Advanced Recommender Systems

University of California, San Diego

Team Sharknado

Spring 2017

The Advisor

Julian McAuley

The Team

Julius

Alex Egg

Deepthi Mysore Nagaraj

Peyman Hesami

Lesson Learned

“ If you torture the data long enough, it will confess”

Ronald H. Coase

AND USE AN OPTIMAL TENSORFLOW OPTIMIZER

“ If you torture data long enough , it will confess”

Ronald H. Coase+Team Sharknado

Outline

- Introduction to Recommender Systems
- Data Collection/Preparation
- Modeling
- Evaluation Methodology
- Findings
- Scalability
- Demo

What is a Recommender System?

The age of search has come to an end...

- ...long live the age of recommendation!
- "We are leaving the age of information and entering the age of recommendation" - ***Chris Anderson in "The Long Tail"***
- "The Web, they say, is leaving the era of search and entering one of discovery. What' the difference? Search is what you do when you are looking for something. Discovery is when something wonderful that you did not know existed, or didn't know how to ask for, finds you." - ***CNN Money, "The race to create a 'smart' Google"***
- "Judging by Amazon's success, the recommendation system works. ***The company reported a 29% sales increase to $12.83 billion during its second fiscal quarter, up from $9.9 billion during the same time last year.*** A lot of that growth arguably has to do with the way Amazon has integrated recommendations into nearly every part of the purchasing process from product discovery to checkout." - ***Fortune, Amazon's recommendation secret***

Information Overload

- People read about 10MB worth of material a day, hear 400MB a day, and see 1MB of information every second -- ***The Economist***
- In 2015, consumption will rise to 74GB a day - ***UCSD Study 2014***
- Is having more choices a good idea? -- ***Paradox of Choic***e

The Value of Recommendations

- ***Netflix:*** 2/3 of the movies watched are recommended
- ***Google News:*** recommendations generate 38% more click through
- ***Amazon:*** 30% sales from recommendations
- ***Choicestream:*** 28% of the people would buy more music if they found what they liked

Problem Statement

Goal:

- Implement state of the art recommender system and improve its performance
    - Surface new content (Pigeon Hole Problem)
    - Recommend new items (Cold Start Problem)

Product:

- An advanced recommender system presented in the form of a web interface

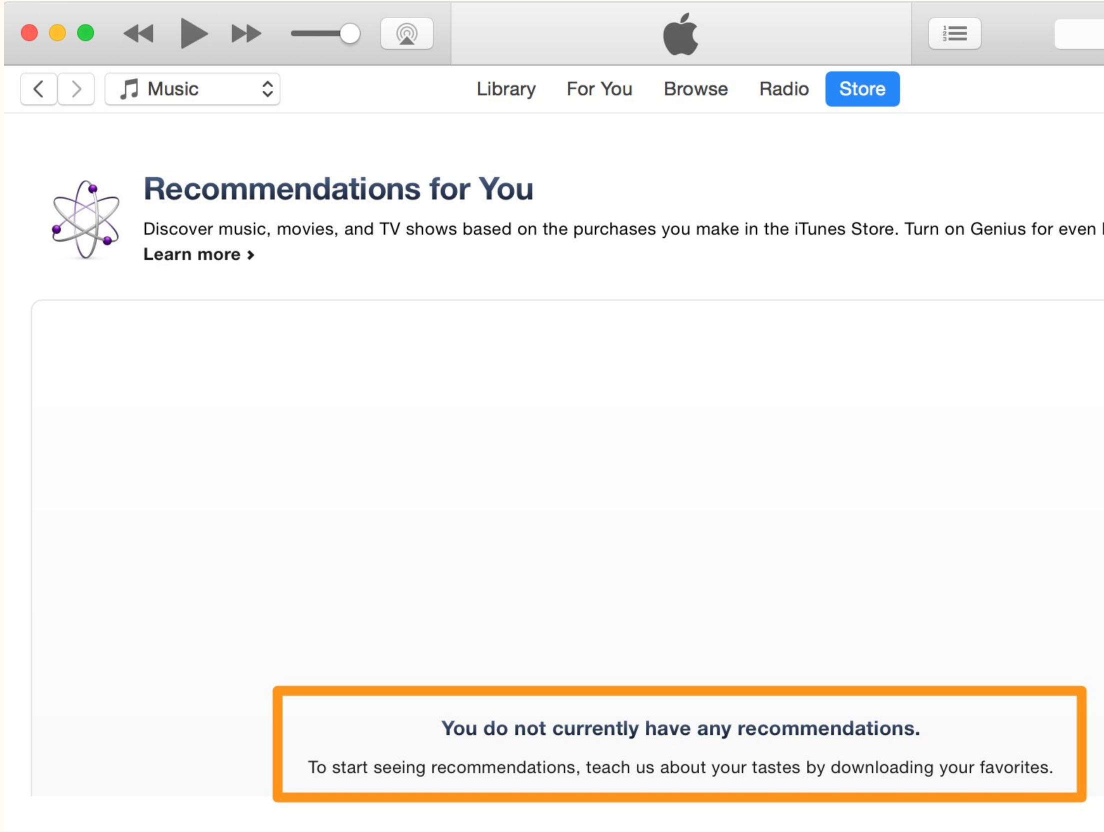
Cold Start Problem

New *users* w/ no ratings (right):

- Recommend popular items
- Start-up questions (e.g., ”tell me 10 songs/categories you love”)

New *items* w/ no ratings?

- Solicit ratings (focus group)
- **Profiling**: Suggest items w/ similar characteristics



Types of Recommendation Systems

Content-based filtering:

Consumer preferences for product attributes

- Users similarity distance measure
- Recommend objects w/ smallest distance measure

Pigeon Hole Problem: Does not surface original content

Cold-start Problem: Handles well

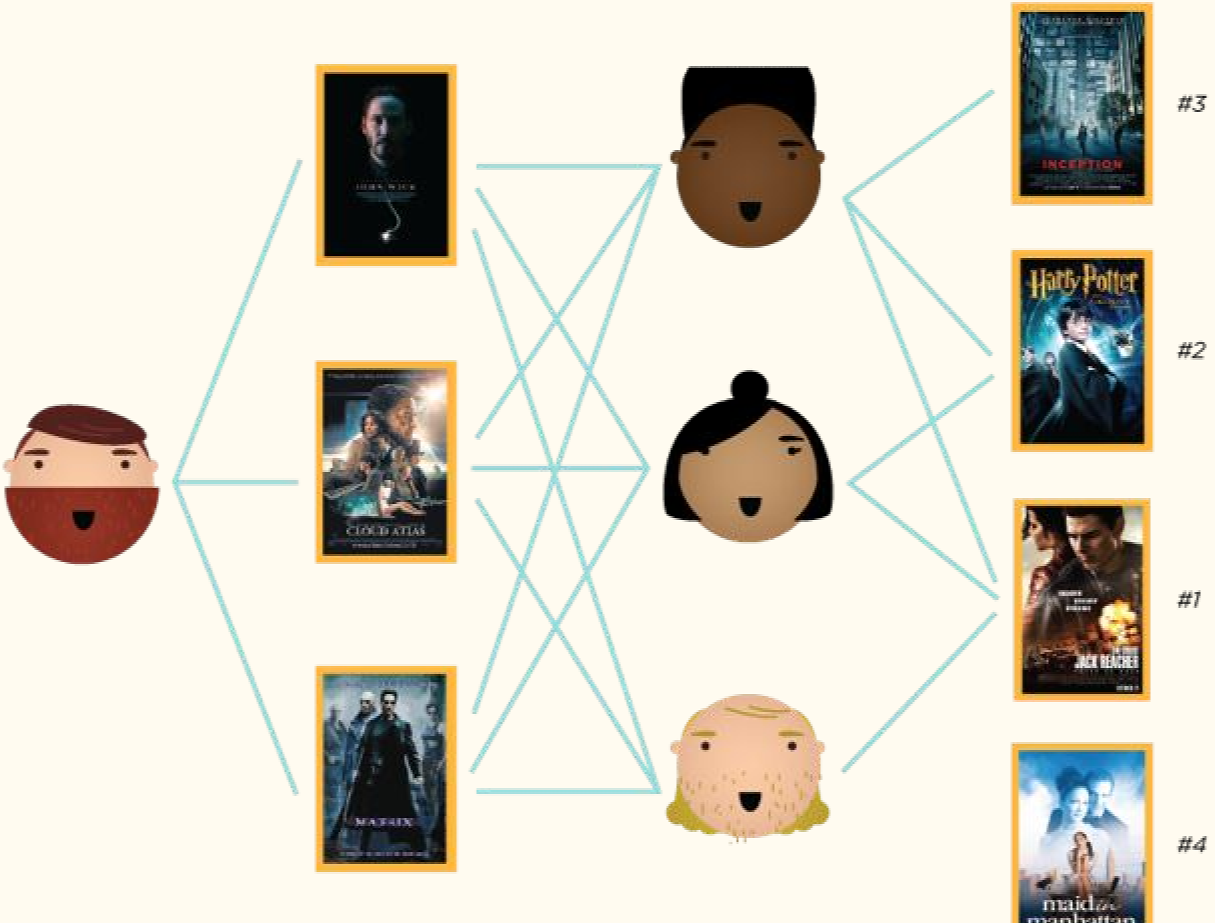
Collaborative filtering:

