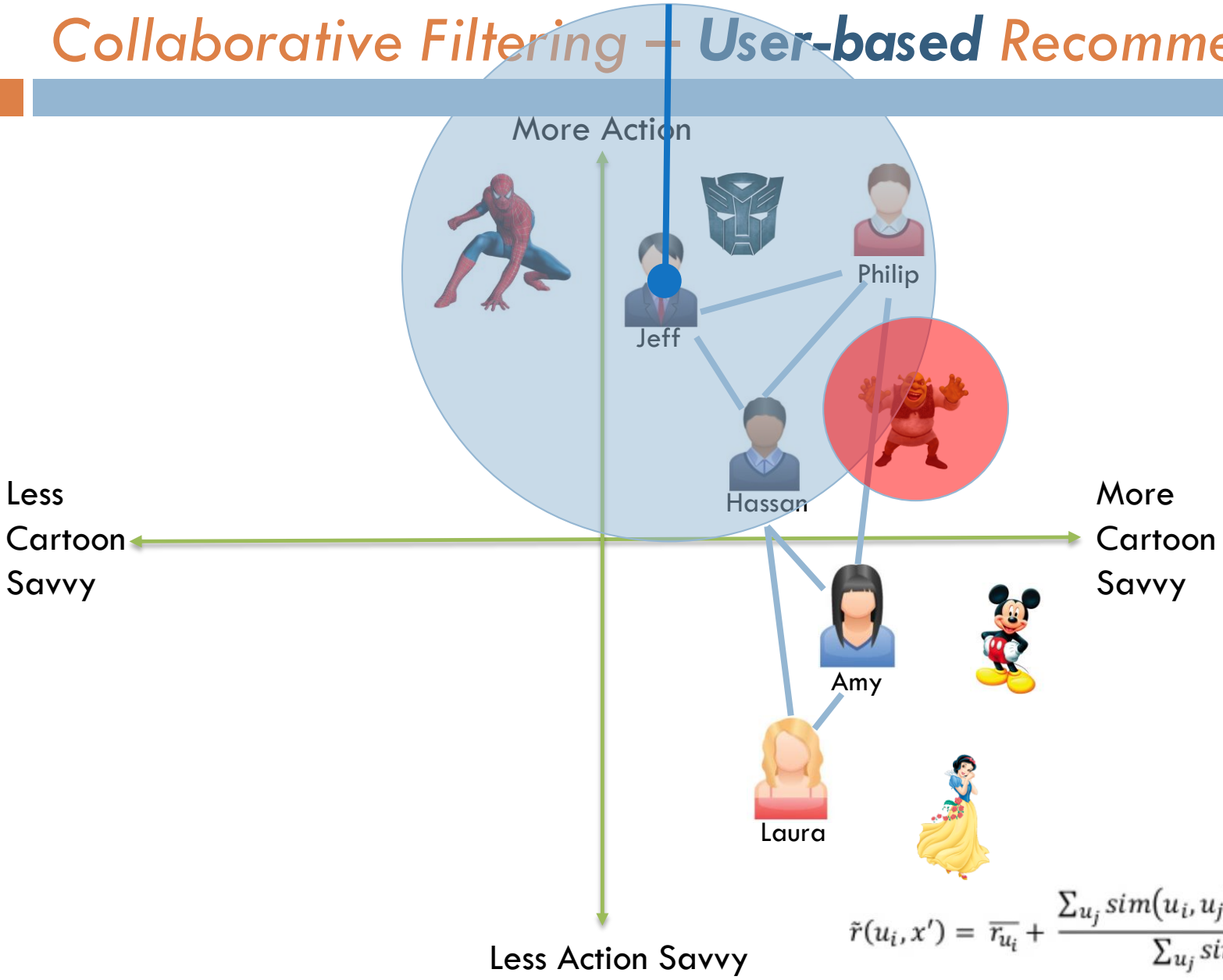
*Collaborative Filtering – **User-based** Recommender*



$$\tilde{r}(u_i, x') = \overline{r_{u_i}} + \frac{\sum_{u_j} sim(u_i, u_j) * \left(r_{u_j, x'} - \overline{r_{u_j}}\right)}{\sum_{u_j} sim(u_i, u_j)}$$

Recommender Systems

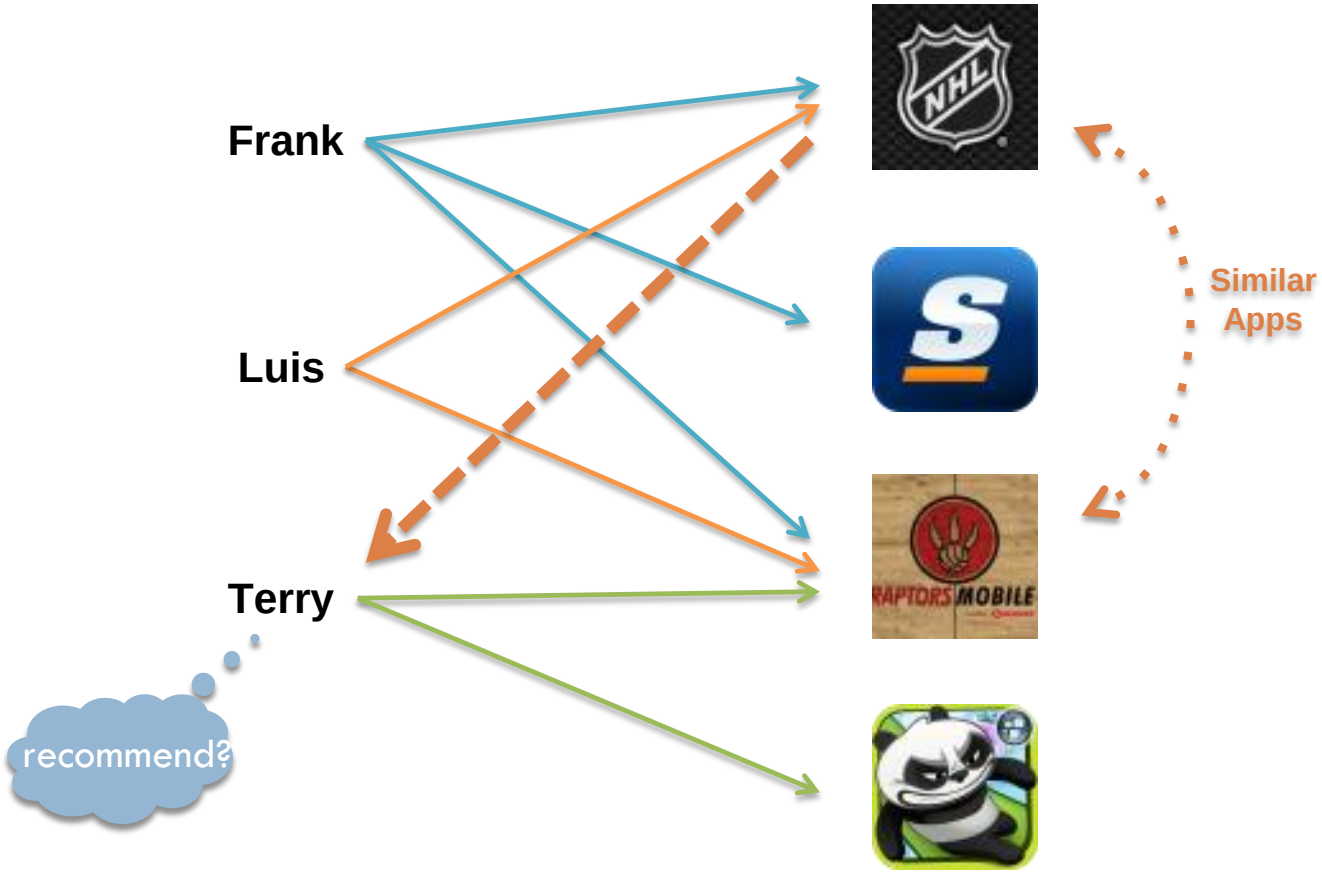
*Collaborative Filtering – **User-based** Recommender*



