Advanced Machine Learning

Intro, Recommendation Systems

Yannet Interian
yinterian@usfca.edu

Outline

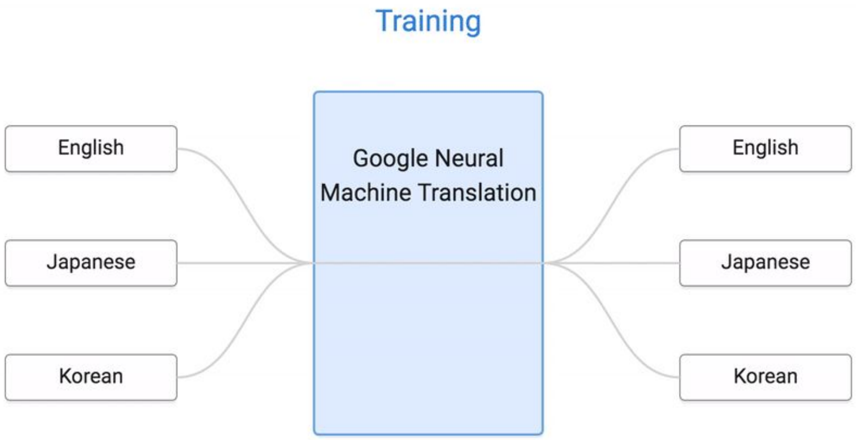
- Trends in ML
- ML in industry
- Admin stuff
  - TA, Content, Textbook, Work, Schedule, Final project, Lecture style
- Homework 1
- Recommendation Systems
  - Content Based Recommendations
  - Case Study: YouTube

Trends in ML: Deep Learning

- superhuman Go playing,
- superhuman speech transcription,
- superhuman translation,
- superhuman lip reading

http://blog.evjang.com/2017/01/nips2016.html

Google Translate (deep learning)



Zero-shot translation:

- single system to translate between multiple languages
- translation between language pairs never seen explicitly by the system
- the system to transfer the *translation knowledge* from one language pair to the others

https://research.googleblog.com/2016/11/zero-shot-translation-with-googles.html

Trends in ML: Transfer Learning

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned.

Transfer learning domain adaptation and semi-supervised learning alleviate the data-hungry requirements of deep learning, and are starting to work really well.

http://blog.evjang.com/2017/01/nips2016.html, http://cs231n.github.io/transfer-learning/

Trends in Applied ML

- What methods win Kaggle competitions?
  a. Gradient tree boosting(especially XGBoost)
  b. deep neural nets (especially convolutional nets for images and RNNs for some time series problems).
- Ensembles add 2–5% in performance over the best individual methods
  a. but also lead to more complex systems, so are often not worth it in practice.

https://blog.ought.com/nips-2016-875bb8fadb8c#.f5rax3mdv

ML in Industry: Clickthrough rate (CTR)

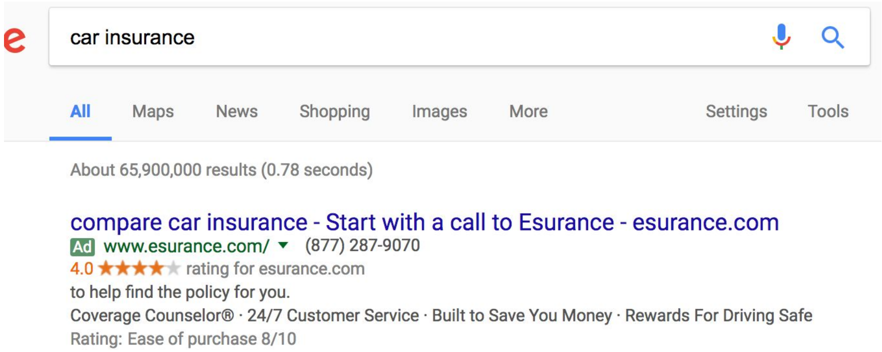
A ratio showing how often people who see your ad end up clicking it.

- CTR is the number of clicks that an ad receives divided by the number of times your ad is shown: clicks ÷ impressions = CTR. For example, if you had 5 clicks and 1000 impressions, then your CTR would be 0.5%.

https://support.google.com/adwords/answer/2615875?hl=en

ML in Industry:

Predicting Clickthrough rate (CTR)



A ratio showing how often people who see your ad end up clicking it.

- CTR number of clicks / the number of times your ad is shown:
- For example, if you had 5 clicks and 1000 impressions, then your CTR = 0.5%.

https://support.google.com/adwords/answer/2615875?hl=en

ML in Industry:

Predicting Clickthrough rate (CTR)

Who is solving this problem?