Mimics word-of-mouth based on analysis of rating/usage/sales data from many users

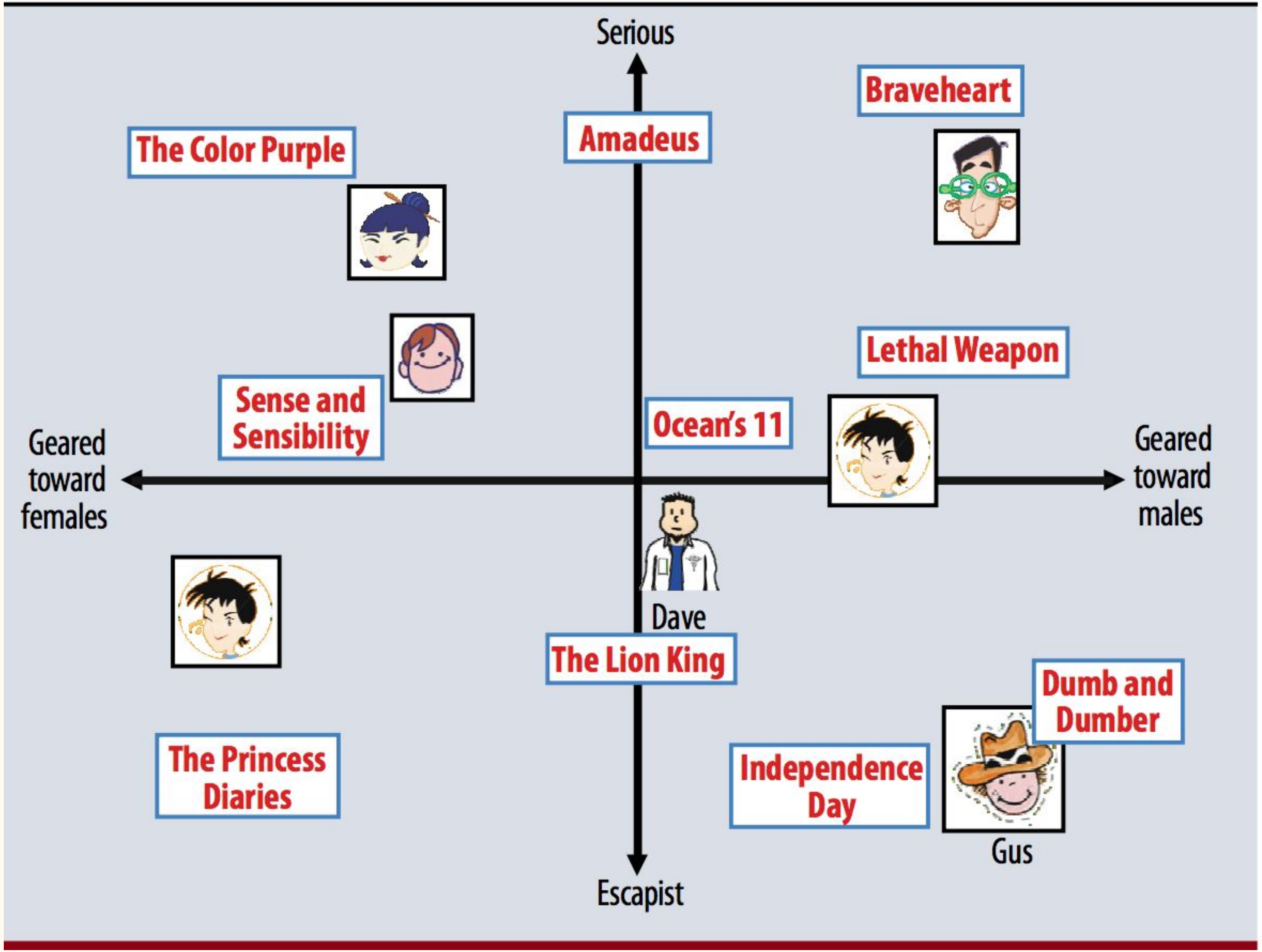
- Make predictions (filtering) about the interests of a user by collecting preferences or taste information from many other users (collaboration)
- Assumption: Those who agreed in the past tend to agree again in the future

Pigeon Hole Problem: handles well

Cold-start problem: consideration needed



Koren et al, "Matrix Factorization Techniques for Recommender System." IEEE Computer, 2009

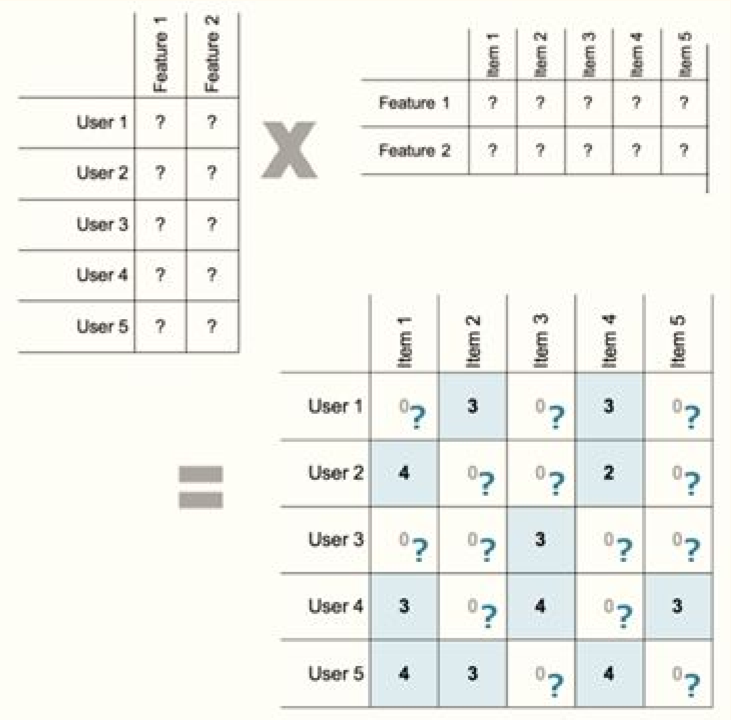

Serious
Braveheart
The Color Purple
Amadeus
Lethal Weapon
Sense and Sensibility
Ocean's 11
Geared toward females
Geared toward males
Dave
The Lion King
Dumb and Dumber
The Princess Diaries
Independence Day
Gus
Escapist

Collab. Filtering: Matrix Factorization

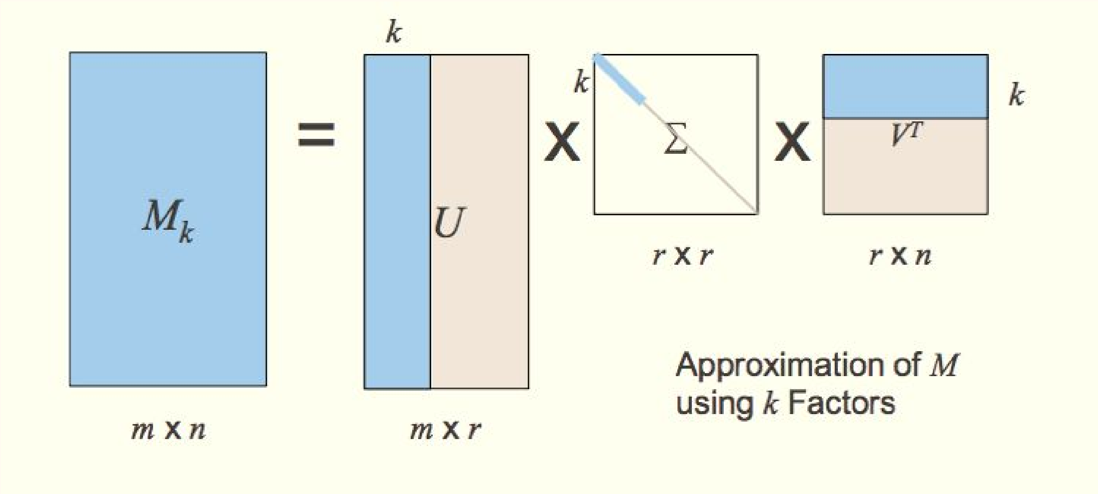
	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

$$x_u \in R^f$$
$$y_i \in R^f$$

$$\hat{r_{ui}} = x_u^T y_i$$



CF Model Based - Matrix Factorization (SVD)