$$\tilde{r}(u_i, x') = \overline{r_{u_i}} + \frac{\sum_{u_j} sim(u_i, u_j) * \left(r_{u_j, x'} - \overline{r_{u_j}}\right)}{\sum_{u_j} sim(u_i, u_j)}$$

Recommender Systems

*Collaborative Filtering – **Item-based** Recommender*



Recommender Systems

*Collaborative Filtering – **Item-based** Recommender*





Movie Watching Logs	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Jeff	1			1	
Amy		1	1		1
Laura			1		1
Hasan	1	1			1
Fillip	1	1		1	

Movie Watching Matrix





Co-occurrence Matrix	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Spiderman	3	2	0	2	1
Shrek	2	3	1	1	2
Snowwhite	0	1	2	0	2
Transformers	2	1	0	2	0
Mickey Mouse	1	2	2	0	3





Co-occurrence Matrix

Recommender Systems

*Collaborative Filtering – **Item-based** Recommender*





Co-occurrence Matrix	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Spiderman	3	2	0	2	1
Shrek	2	3	1	1	2
Snowwhite	0	1	2	0	2
Transformers	2	1	0	2	0
Mickey Mouse	1	2	2	0	3

Co-occurrence Matrix





Cosine Similarity Matrix	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Spiderman	1	0.67	0	0.82	0.33
Shrek	0.67	1	0.41	0.41	0.67
Snowwhite	0	0.41	1	0	0.82
Transformers	0.82	0.41	0	1	0
Mickey Mouse	0.33	0.67	0.82	0	1





Cosine-Similarity Matrix

Recommender Systems

*Collaborative Filtering – **Item-based** Recommender*

Movie Watching Logs	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Jeff	1			1	
Amy		1	1		1
Laura			1		1
Hasan	1	1			1
Fillip	1	1		1	

Movie Watching Matrix

×

Cosine Similarity Matrix	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Spiderman	1	0.67	0	0.82	0.33
Shrek	0.67	1	0.41	0.41	0.67
Snowwhite	0	0.41	1	0	0.82
Transformers	0.82	0.41	0	1	0
Mickey Mouse	0.33	0.67	0.82	0	1

Cosine Similarity Matrix

=

Prediction	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Jeff	1	1.08	0	1	0.33
Amy	1	1	1	0.41	1
Laura	0.33	1.08	1	0	1
Hasan	1	1	1.23	1.23	1
Fillip	1	1	0.41	1	1

Movie Watching Logs	Spiderman	Shrek	Snowwhite	Transformers	Mickey Mouse
Jeff	1			1	

×

Cosine Similarity Matrix	Shrek
Spiderman	0.67
Shrek	1
Snowwhite	0.41
Transformers	0.41
Mickey Mouse	0.67

=

Prediction	Shrek
Jeff	1.08

Recommender Systems

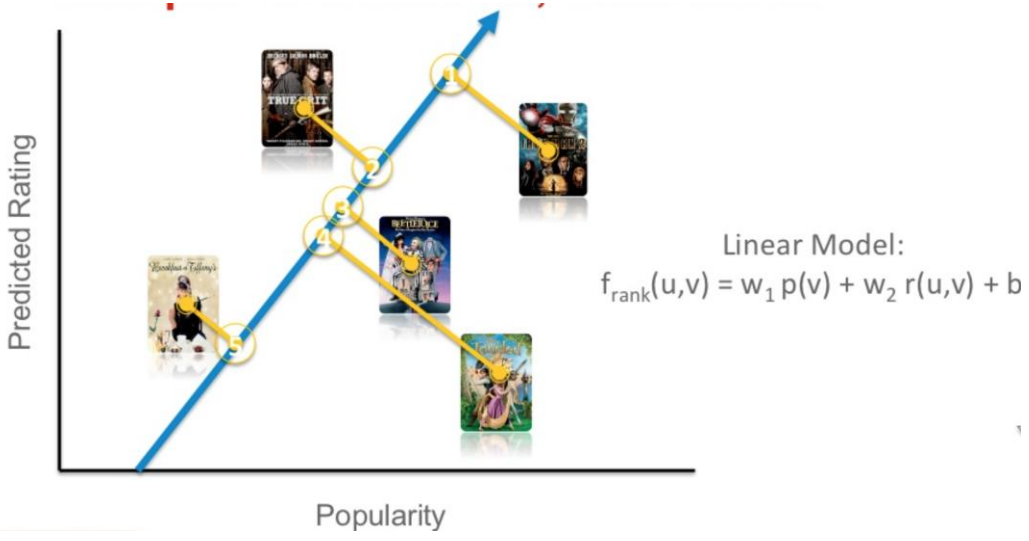
Collaborative Filtering – Pros and Cons

- Pros
  - Works for any kind of item
    - No feature selection needed
  - Predictive when you have lots of data
- Cons
  - Cold start (not enough users/items)
  - Sparsity
    - The user/ratings matrix is sparse
    - Hard to find users that have rated the same items
  - Popularity-driven
    - Cannot recommend items to someone with unique tastes
    - Cannot recommend items that are less popular