- Google
- Vungle (mobile advertising)
- Any search engine with advertising (Bing, Baidu)

https://support.google.com/adwords/answer/2615875?hl=en

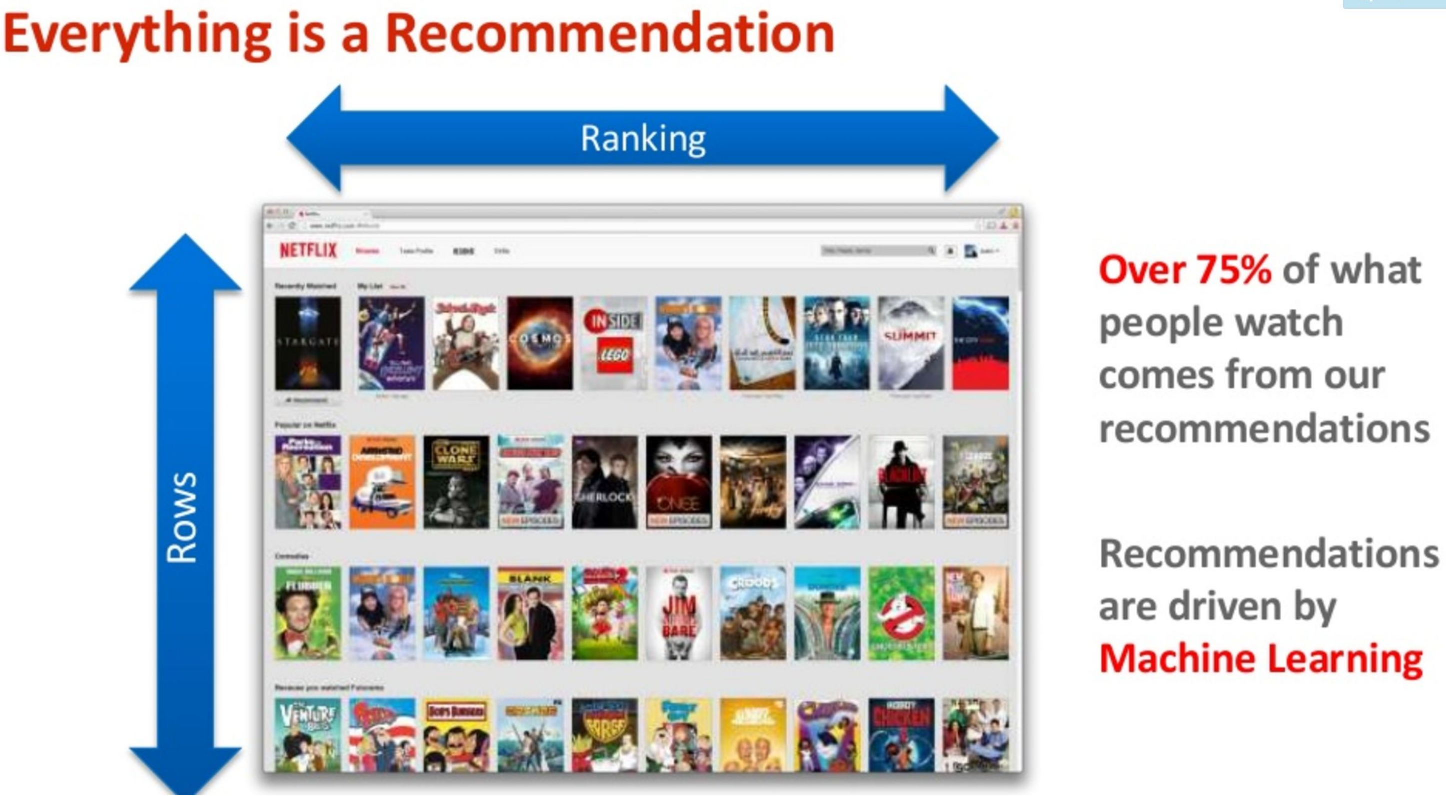
ML in Industry: Recommender systems

Recommendation systems: web applications that involve predicting user responses to options.