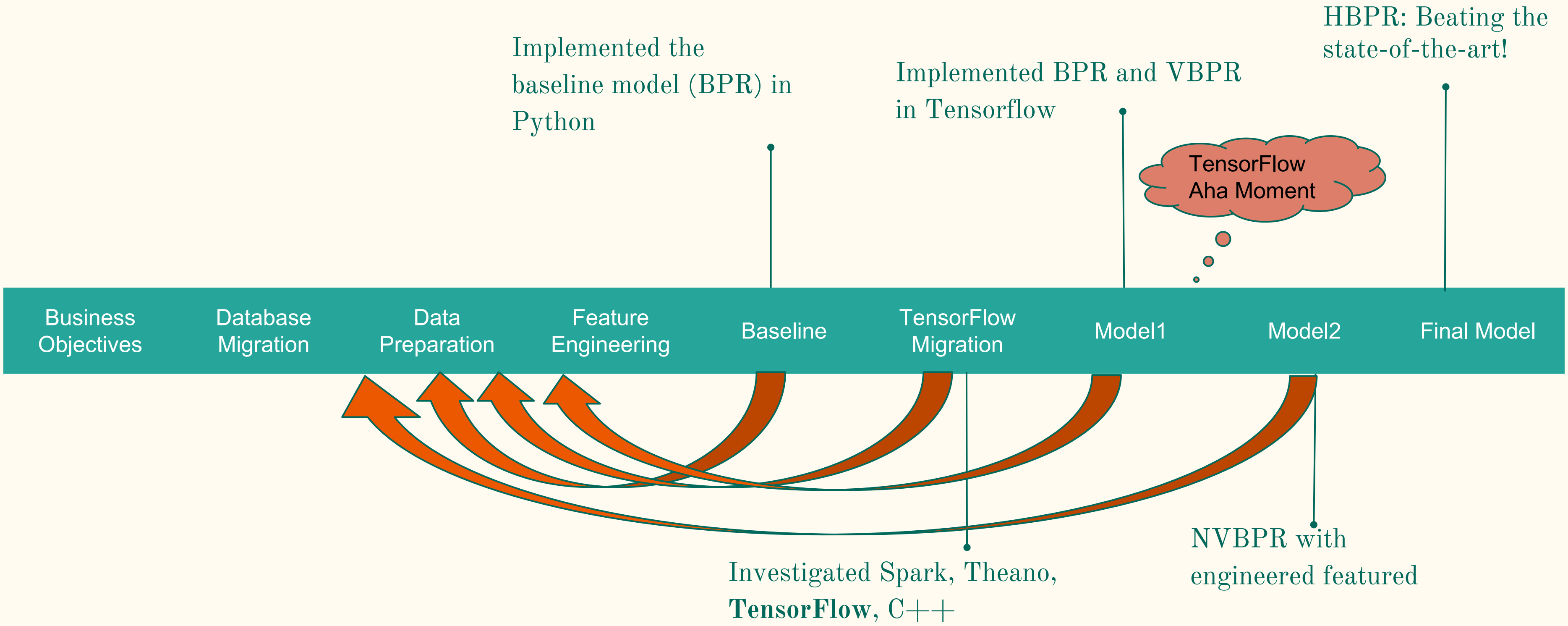


Inspiration is based on Matrix Factorization: Singular Value Decomposition

SVD: $R = X.Y^T$

If we can build R from U & V, then we can learn U & V

The Journey



Implemented the baseline model (BPR) in Python
Implemented BPR and VBPR in Tensorflow
HBPR: Beating the state-of-the-art!
TensorFlow Aha Moment
Business Objectives
Database Migration
Data Preparation
Feature Engineering
Baseline
TensorFlow Migration
Model1
Model2
Final Model
Investigated Spark, Theano, **TensorFlow**, C++
NVBPR with engineered featured

Data Collection and Management

Amazon Data Collection

1. ***142.8M Amazon product reviews*** and metadata from May 1996 - July 2014 in a json format (Across 24 product categories) - Over 2 TB!

2. ***Reviews*** (ratings, text, helpfulness votes)

3. ***Product metadata*** (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs)

4. ***Visual features*** were extracted from each product image using a deep CNN. Image features are stored in a binary format, which consists of 10 characters (the product ID), followed by 4096 floats (repeated for every product)

Example: Uranium Ore

26,481 of 27,017 people found the following review helpful

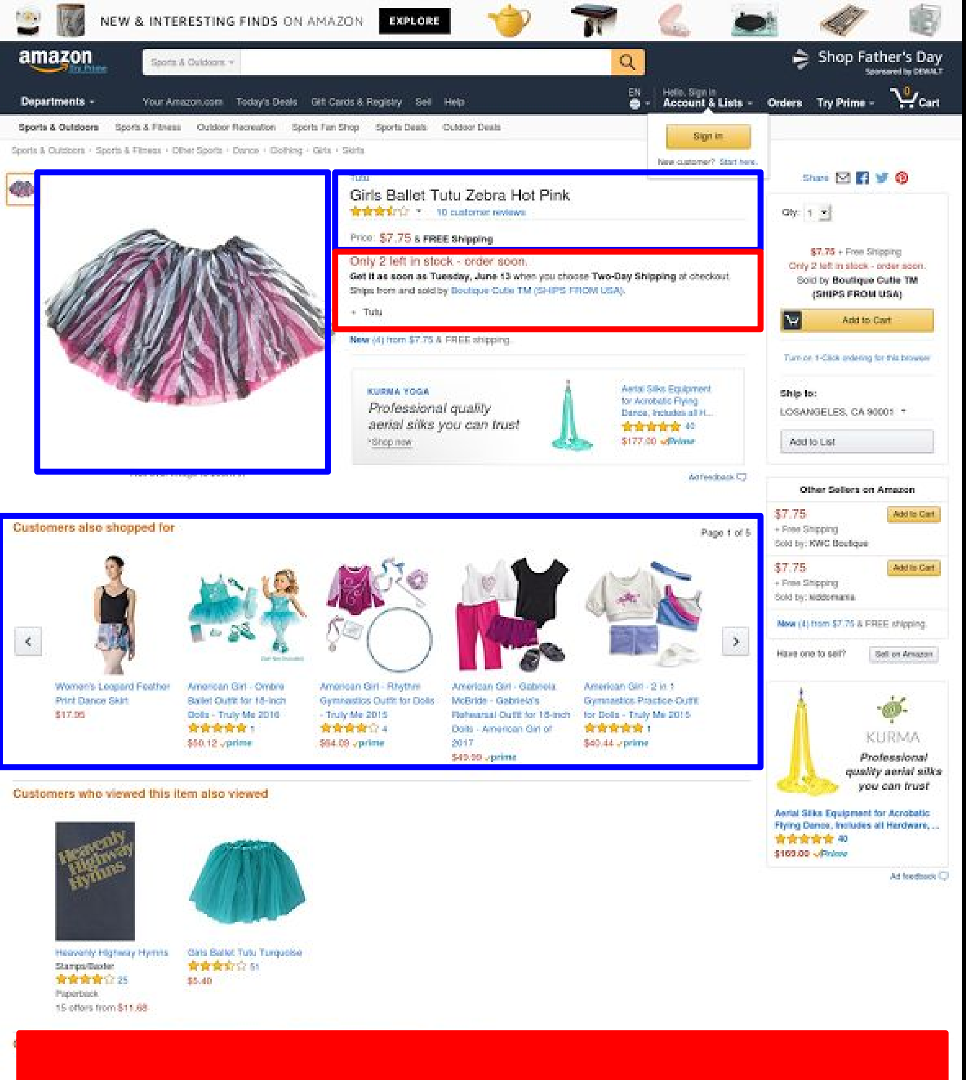
Great Product, Poor Packaging, May 14, 2009

By Patrick J. McGovern

This review is from: **Uranium Ore**

I purchased this product 4.47 Billion Years ago and when I opened it today, it was half empty.