Recommender Systems

Hybrid Recommenders

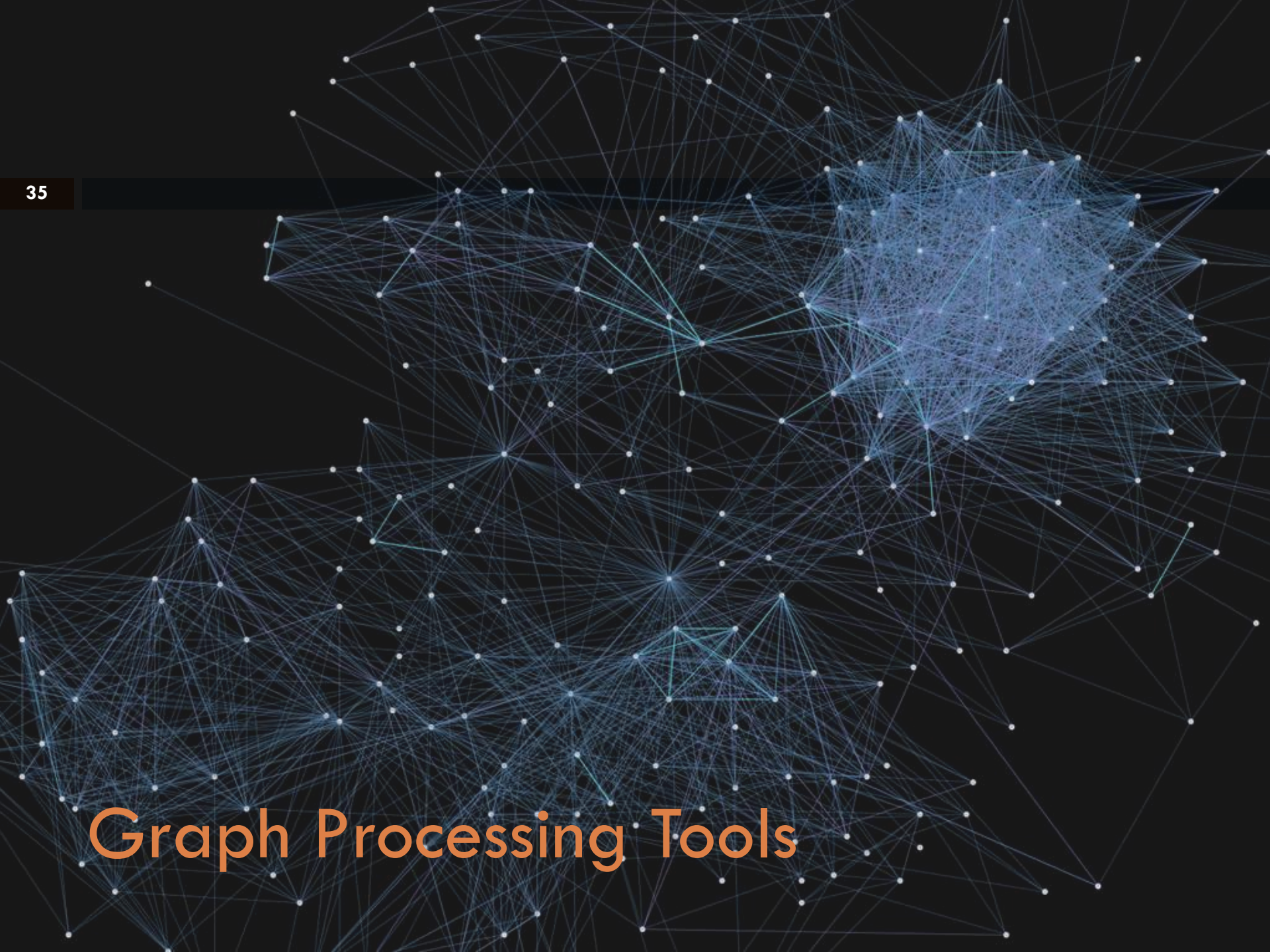
- Hyrbrid Approaches
  - Implement two or more different recommenders and combine predictions
  - Add content-based methods to CF
    - Item profiles for new item problems
    - User profiles for new user problems
  - Learning to Rank!



Recommender Systems

Challenges

- Cold Start
  - New User Problem
  - New Item Problem
- Sparsity & High Dimensionality
  - Matrix Factorization
- Deployment
  - Graph database
  - NoSQL (Cassandra)



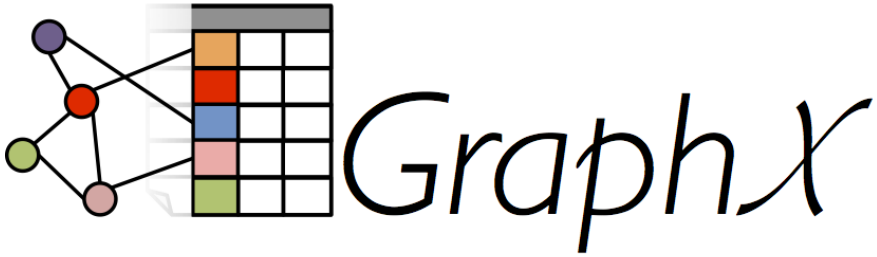
Graph Processing Tools

Problems that can be solved with graph

- Recommendation Algorithms
- Machine Learning
- Graph Computation
- Topic Modeling
- Label Propagation
- Probabilistic Graphic Modeling
- etc.