ML in Industry: Recommender systems

http://www.slideshare.net/justinbasilico/past-present-future-of-recommender-systems-an-industry-perspective

ML at Netflix



http://www.slideshare.net/justinbasilico/lessons-learned-from-building-machine-learning-software-at-netflix

ML at Netflix

Models & Algorithms

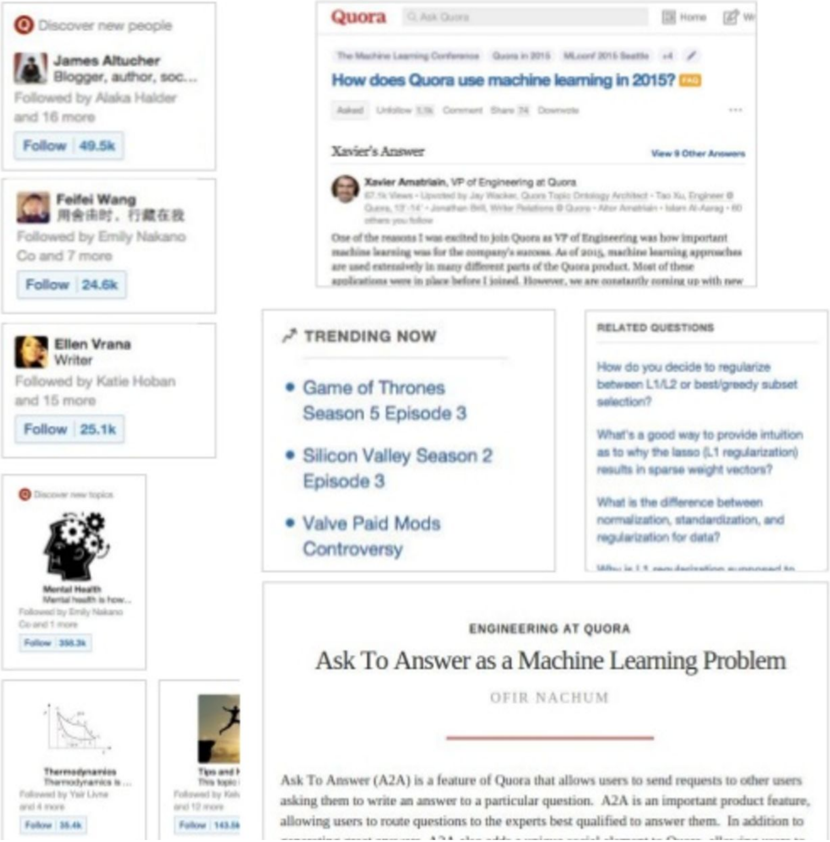
- Regression (Linear, logistic, elastic net)
- SVD and other Matrix Factorizations
- Factorization Machines
- Restricted Boltzmann Machines
- Deep Neural Networks
- Markov Models and Graph Algorithms
- Clustering
- Latent Dirichlet Allocation
- Gradient Boosted Decision Trees/Random Forests
- Gaussian Processes
- …

http://www.slideshare.net/justinbasilico/lessons-learned-from-building-machine-learning-software-at-netflix

ML Applications @ Quora

Quora

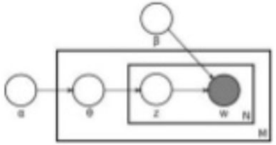
- Answer ranking
- Feed ranking
- Topic recommendations
- User recommendations
- Email digest
- Ask2Answer
- Duplicate Questions
- Related Questions
- Spam/moderation
- Trending now
- ...



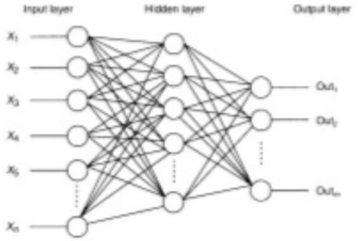
http://www.slideshare.net/xamat/10-more-lessons-learned-from-building-machine-learning-systems/6-ML_Applications_Quora_Answer_ranking

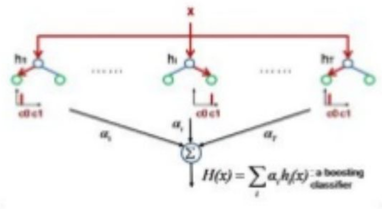
Models

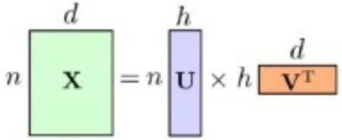
- Logistic Regression
- Elastic Nets
- Gradient Boosted Decision Trees
- Random Forests
- (Deep) Neural Networks
- LambdaMART
- Matrix Factorization
- LDA
- ...