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "Patrick J. McGovern",
  "helpful": [
        26481
        27017
  ],
  "reviewText": "I purchased this product 4.47 Billion Years ago and when I opened it today, it was half empty",
  "overall": 5,
  "summary": "Great Product, Poor Packaging",
  "unixReviewTime": 1242317373,
  "reviewTime": "05 14, 2009"
}
```



{
"asin": "0000031852",
"title": "Girls Ballet Tutu Zebra Hot Pink",
"price": 3.17,
"imUrl":
"http://ecx.images-amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
"related": {
"also_bought": [
"B00JHONN1S",
"B002BZX8Z6",
"B00D2K1M3O",
"0000031909",
"B00613WDTQ",
"B00D0WDS9A"
],
"also_viewed": [
"B002BZX8Z6",
"B00JHONN1S",
"B008F0SU0Y",
"B00D23MC6W",
"B00AFDOPDA",
"B00E1YRI4C",
"B002GZGI4E",
"B003AVKOP2"
],
"bought_together": [
"B002BZX8Z6"
]
},
"salesRank": {
"Toys & Games": 211836
},
"brand": "Coxlures",
"categories": [
[
"Sports & Outdoors",

Data and Code Management Tools

- Google Team Drive - Unlimited Storage!
- Gdrive - open source CLI for Google Drive
- GitHub - source control

Data Cleaning

Data Cleaning (Cntd..)

1. Duplicate Reviews
   a. Amazon merges the near-identical product reviews which resulted in duplicate reviews which deduped before using. (Eg: VHS and DVD versions of the same movie)
2. Missing Values and New Feature via Web Scraping
   a. Price - 62% missing
   b. Brand - 89% missing
   c. Product Title -15% Missing
   d. Descriptive Text Missing
3. Multiple Prices?
   a. Use Average

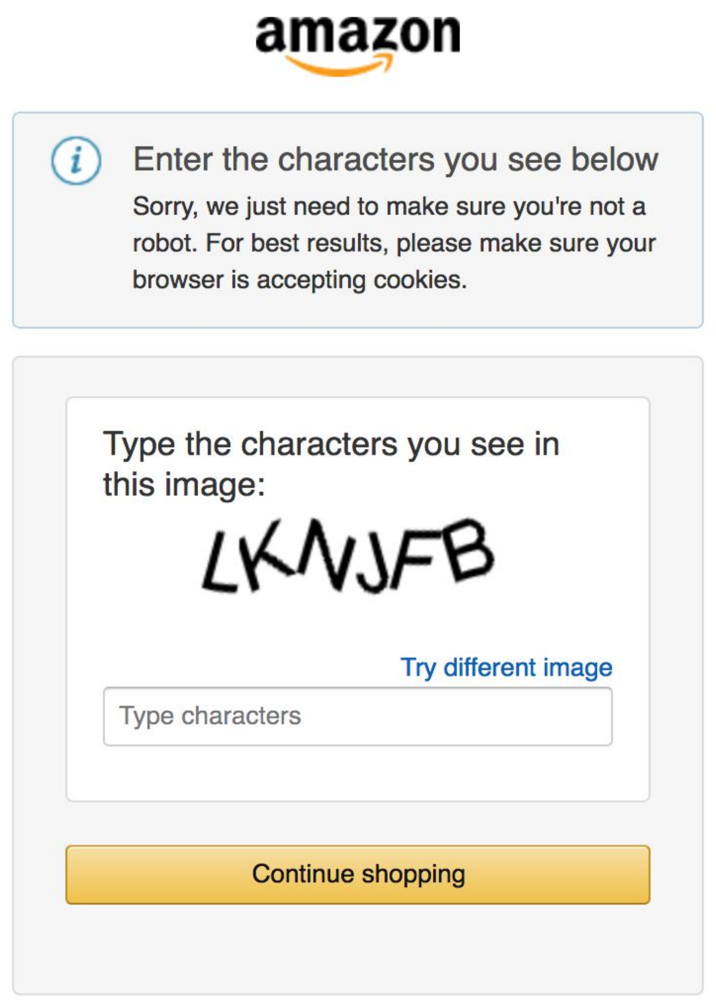
Web Scraping - Challenges

Why?

- Data Sparsity - Missing Data
  - Price, Brand, Title
- New Features: Description, Features, Specs.

Challenges:

- Old code doesn't work anymore
  - CAPTCHA, not captcha!
- IP Bans or Worse ...
- Page Variation
- Time



Web Scraping - Setup

Libraries:

- Scrapy - web scraping framework
- Lxml - XML/HTML/XHTML parser with
  full XPATH support

Anonymization Techniques:

- Public Proxies Pooling (Over 200 Servers)
- Immutable Cookies
- Dynamic User Agent modeling

User Agent Strings

Standard defined by RFC2616, sent with every request

Used to track web browsers for traffic statistics

- **Ua_type:** Desktop
- **Os_name:** macOS
- **Os_version:** 10.12.5
- **Browser_name:** Safari
- **Browser_version:** 10.1.1
- **Engine_name:** WebKit

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/603.2.4 (KHTML, like Gecko) Version/10.1.1
Safari/603.2.4
```

Web Scraping - User Agent Stats - w3schools

2017	Chrome	IE/Edge	Firefox	Safari	Opera
April	75.7 %	4.6 %	13.6 %	3.7 %	1.1 %
March	75.1 %	4.8 %	14.1 %	3.6 %	1.0 %

2017	Win10	Win8	Win7	Vista	WinXP	Linux	Mac	Chrome OS	Mobile
April	34.3%	10.1%	31.9%	0.2%	0.8%	5.5%	10.8%	0.2%	6.3%
March	33.1%	10.2%	33.2%	0.2%	0.9%	5.5%	10.6%	0.2%	6.1%

Web Scraping - Did it work?

Improvement:

- Missing Price: **61.71% → 23.47%**
- Missing Brand: **88.83% → 9.35 %**
- **But actually 100%**

The SECRET formula:

$$B = N(\mu,\sigma^2)$$

New Features:

- Product Description, Bulleted Features, Information, Specifications, etc
- ... All the pages were saved!

Exploratory analysis and Feature Engineering

Exploratory Data Analysis

Terminology:

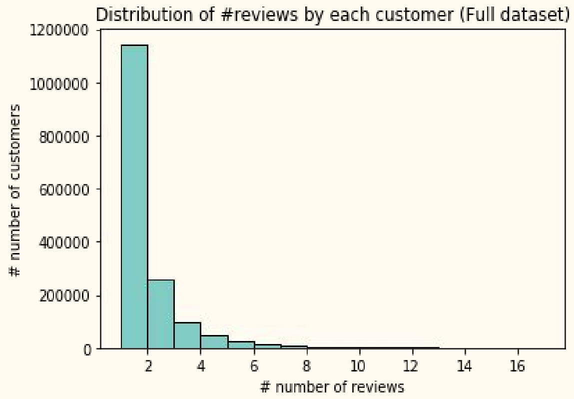
5-core Data set

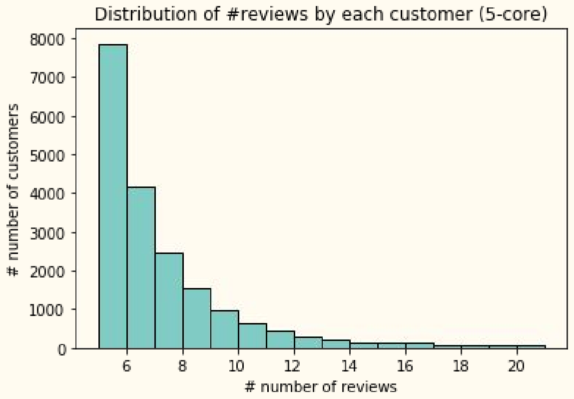
- Users and items with 5 reviews each. (5% of the total data)
- To reduce sparsity and computational overhead

Full Data set

- All users and items available in the scraped data

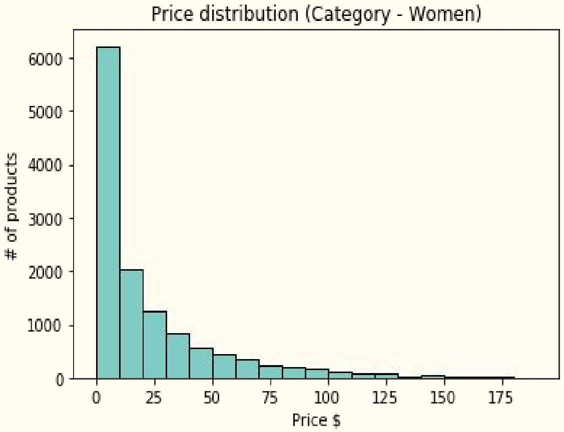
Note: We used Women's Clothing and Cell phones



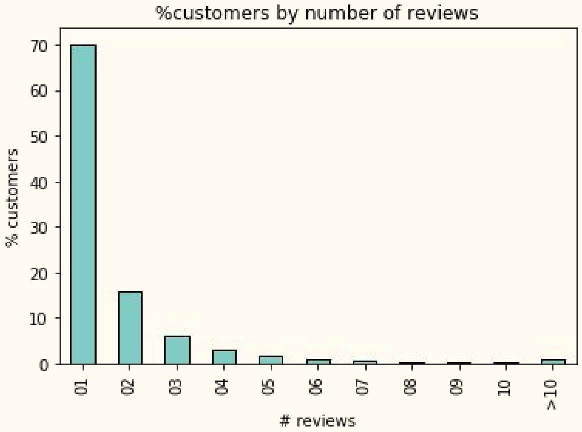


These graphs are for Women's clothing category

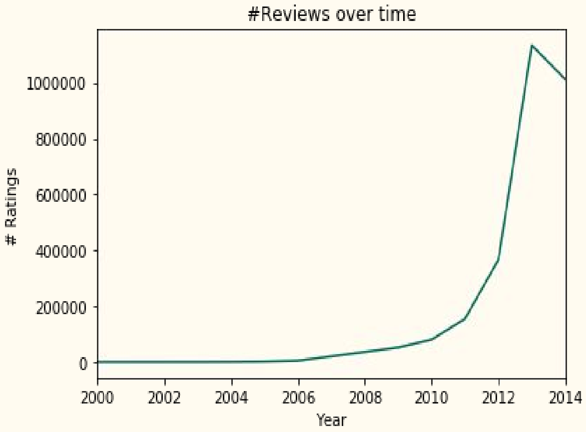
Exploratory data analysis.. cntd..



Most products are priced <$100



Most reviews are from customers reviewing only one product



Most reviews are after 2010

Feature Engineering

In many categories such as ‘Books’ or ‘Cell Phones’ visual features might not have a huge impact. So we explored a few non-visual features which might boost the performance even when visual features don’t seem to work.