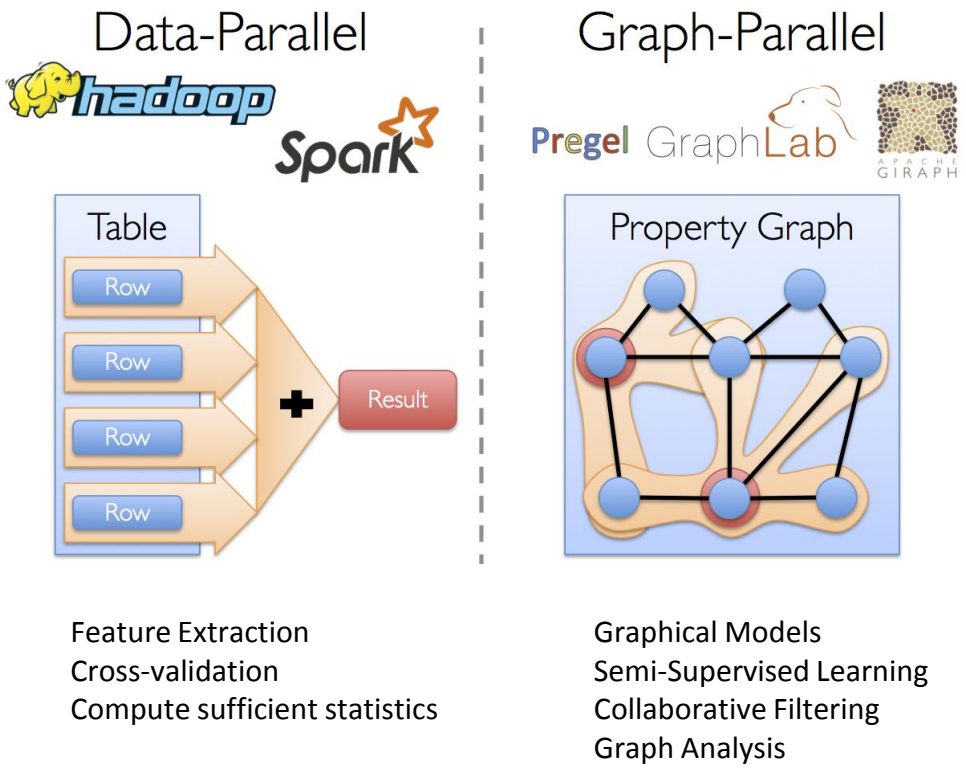
Graph Processing Tools

- GraphX
- GraphLab
- Giraph



Data Parallel vs Graph Parallel

- Challenges in partition and distribute graphs
- Orders of magnitude faster than more general *data-parallel* systems
- Dependency graph



Feature Extraction
Cross-validation
Compute sufficient statistics

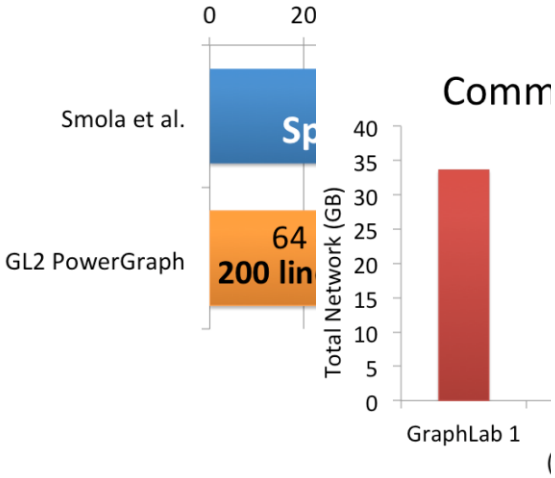
Graphical Models
Semi-Supervised Learning
Collaborative Filtering
Graph Analysis

GraphLab

- English language Wikipedia
  - 2.6M Documents, 8.3M Words, 500M Tokens
  - Computationally intensive algorithm

PageRank on the Twitter Follower Graph

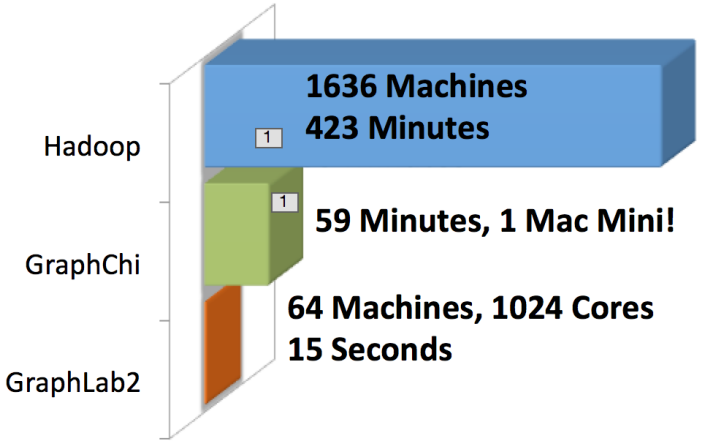
Natural Graph with 41M Users, 1.4 Billion Links



Triangle Counting on Twitter Graph

40M Users
1.2B Edges

Total: 34.8 Billion Triangles



Additional Lab

- Install GraphLab-Create on Cloudera 5.3 VM
  - Follow the instructions posted on blackboard
- Run a basic recommender on movie-lens dataset

Where to go from here?

Where to go from here?

Practice, Practice, Practice!

- Taking 50 data science courses don't make you a great data scientist
  - http://www.datasciencecentral.com/m/blogpost?id=6448529:BlogPost:262632
  - Practice
  - Read best practices
  - Understanding why!
- Books → read 2 to 3 good ones
  - One Practical predictive modeling
  - One statistical learing
  - One data preparation
  - One big data (hadoop, etc.)

Where to go from here?

Capstone Project

- Find a project that will allow you to practice Hadoop, R/Python and Visualization
  - Find a dataset
  - Define your problem
  - Set up your analytics environment (AWS)
  - Data munging in Hadoop
  - Modeling in R, Python and Spark
    - Try at least 4 different algorithms
  - Build a data pipeline
  - Publish the results to github as notebook
  - Create visualizations (data exploration, final product)

Where to go from here?