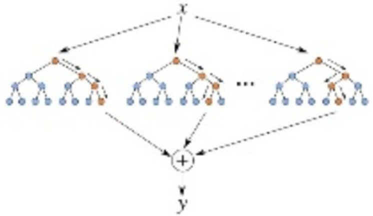


$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$





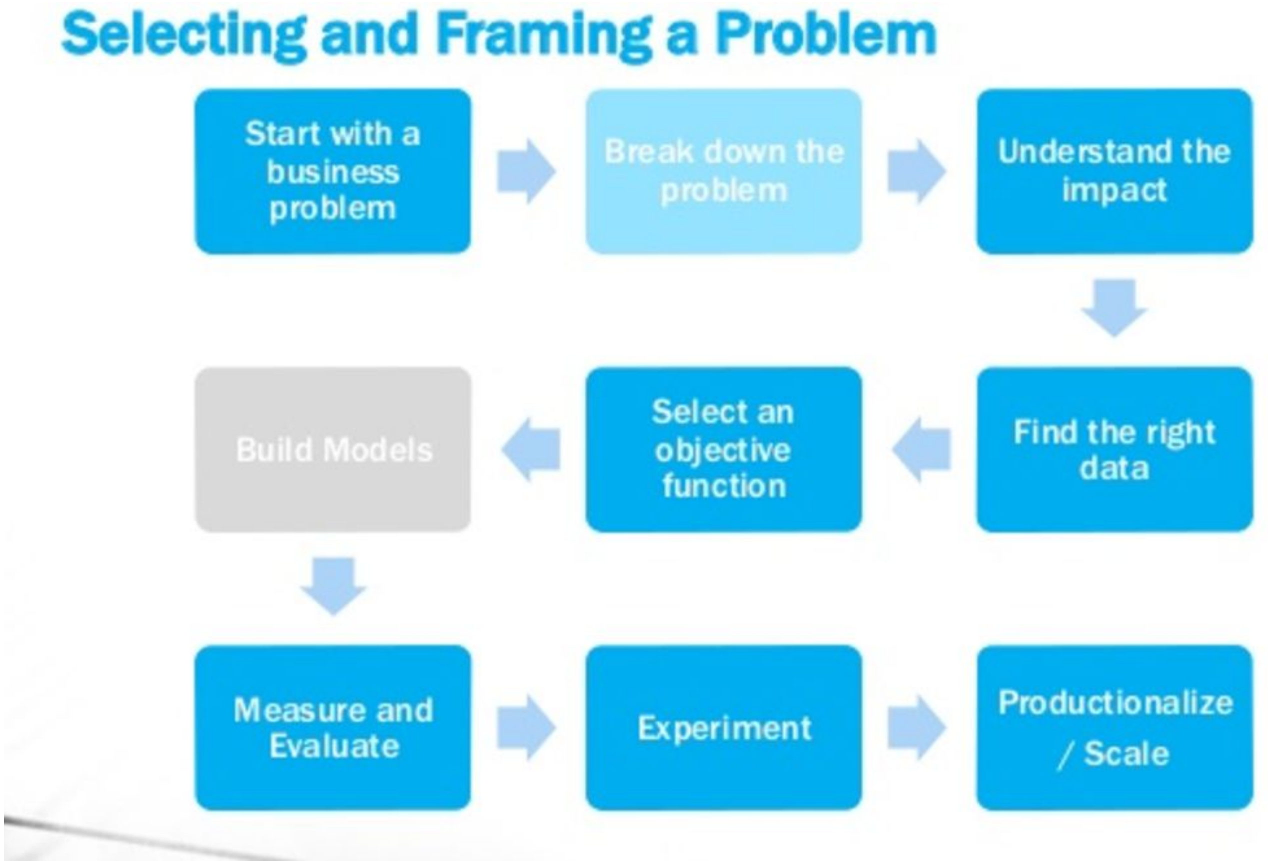




$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1).$$

http://www.slideshare.net/xamat/10-more-lessons-learned-from-building-machine-learning-systems/6-ML_Applications_Quora_Answer_ranking

Not just building models



http://www.slideshare.net/SessionsEvents/3-ewa-dominowska-managing-machine-learning-projects-in-industry

Admin stuff

Our TA

Yun Jin will be grading your Homework

Submit Homework to Canvas / Github

Content

- Recommendation Algorithm: Collaborative filtering, Low rank matrix decomposition
- Boosting and Combining Models
- Neural Networks
- Support Vector Machines and Image Classification
- Expectation Maximization (EM) for Gaussian Mixtures
- Hidden Markov Model (HMM): model for sequential data, tagging text
- Deep Learning

Textbooks

- **Pattern Recognition and Machine Learning. Bishop** (required)
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Deep Learning. Ian Goodfellow, Yoshua Bengio and Aaron Courville
- Mining of Massive Datasets. Jure Leskovec, Anand Rajaraman and Jeffrey D. Ullman (Chapters 8, 9).
- Machine Learning: A Probabilistic Perspective. Kevin P. Murphy

Work

Homework 30% (~ 5 hw)