1. Price Features
2. Brand Features
3. Product Description Features

Price features

Price range for Women's category varies from 0 to 100

Price was quantized and 10 buckets were created

Item	Price <10	Price 11 to 20	Price 21 to 30	Price 31 to 40	Price 41 to 50	Price 51 to 60	Price 61 to 70	Price 71 to 80	Price 81 to 90	Price >91
1	0	1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0
n	0	0	1	0	0	0	0	0	0	0

Other ways tried:

1. Directly include price as a feature
2. Normalized feature

Brand Features

There are about 1992 brands in the 5-core dataset

Brands were included as a binary vector

Item	Crazy for Bargains	Serenity Crystals	Mango	DKNY	New Balance	Reebok	Nike	Bebe	Adidas	Swatch	…….
1	0	1	0	0	0	0	0	0	0	0	
2	0	0	0	1	0	0	0	0	0	0	
	0	0	0	0	0	0	1	0	0	0	
n	1	0	0	0	0	0	0	0	0	0	

Other ways tried:

1. Frequency of brand purchased by the user (instead of binary vector)

Product Features

Product Title was used to extract features

Based on the analysis of frequency of n-grams, 2-grams were found to be most meaningful

A binary vector of length 4525 was included

Bi-grams	Term frequency
sterling silver	963
plus size	331
long sleeve	245
running shoe	231
stainless steel	230
pendant necklace	227
stud earrings	211
cubic zirconia	197
925 sterling	150
sandal black	144

Item	Sterling silver	Plus size	Long sleeve	Running shoe	Stainless steel	Pendant necklace	Stud earrings	Cubic zirconia	925 sterling	Sandal black	..
1	0	1	0	0	0	0	0	0	0	0	
2	0	0	0	1	0	0	0	0	0	0	
	0	0	0	0	0	0	1	0	0	0	
n	1	0	0	0	0	0	0	0	0	0	

Modeling Recommender System

Matrix Factorization Model

Model ranking of user's preference (rating) to products as :

$$x_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

$$R = \left.\underbrace{\begin{pmatrix} 5 & 3 & \cdots & \cdot \\ 4 & 2 & & 1 \\ 3 & \cdot & & 3 \\ \cdot & 2 & & 4 \\ 1 & 5 & & \cdot \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & \cdot \end{pmatrix}}_{\text{items}}\right\} \text{users}$$

Where:

α -- popularity of item

β_u -- user tendency to rate things above the mean

β_i -- item tendency to receive higher ratings than others

$\gamma_u \cdot \gamma_i$ -- 'compatibility' between the user u and the item i

Bayesian Personalized Ranking (**BPR**) to optimize it

Advanced BPR- Visual BPR (VBPR)

- Base model:

$$x_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

- Model Extension:

$$x_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i + \theta_u^T \theta_i + \beta'^T f_i$$

Image features: FC7 of pre-trained CNN (5C and 3FC) on 1.2 million images (ILSVRC2010)

$\theta_u^T(\mathbf{E}\, f_i)$ -- 'compatibility' between the user u and the visual features of item i

$\beta'^T f_i$ -- users' overall opinion toward the visual appearance of item i

Advanced BPR- NonVisual BPR (NVBPR)

- Base model:

$$x_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$$

- Model Extension:

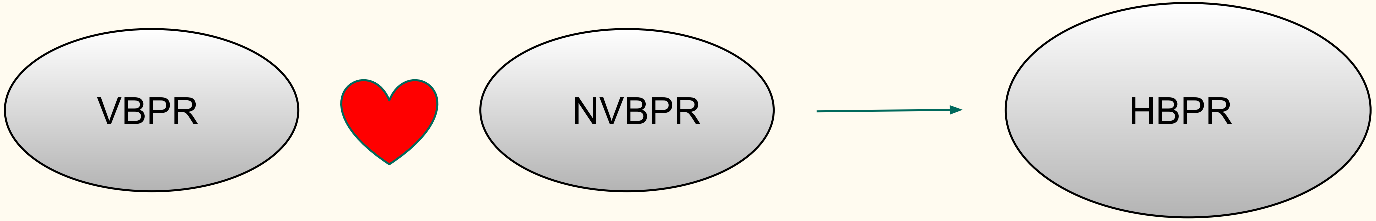
$$x_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i + \theta_u{}^T(\mathbf{E}\, f_i) + \beta'^T f_i$$

Engineered Price, Brand, and Product Description Features

$\theta_u{}^T(\mathbf{E}\, f_i)$ -- 'compatibility' between the user u and the visual features of item i

$\beta'^T f_i$ -- users' overall opinion toward the visual appearance of item i

Advanced BPR- Hybrid BPR (HBPR)

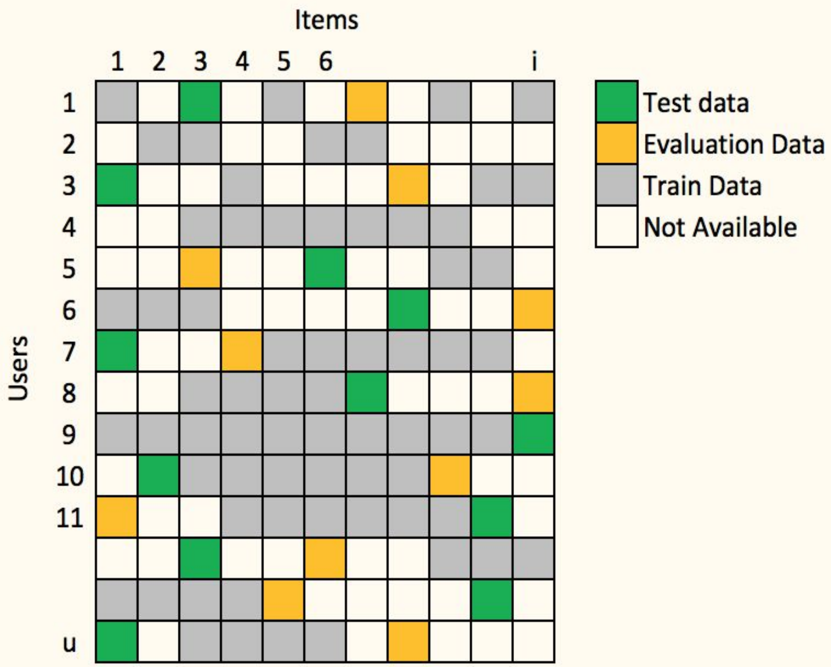


Performance and Evaluation

Evaluation Metric-- Area Under the Curve (AUC)

$$AUC = \frac{1}{|\mathcal{U}|}\sum_{u}\frac{1}{|E(u)|}\sum_{(i,j)\in E(u)}\delta(\widehat{x}_{u,i} > \widehat{x}_{u,j})$$

$$E(u) = \{(i,j) | (u,i) \in \mathcal{T}_u \wedge (u,j) \notin (\mathcal{P}_u \cup \mathcal{V}_u \cup \mathcal{T}_u)\}$$