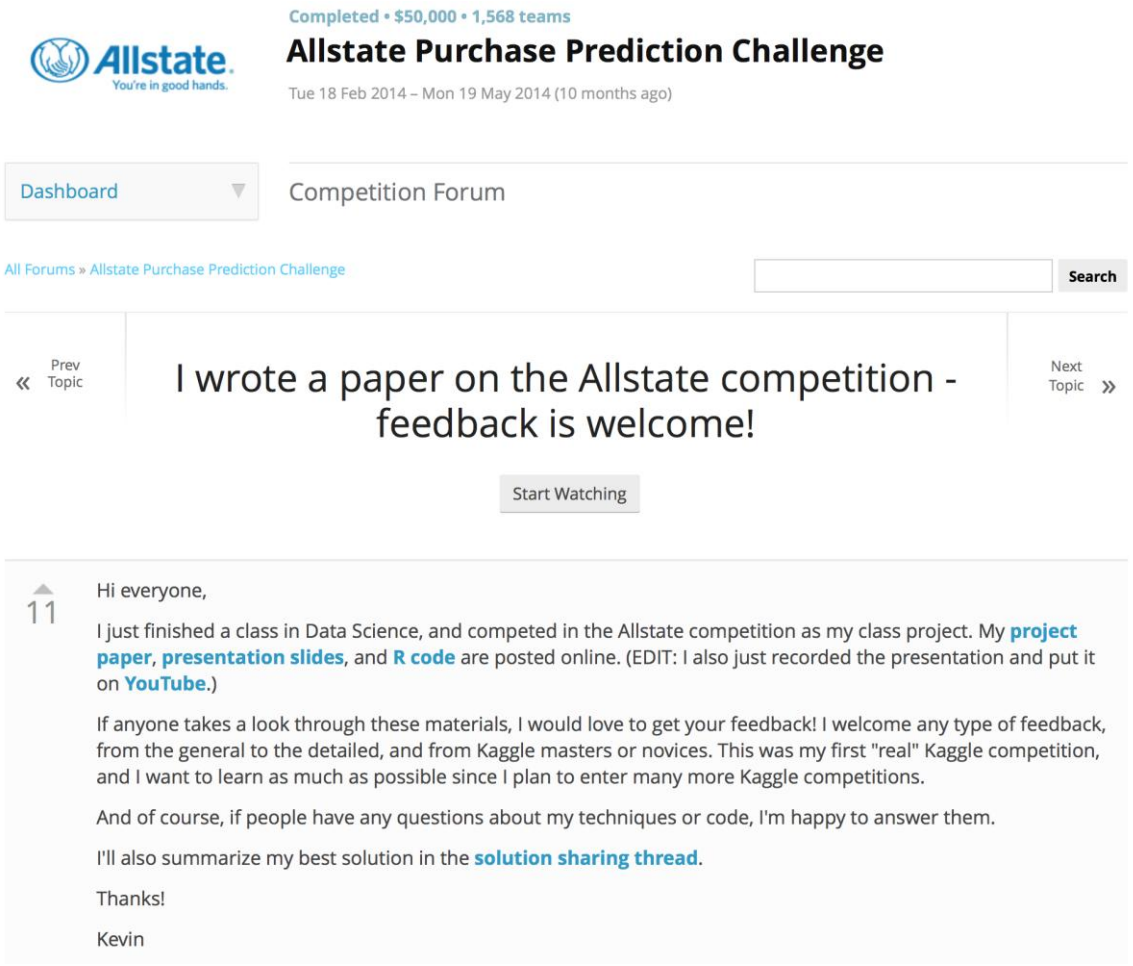
Work with Interesting Datasets

- Retail/Marketing
  - KDD Cup 1998 Direct Mail Campaign
  - Alibaba TianChi Competition
- Telco
  - Churn dataset (KDD Cup 2009 Orange)
- Banking
  - German credit data
- Social
  - Twitter graph
  - Foursquare graph
  - Tencent KDD 2012
- Fraud/Intrusion Detection
  - German credit fraud
  - Enron email fraud
  - Intrusion detection kdd99
- Geo
  - Geo-tagged tweets
  - NYC yellow cab
  - Foursquare
- Textual
  - Yelp! challenge
- Recommender
  - MovieLens
  - Netflix
  - Event Recommender (Kaggle)

Where to go from here?

Learn from Best Practices

- Kaggle Competition
  - Interesting problems
  - Tons of Python/R solutions posted
  - Learn from the best!
  - Join the competition to practice