Quizzes 30% (~5 quizzes)

Final Project 30%

Labs / class participation /   10%

Final Project:

Clickthrough rate prediction on Vungle data

- Large dataset
- Spark
- Vungle will provide a databrick (Spark) account
- Feature eng
- Team of 3
- Friendly competition

Deep Learning

- On Vungle data (max two teams)
- Kaggle: Data Science Bowl 2017 (one or two teams)
    - lung cancer detection
    - CT images
    - You are responsible for AWS charges

Technology for the class

I am going to assume knowledge of

- Spark
- AWS (getting a cluster, GPU, single node)
- You may need a better computer for hw 1

Schedule (approx)

Quizzes: 10:30am Tuesdays

HW due dates: Feb 2rd, 9th, 18th, 23th, Mar 1nd

Project due dates:

Team: Jan 29th

Proposal: Feb 4th

Project update: Feb 18th

Presentation: Match 11th, Slides due: March 10th at 5pm

Final write-up: March 7th.

**Quizzes are closed books and no notes.

Lecture Style

- Lectures that require a lot of math will be in the blackboard
- Lecture comes with required readings
- I provide lecture notes

Attitude

- This class is hard
- Do the required reading
- Start homework early (no extensions)
- Come to my office hours
- Learning happens in class and outside class
- Remember: **You are here to learn and get a job**

Class policy

- NO slack, No phone, NO facebook

Any questions?

Papers on CTR

https://www.eecs.tufts.edu/~dsculley/papers/ad-click-prediction.pdf

http://people.csail.mit.edu/romer/papers/TISTRespPredAds.pdf

Homework 1

Recommendation Systems:
Agenda

- Content based Recommendations
- Case Study: YouTube
- Collaborative Filtering (Thursday)

Recommendation Systems

Required Reading:

- Chapter 9 from ``Mining of Massive Datasets". Jure Leskovec, Anand Rajaraman and Jeffrey D. Ullman
- https://engineering.quora.com/Machine-Learning-at-Quora

Optional:

https://www.youtube.com/watch?v=bLhq63ygoU8

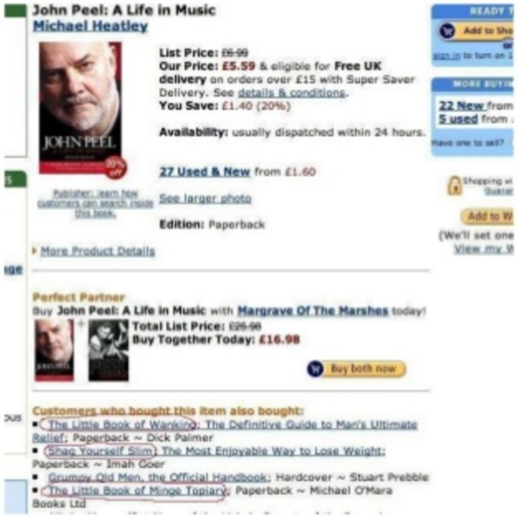
Recommendation System:

Definition: Web application that involve predicting user responses to options.

- Google News offers news articles to on-line newspaper readers, (based on a prediction of reader interests)
- Amazon offers suggestions about what user might like to buy (based on their past history of purchases and/or product searches)
- Recommending YouTube videos.
- Netflix offers users recommendations of movies
- Pandora recommends songs.
- Quora recommends stories to users.

Value of Recommendation

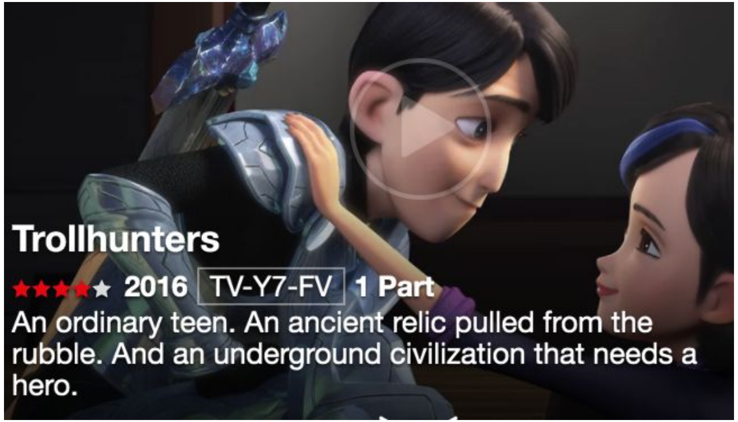
- Netflix: 2/3 of the movies watched are recommended
- Google News: recommendations generate 38% more clickthrough
- Amazon: 35% sales from recommendations
- Choicestream: 28% of the people would buy more music if they found what they liked.