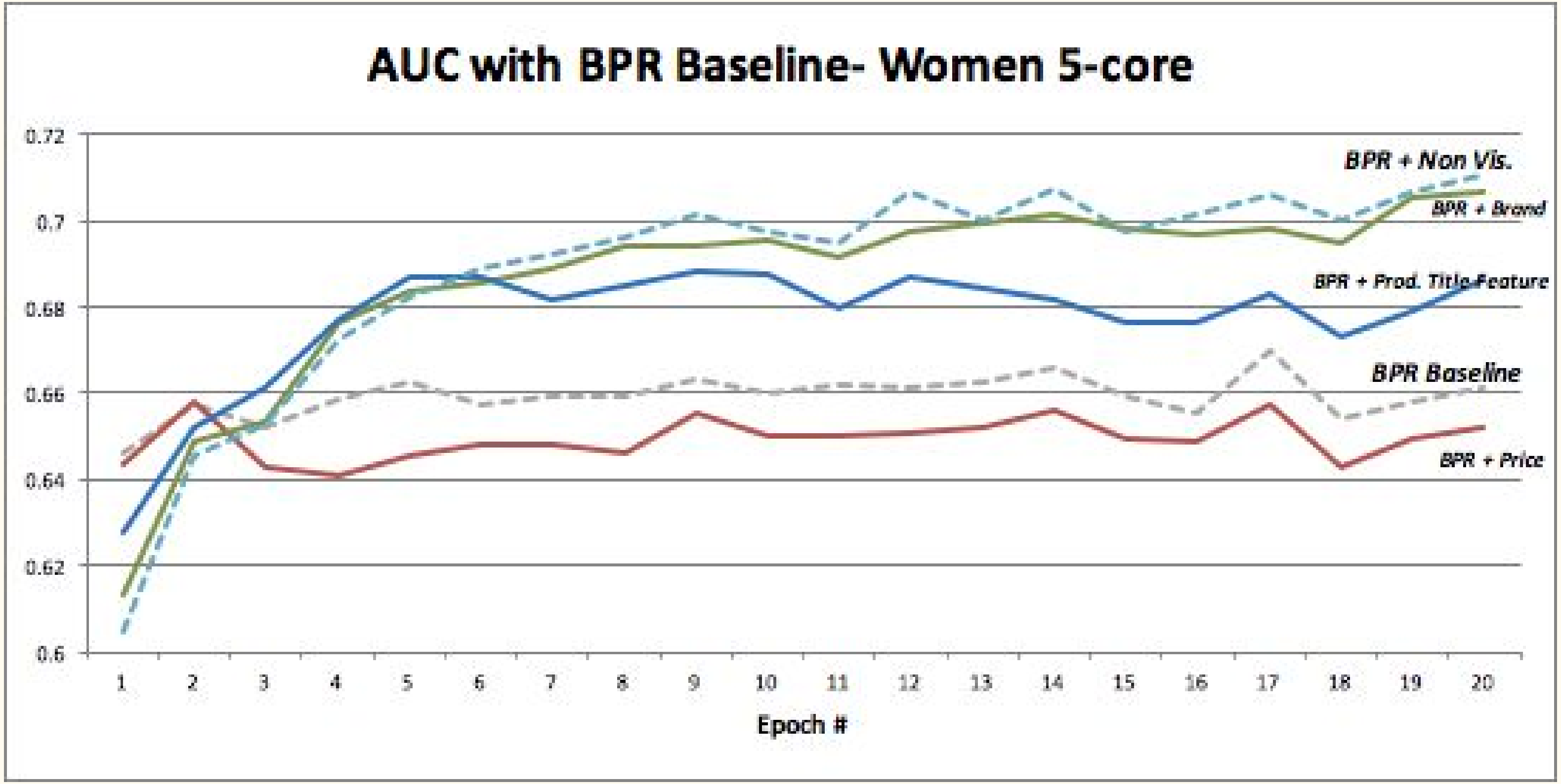


Findings

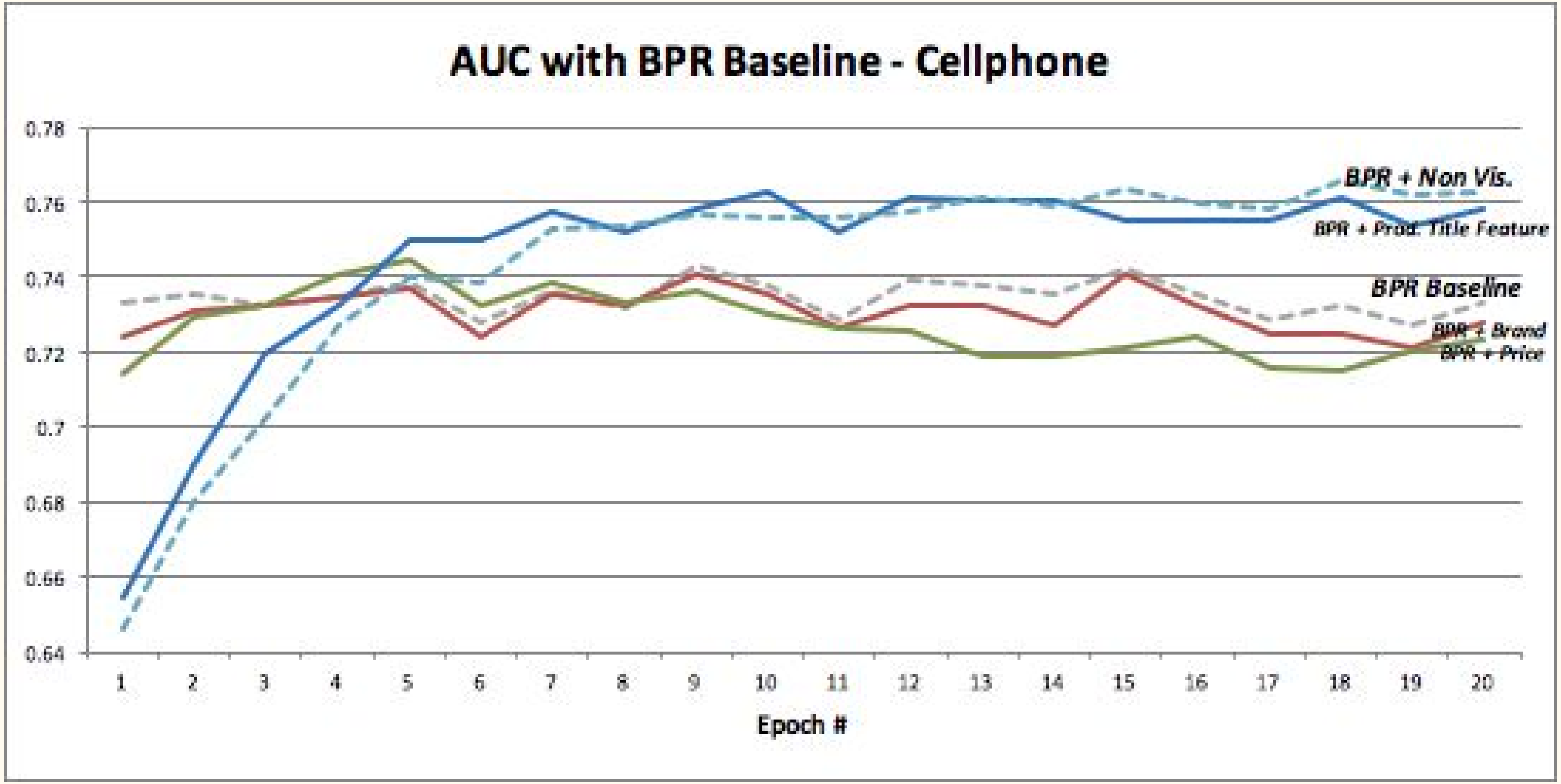
9%

Improvement in the quality of the baseline recommender system

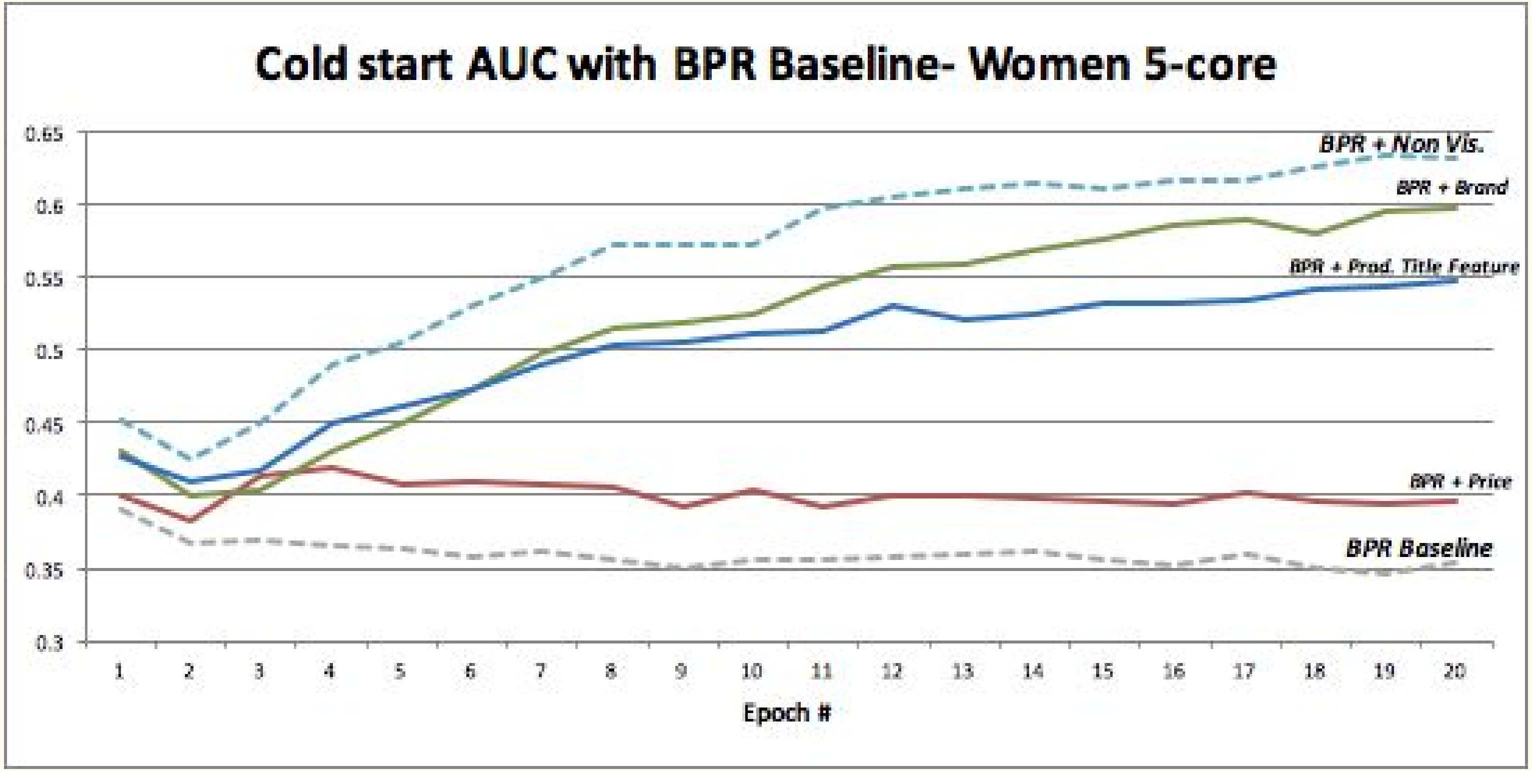
NVBPR


AUC with BPR Baseline- Women 5-core
0.72
0.7
0.68
0.66
0.64
0.62
0.6
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Epoch #
BPR + Non Vis.
BPR + Brand
BPR + Prod. Title Feature
BPR Baseline
BPR + Price