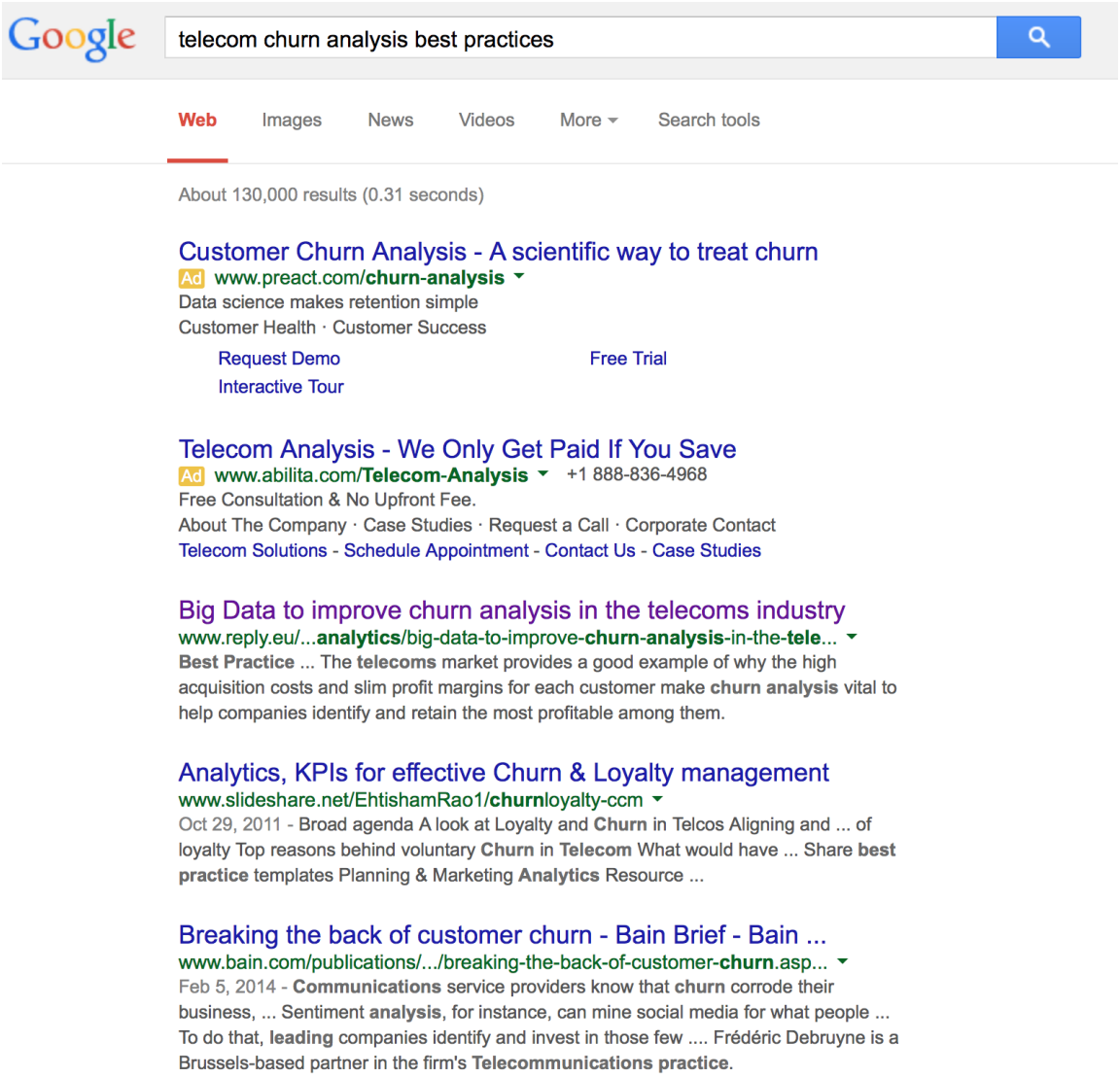


http://www.kaggle.com/c/allstate-purchase-prediction-challenge/leaderboard/private
https://github.com/justmarkham/kaggle-allstate/blob/master/allstate-paper.md

Where to go from here?

Learn from Best Practices

- Google



Where to go from here?

Learn from Best Practices

- GitHub
  - Learn from others
  - Start building your own repos!

ipython / ipython

Watch 490 Star 6,106 Fork 1,826

A gallery of interesting IPython Notebooks

Edit New Page

Fernando Perez edited this page 2 days ago · 229 revisions

This page is a curated collection of IPython notebooks that are notable for some reason. Feel free to add new content here, but please try to only include links to notebooks that include interesting visual or technical content; this should *not* simply be a dump of a Google search on every ipynb file out there.

Important contribution instructions: If you add new content, please ensure that for any notebook you link to, the link is to the rendered version using nbviewer, rather than the raw file. Simply paste the notebook URL in the nbviewer box and copy the resulting URL of the rendered version. This will make it much easier for visitors to be able to immediately access the new content.

Note that Matt Davis has conveniently written a set of bookmarklets and extensions to make it a one-click affair to load a Notebook URL into your browser of choice, directly opening into nbviewer.

Table of Contents

1. Entire books or other large collections of notebooks on a topic
   - Introductory Tutorials
   - Programming and Computer Science
   - Statistics, Machine Learning and Data Science
   - Mathematics, Physics, Chemistry, Biology
   - Earth Science and Geo-Spatial data
   - Linguistics and Text Mining
   - Signal Processing
2. Scientific computing and data analysis with the SciPy Stack

Pages 138

Find a Page...

Home

A gallery of interesting IPython Notebooks

Code blocks and other ideas

Cookbook: Branding the IPython notebook

Cookbook: Connecting to a remote kernel via ssh

Cookbook: Dated logging

Cookbook: deploying ipython

Cookbook: Dynamic prompt

Cookbook: Hotkeys for faster notebook editing

Cookbook: Ignoring some commands in history

Cookbook: Index

Cookbook: IPython and Emacs

Cookbook: IPython and Python.NET

Cookbook: Job control extension

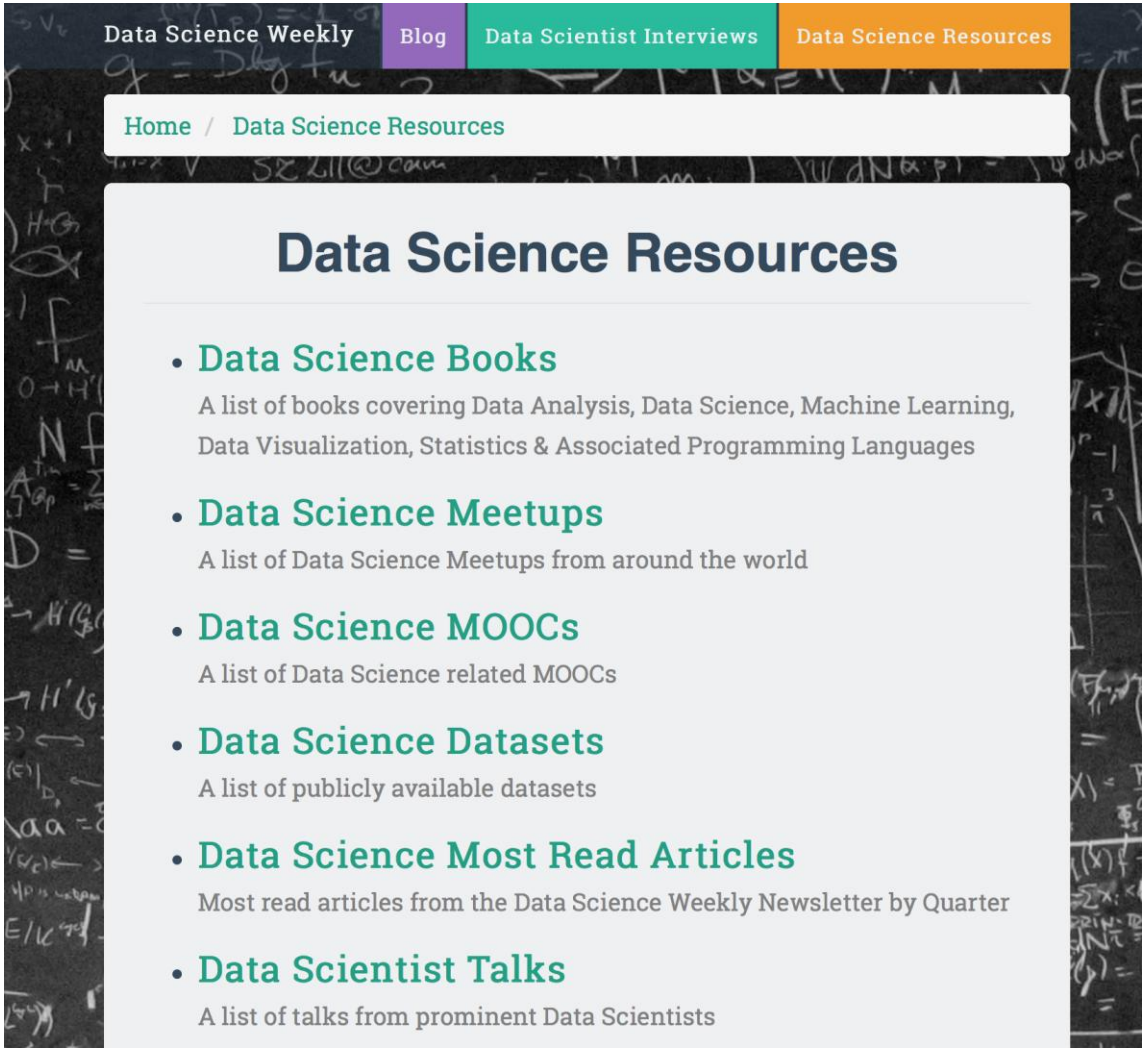
Cookbook: Launching IPython on OSX

https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks

Where to go from here?