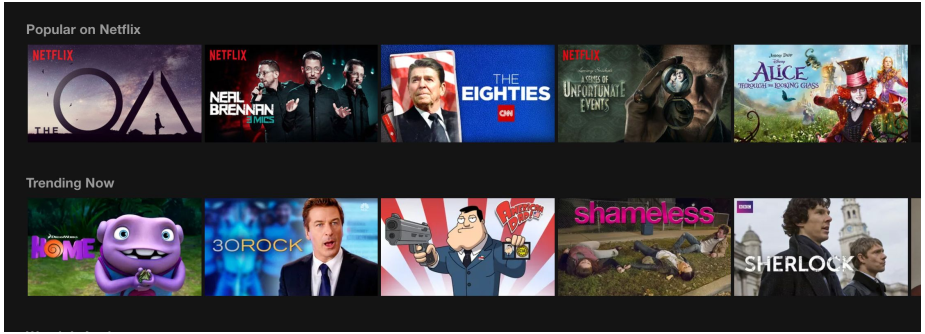
http://www.slideshare.net/xamat/recommender-systems-machine-learning-summer-school-2014-cmu

Recommendation System Problems

Rating Prediction: predict on a 5 star system (it could also be a probability of click)



Ranking: Predict what to display in an actual real system



Most books/ publications talk

About Rating Prediction

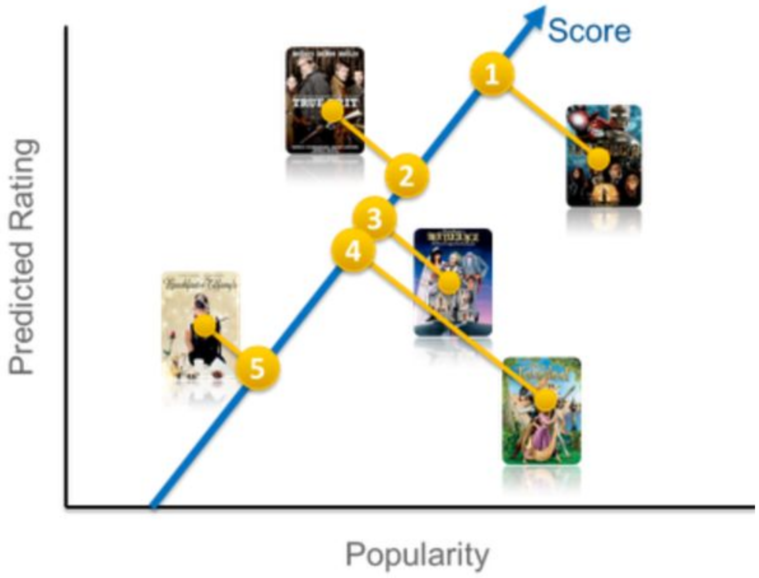
Ranking

If you have a “rating prediction” and a “popularity prediction” models you can model ranking.

`ranking(u,v) = w1 p(v) + w2 r(u,v) + b`,

`u=user`, `v=item`, `p=popularity` and `r=predicted rating.`

You learn w1, w2, b from data

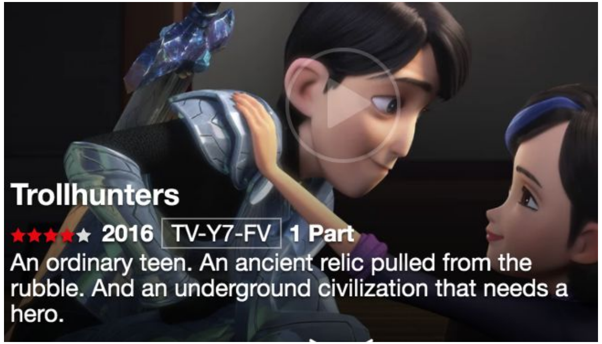


http://technocalifornia.blogspot.com/2013/07/recommendations-as-personalized.html

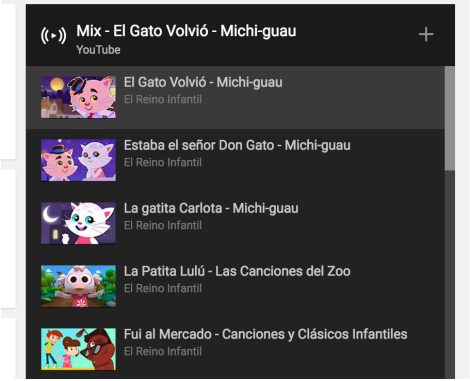
Recommendation System Problems

Rating Prediction:

- predict on a 5 star system



- predict a probability of click



Classification of Recom. Systems

- **Content-based systems:** consists in matching up the attributes of a user profile with the attributes of a content object (item), in order to recommend to the user new interesting items.
  - If a Netflix user has watched many cowboy movies, then recommend a movie of ``cowboy" genre.
- **Collaborative filtering:** systems recommend items based on similarity measures between users and/or items. The items recommended to a user are those preferred by similar users.