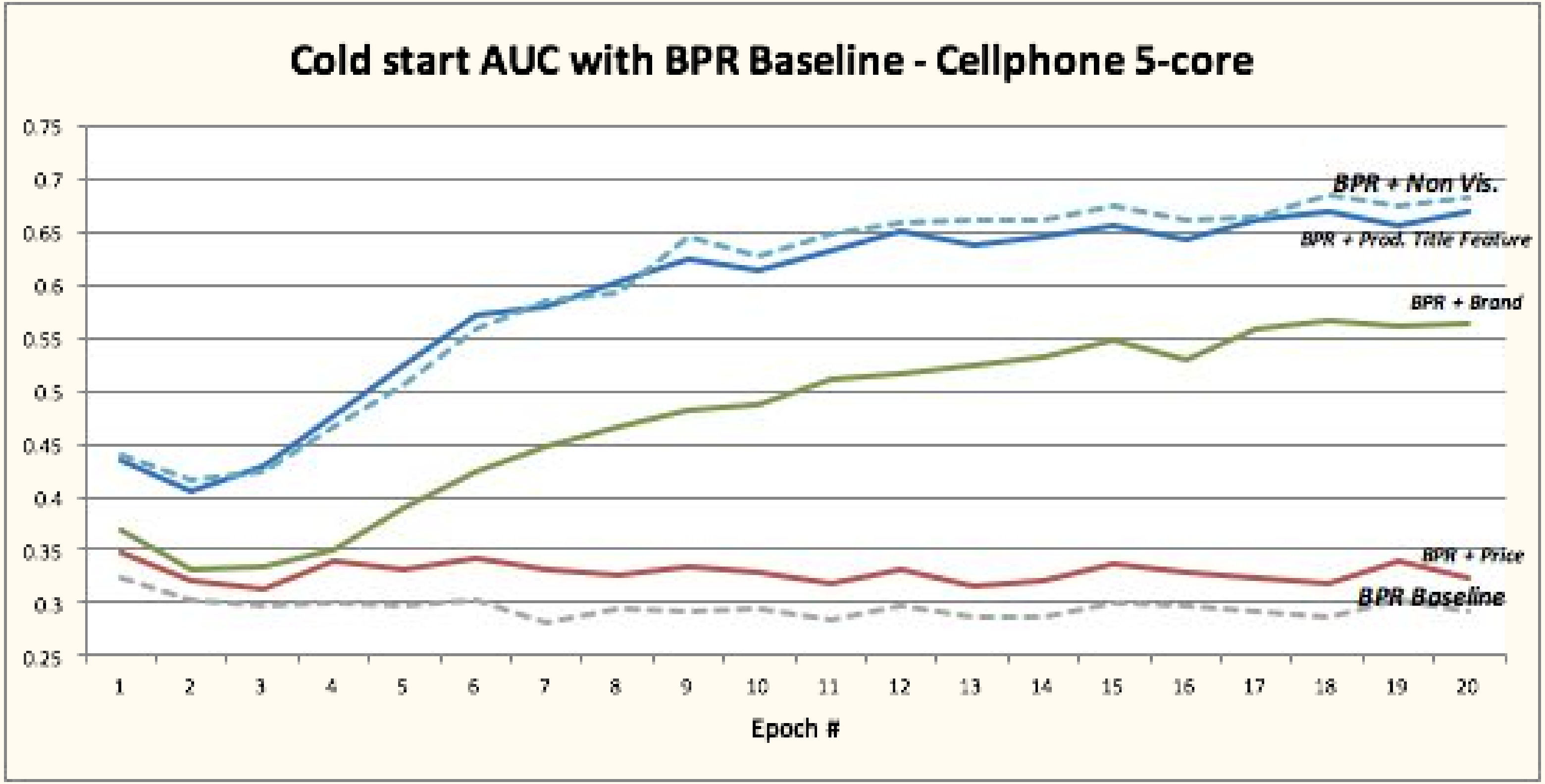
NVBPR


AUC with BPR Baseline - Cellphone
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Epoch #
BPR + Non Vis.
BPR + Prod. Title Feature
BPR Baseline
BPR + Brand
BPR + Price

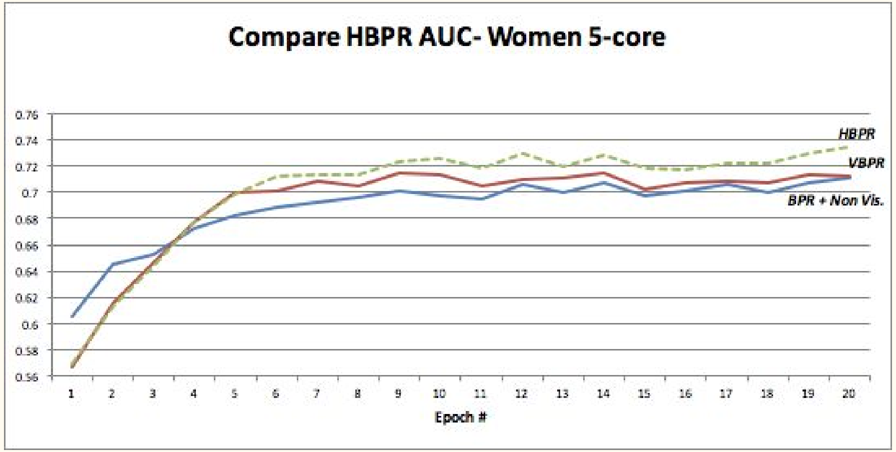
NVBPR - Cold Start

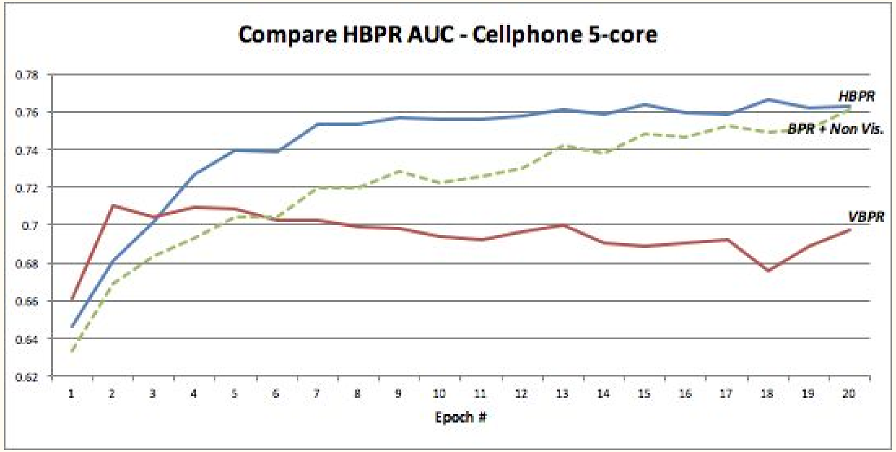


NVBPR - Cold Start

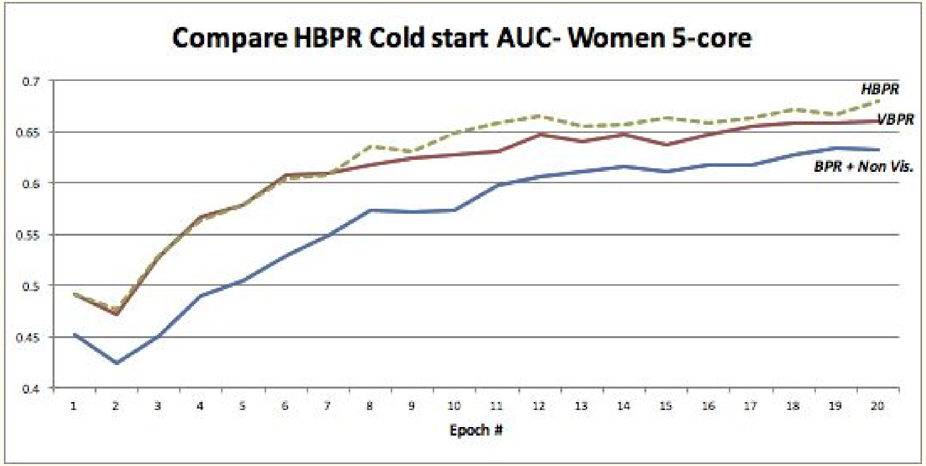


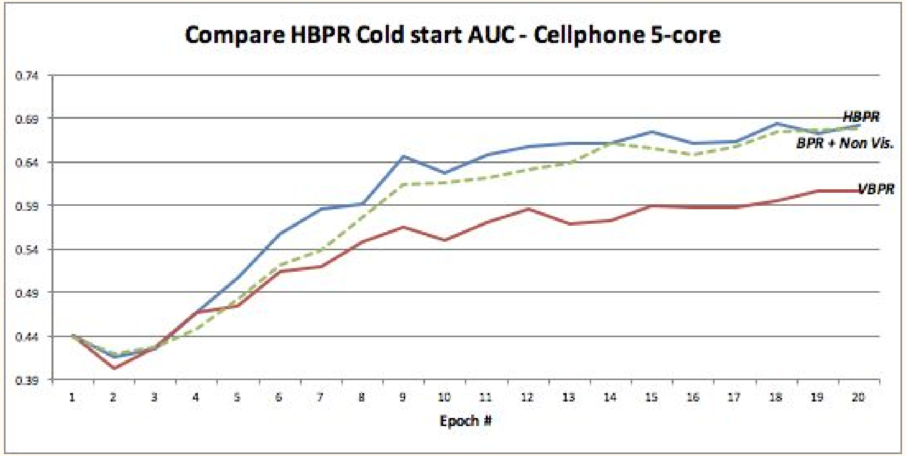
HBPR


Compare HBPR AUC- Women 5-core
HBPR
VBPR
BPR + Non Vis.
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Epoch #


Compare HBPR AUC - Cellphone 5-core
HBPR
BPR + Non Vis.
VBPR
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Epoch #

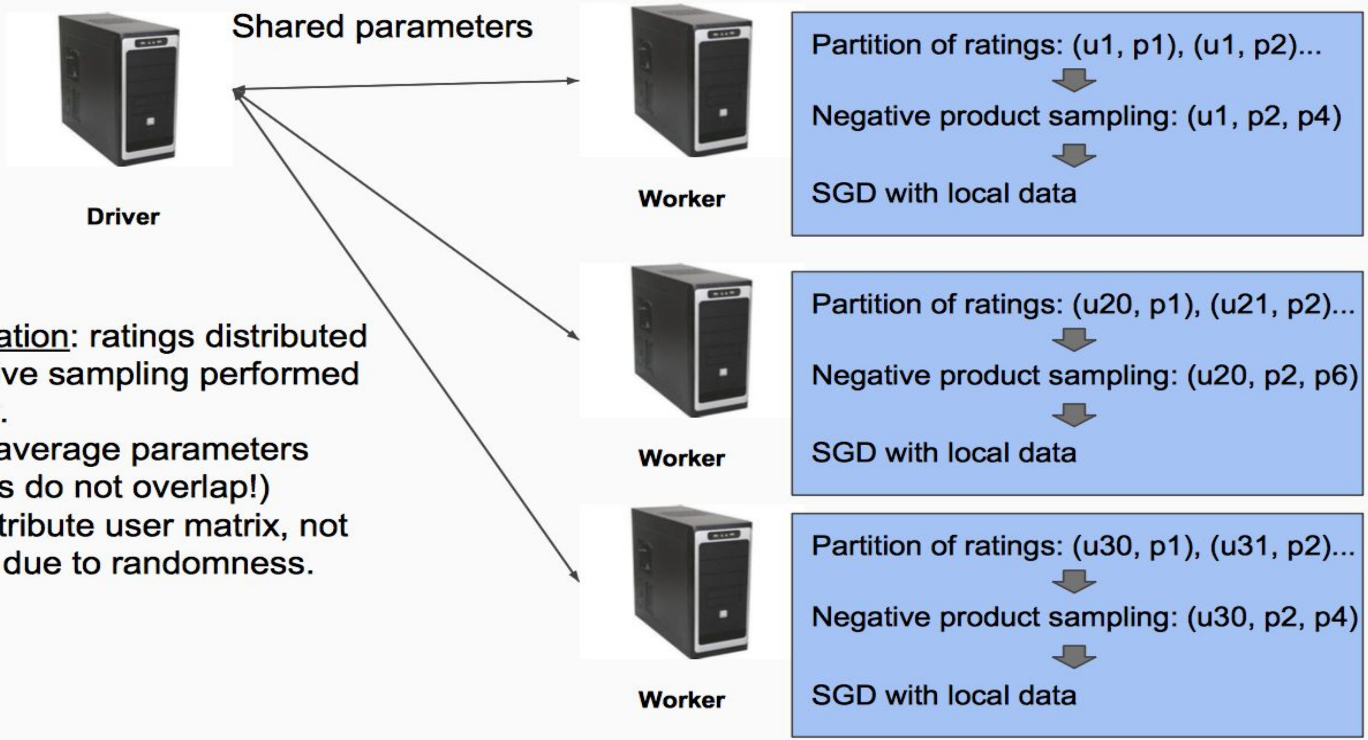
HBPR- Cold Start





Scalability

Spark



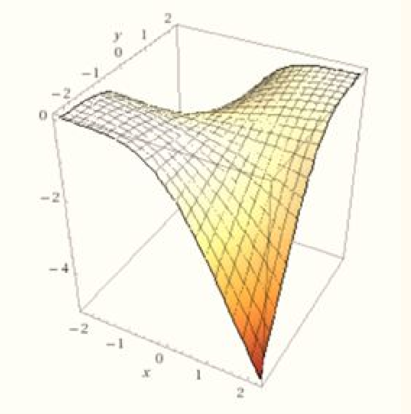
- Data parallelization: ratings distributed by user, negative sampling performed in each worker.
- Parallel SGD: average parameters (note that users do not overlap!)
- Possible to distribute user matrix, not product matrix due to randomness.

TensorFlow

- Auto differentiation
  - Removes SGD boilerplate
  - gradient computations

TensorFlow
Aha Moment

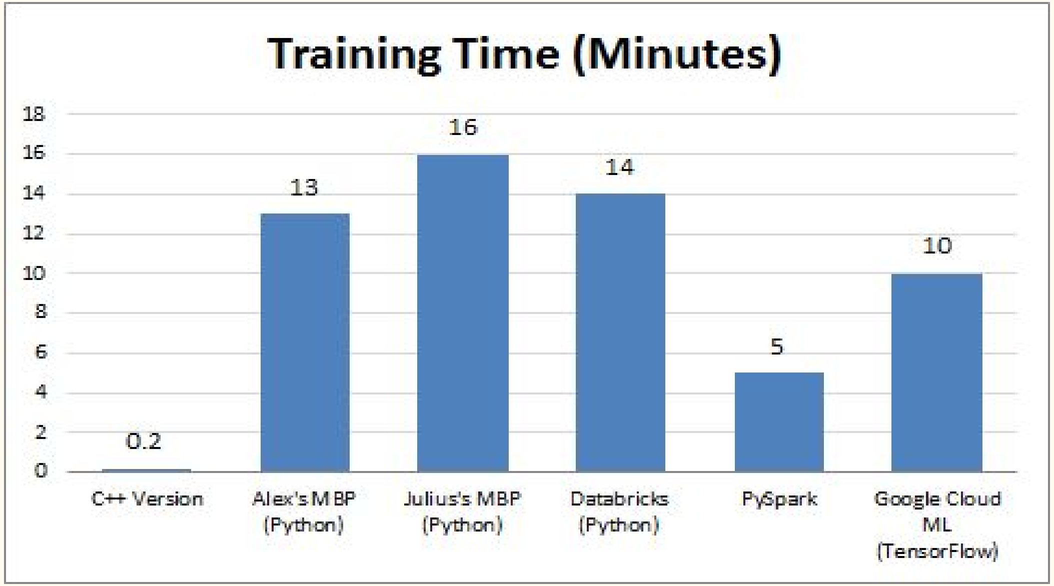
```
bprloss = regulation_rate * l2_norm - tf.reduce_mean(tf.log(tf.sigmoid(x)))

optimizer = tf.train.GradientDescentOptimizer(learning_rate)
train_op = optimizer.minimize(bprloss)
```



- Parallel matrix multiplications (GPU)
  - Speedup when using mini-batch SGD

Computation Cost Analysis

- PySpark and TensorFlow were the winners!
- TensorFlow was the most feasible!



Demo

Business Value

10% improvement→$1M

31337
lines of chat

20 hours of
Video Chat

10k lines of
codes

10k lines of
codes

Beating the
state-of-the-art

Business Value

Conclusions

- Visual aspects of items bias users' opinion toward them in some categories
- Non-visual aspects of items bias users' opinion toward them in other categories
- Combination of these two can help build a performant and generic recommender system
- Scalability is crucial in recommender systems

Future work

- Tuning the current model
    - # of quantized levels of price
    - # of elements to use from product description
    - Use purchase frequency of a brand for a user
- Incorporate item grouping (clustering similar items)
- Temporal dynamics to capture drifting price/fashion tastes over time

Thank You!