Learn from Best Practices

- Read Data Science News/Blogs
  - DataTau
  - Data Science Weekly
  - Data Science Central
  - etc. (just Google)

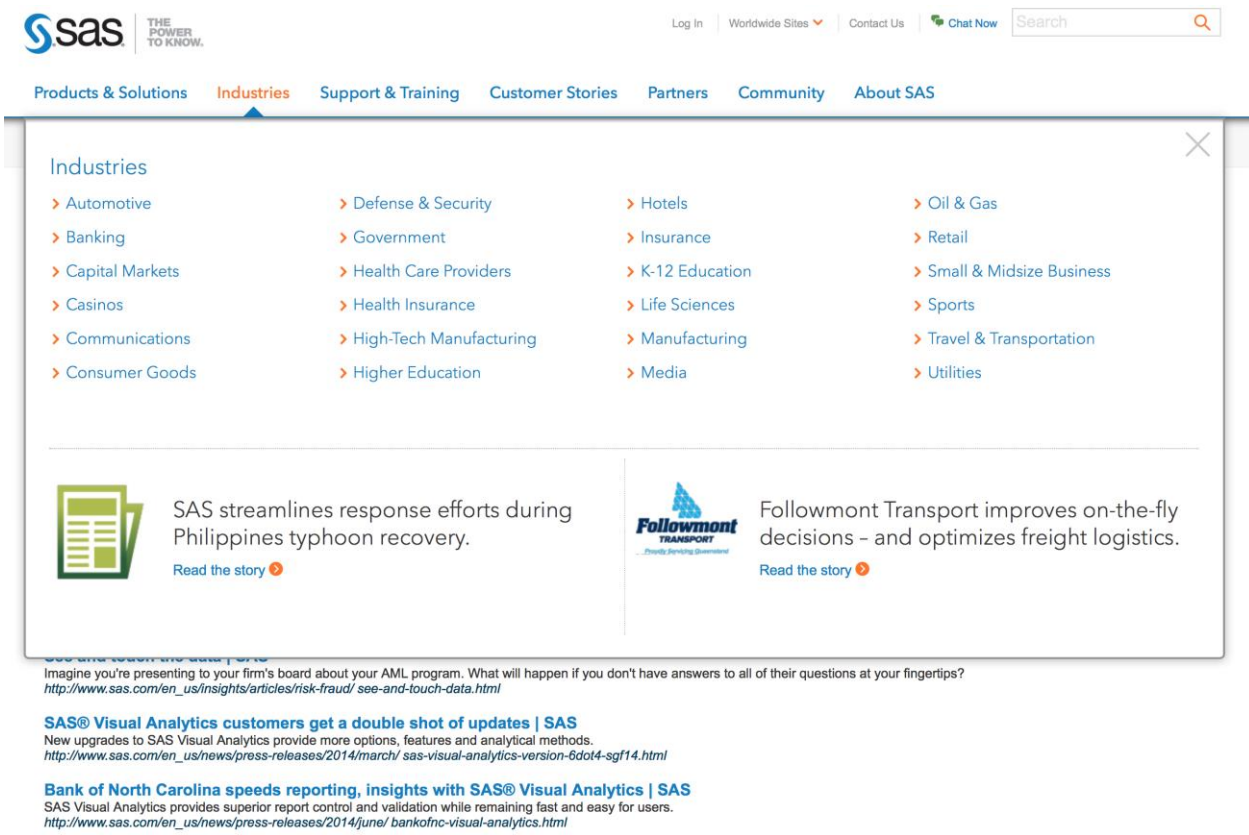


http://www.datascienceweekly.org/data-science-resources

Where to go from here?

Learn from Best Practices

- Read Industry Whitepapers
  - It is important to understand how things are actually applied
  - You'll find a lot of goodies in whitepapers



Where to go from here?

Sharpen your analytics skills

- Online Courses
  - Coursera – Machine Learning, Intro to Data Science
  - CMU – Machine Learning Summer School (free online)
- Blogs
- Data Science Bootcamps
- Kaggle
- Internship/Consulting
- Get Certified
  - Ryerson Certificate
  - Coursera Certificate
  - SAS Certificate in Predictive Modeling, Programming
  - Cloudera/Hortonworks

Where to go from here?

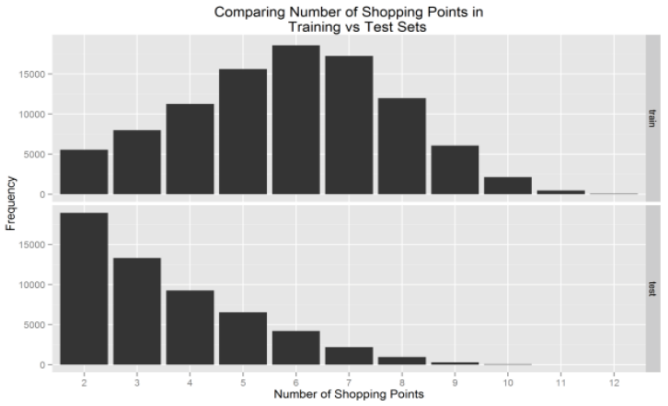
Preparing for Job Interviews

- Find an industry of interest
- Learn specific use cases
- Build your skills/portfolio (github)
- Don't get intimidated by the job requirements
- Find a way to talk to the hiring manager directly (LinkedIn)
  - Send her your github page
- Demonstrate your ability to learn new things
  - Most tools are easy to learn such as visualization tools, BI tools, etc.
- Remember that you're also interviewing the hiring manager

Data Exploration and Visualization

Here are some of my key findings from the exploratory process, and what I concluded from those findings.