Users and Items, Utility Matrix

Items: movies, videos, stories, songs, books

Utility matrix: degree of preference that a user has for an item. Example: 1-5 scale ratings of movies

2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

http://www.slideshare.net/xamat/recommender-systems-machine-learning-summer-school-2014-cmu

Implicit versus Explicit Rating

- **Ask users to rate items** (explicit rating). Movie ratings are generally obtained this way, and some online stores try to obtain ratings from their purchasers.
  - users are unwilling to provide responses
  - biased by the very fact that it comes from people willing to provide ratings.
- **Make inferences from users' behavior** (implicit rating).
  - a user that buys a product at Amazon, watches a movie on YouTube, or reads a news article
  - ``like" this item.

Content-Based Recommendations

- Based on content as opposed to user behaviour
- Common for recommending **text based products** (web pages, news)
- Items and users are described by a set of **features**

Item profile: example

Example of movie profile:

- The set of actors of the movie.
- The director.
- The year in which the movie was made.
  - Some viewers prefer old movies,others watch only the latest releases.
- The genre or general type of movie.
  - Some viewers like only comedies, others dramas or romances.

Can you think about other features? Can we make features from reviews? How we get a User profile?

Item profile (2)

- Source
  - Author, publisher
- Location
  - Movies (only interesting for a region), pictures can be tagged to location
  - Represented with latitude, longitude or country/state/city
- Image features
- Audio features
- Application specific

User Profile: example

Example 2: **Netflix recommendations**

Item profile:

x_1 = has Julia Roberts,

x_2 = directed by Lars von Trier,

x_3 = is horror,

x_4 = is a comedy

User profile:

x_1 = probability that the user likes movies with Julia Roberts,

x_2 = probability that the user likes movies directed by Lars von Trier,

x_3 = probability that the user that likes horror movies,

x_4 = probability that the user likes comedies

- **Content based profile:** summary of the profile of the items these user liked / purchased

User Profile (2)

- Demographics
- Declared Interests
- User current location (from IP address)
- Usage based features
  - Last time visit, frequency of visits (weekly, monthly), frequency per device
- Search history
  - Bag of word model
- Item set
  - Set of items the user showed interest (e.g. clicked, shared, liked)

Features from document

- Remove stopwords
- Compute TF-IDF scores
- Keep works with high score (top N= 1000, 5000, 1M)

Computing TF-IDF

$$\mathrm{TF}(t) = \frac{\text{Number of times term } t \text{ appears in a document}}{\text{Total number of terms in the document}}.$$

$$\mathrm{IDF}(t) = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}}\right).$$

$$\text{TF-IDF}(t) = \mathrm{TF}(t) \cdot \mathrm{IDF}(t)$$

Features from document (2)

- Topic Modeling
  - Unsupervised or supervised methods for computing topic models
  - Next class will talk about Matrix Factorization for modeling topics
  - Each document has a probability in a topic

`X_d = (p_1, p_2, … ,p_K)`

Where `p_i` is the probability of the document being in topic `i`

Similarities

- Measure of how similar are a pair of items, users, (user, item).
- Which similarity to use depends on the application and they type of features

Jaccard Similarity

$$\frac{|A \cap B|}{|A \cup B|}$$

- Given two set A and B
- Measure of similarity between two sets
- Between 0 and 1
- When should be used?
    - Binary features (how to take binary features to sets?)
    - Lose info if used in non-binary features
    - Models well “lack of ratings”

Cosine Similarity

- Used often in similarities between documents
- Lack of rating is treated a “0” (more similar to disliking than liking)
- Measures the cosine of the angle between vectors
- Often used in “positive” space

$$cosine(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Pearson Correlation Coefficient

- Measure of linear correlation between two variables
- value between -1 and -1
  - 1 is total positive correlation, 0 is no correlation, -1 is total negative correlation

$$r(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$w_{ij} = \frac{\sum_{k \in I_i \cap I_j}(R_{ik} - \bar{R}_i)(R_{jk} - \bar{R}_j)}{\sqrt{\sum_{k \in I_i \cap I_j}(R_{ik} - \bar{R}_i)^2 \sum_{k \in I_i \cap I_j}(R_{jk} - \bar{R}_j)^2}}$$