1. **Missing values:**
   - risk_factor was NA is 36.1% of the training set and 38.0% of the test set. As a predictor that I considered potentially useful, I decided to impute the risk_factor for those customers using a linear regression model based on other customer characteristics.
   - C_previous and duration_previous were NA for 2.8% of the training set and 4.9% of the test set. I decided that those NA values were probably indicative of new customers, and thus I imputed values of 0 for duration_previous and "none" (a categorical variable) for C_previous.
   - location was NA for 0.3% of the test set. I decided to impute the location for each customer by copying the location from another customer in the same state.
3. **Number of shopping points:**
   - As seen in the plot below, the training set contained a roughly normal distribution of "shopping points" (the number of quotes a customer reviewed), whereas the test set contained a a very different distribution.



https://github.com/justmarkham/kaggle-allstate/blob/master/allstate-paper.md

- ✓ Good presentation
- ✓ Documentation skills, knowledge sharing
- ✓ Learning, constantly improving

By Application	Today, 1:13 AM
Call Centre Service	Dec 23, 2014, 4:38 PM
Clinical.Bioinformatic	Dec 23, 2014, 5:52 PM
Credit Risk	Dec 23, 2014, 5:47 PM
Cross_Up_sell	Dec 23, 2014, 5:46 PM
Customer Churn & Retention	Dec 23, 2014, 5:50 PM
Customer LTV	Dec 23, 2014, 5:43 PM
Database Marketing	Dec 23, 2014, 4:52 PM
Forecasting	Dec 23, 2014, 4:42 PM
Fraud Detection	Dec 23, 2014, 4:48 PM
Insurance	Dec 23, 2014, 5:46 PM
Real-time	Dec 23, 2014, 4:36 PM
Retailing	Dec 23, 2014, 4:42 PM
STOCK	Dec 23, 2014, 4:46 PM
Survival Analysis	Dec 23, 2014, 5:48 PM
Telecomm	Dec 23, 2014, 4:48 PM
Adaptive Customer Value Optimization for Communications Service Providers.pdf	Sep 11, 2007, 10:45 PM
Advanced Marketing Automation for telecomm.pdf	Sep 11, 2007, 10:45 PM
managing customer profitability in the telecomm industry.pdf	Sep 11, 2007, 10:45 PM
MG104_Overcoming%20CRM%20Challenges.pdf	Sep 6, 2008, 9:34 PM
MG105_How%20Cust%20Intelligence%20Improves%20CRM.pdf	Sep 6, 2008, 9:34 PM
The_Role_of_Mobile_Phones_in_Sustainable_Rural_Poverty_Reduction_June_2008.pdf	Sep 12, 2008, 4:00 PM
Text Mining	Dec 23, 2014, 4:50 PM
Uplift Modeling	Dec 23, 2014, 5:45 PM
By Techniques	Yesterday, 9:19 PM
# Data Mining Best Practice	Yesterday, 9:19 PM
# In General	Today, 12:25 PM
# SAS - Useful Macros	Yesterday, 9:15 PM
1 - Data Preparation	Dec 23, 2014, 4:35 PM
2 - Feature Selection	Dec 23, 2014, 4:38 PM
3 - Algorithms	Yesterday, 6:01 PM
4 - Assessment & Ensemble	Yesterday, 6:00 PM
5 - Big Data	Jan 4, 2015, 12:54 AM
Books	Feb 22, 2015, 3:36 PM
References	Jan 4, 2015, 12:50 AM
5 - Model Deployment & Validation	Yesterday, 5:59 PM

5 - Big Data	Jan 4, 2015, 12:54 AM	--
Books	Today, 12:51 PM	--
Cascading	Dec 23, 2014, 6:22 PM	--
Cassandra	Jan 4, 2015, 12:46 AM	--
Cloud	Dec 23, 2014, 4:37 PM	--
Coding Interview	Jan 11, 2015, 2:06 PM	--
Data Science	Jan 17, 2015, 11:03 PM	--
ElasticSearch	Jan 23, 2015, 1:23 PM	--
Graph Databases	Feb 23, 2015, 4:08 PM	--
h2oworld	Jan 4, 2015, 12:46 AM	--
Hadoop	Mar 3, 2015, 11:08 PM	--
Hadoop KaFaZhe	Dec 23, 2014, 4:53 PM	--
HBase	Jan 16, 2015, 11:57 PM	--
Information Retrieval	Jan 11, 2015, 2:06 PM	--
InternetOfThings	Dec 23, 2014, 6:26 PM	--
Linux Shell	Jan 20, 2015, 7:46 PM	--
Mahout	Jan 11, 2015, 2:05 PM	--
MapReduce Algorithms	Jan 11, 2015, 2:07 PM	--
NewSQL	Feb 22, 2015, 3:37 PM	--
NLP	Jan 18, 2015, 10:26 AM	--
Node.js	Jan 4, 2015, 11:07 AM	--
Pig/Hive	Feb 28, 2015, 2:13 PM	--
Programming	Feb 19, 2015, 10:37 PM	--
Python Data Analysis	Feb 14, 2015, 10:53 AM	--
R	Jan 11, 2015, 2:05 PM	--
Redis	Jan 2, 2015, 2:21 PM	--
REST API	Jan 11, 2015, 2:03 PM	--
Semantic Web	Jan 11, 2015, 2:05 PM	--
SocialWeb	Feb 14, 2015, 11:42 PM	--
Solr	Jan 4, 2015, 12:46 AM	--
Spark	Mar 20, 2015, 10:10 AM	--
Storm	Jan 4, 2015, 12:46 AM	--
Wiley.Data.Smart.Nov.2013.pdf	Jan 9, 2015, 2:28 PM	73.5 MB
References	Jan 4, 2015, 12:50 AM	--
Hadoop	Dec 23, 2014, 5:46 PM	--
In-Database	Dec 23, 2014, 4:53 PM	--
MapReduce	Dec 23, 2014, 5:50 PM	--
Calculating the Jaccard Similarity Coefficient with Map Reduce for Entity Pairs in Wikipedia.pdf	Jul 6, 2011, 6:46 PM	265 KB
Implementing best practices for fraud detection on an online advertising platform.pdf	Jul 6, 2011, 6:42 PM	1.1 MB
Map-Reduce for Machine Learning on Multicore.pdf	Dec 30, 2010, 3:22 PM	477 KB

Stay In Touch

- My Email shaohua.zhang@live.com
- My Blog http://www.dreamcre8or.com
  - working on my new blog
- LinkedIn http://ca.linkedin.com/in/shaohuazhang/en
- LinkedIn Group (Ryerson – Certificate in Big Data) https://www.linkedin.com/grp/home?gid=8258220
  - Thanks to Ali Kazim ☺