We use a different version for the homework

Recommending Items to Users based on content -- Unsupersived (1)

- Compute profile vectors for users and items
- Find a similarity measure and compute similarity between users and items
- Recommend to a user items with *high similarity*
- Scale
  - When you have too many users and items it is not feasible to compute similarity between all of them
  - Locality-sensitive-hashing techniques can be used to place item profiles in buckets
  - Given a user easy to find buckets with high similarity to the user

Recommending Items to Users based on content -- Supervised (2)

- Compute profile vectors for users and items
- Train a model using the feature vector to predict observed ratings
  - Actual ratings, clicks, likes or a combination
  - Regression to predict numerical ratings
  - Classification to predict prob of click
  - Multi-class classification to predict ordinal ratings (e.g. 1-5 stars)

Advantages of Content Based Recom

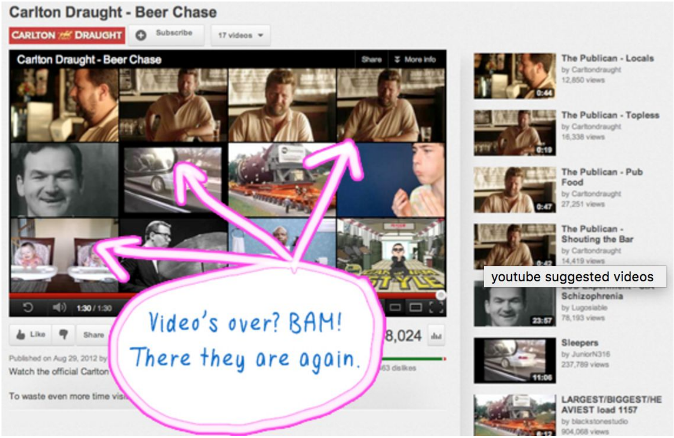
- You don't need data on other users
- You don't have cold-start problem for a new item. Able to recommend new and unpopular items
- You can provide explanations of recommended items by listing content features that caused an item to be recommended

Challenges with Content based Recom

- Constructing the feature vector could be a difficult task (need domain knowledge).
- New genres ``dogme 95".
- Some kind of items are not amenable to easy feature extraction methods (movies, music)
- Hard to exploit quality of judgements of other users.

Case study:

YouTube Recommendation system



http://www.mcdougallinteractive.com/blog/video-marketing/surprise-youtube-has-a-new-ranking-system/

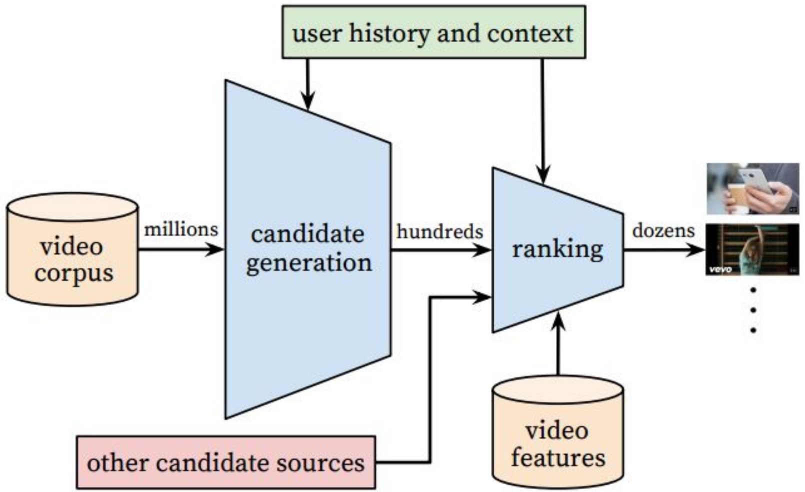
YouTube Recommendations:

- Personalized (per user) recommendation system
- Highly specialized distributed learning algorithms
- Deep learning models
- Models learn approximately one billion parameters and are trained on hundreds of billions of examples

http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf

YouTube Recommendations:
system architecture

- A deep learning network generates a set of candidate videos
  - Returns hundreds of videos that may be relevant to the user
  - Nonlinear generalization of MF
  - Collaborative filtering.
- Another deep learning network ranks the videos.



http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf

YouTube Recommendations:
Recommendation as Classification

$$P(w_t = i|U,C) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$

- Multiclass logistic regression classifier (sofmax)
- Ranking video `i` at time `t` given user `U` and context `V`
- What do you think goes into context?
- u and v_i are “embeddings”

http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf