Recommendation System

Dr. Quang Hieu Vu
Senior Data Scientist at Zalora

Some slides are copied from the presentation of Kien Nguyen
in the 2nd meetup of the Big Data and Data Science community.

Outline

1. Values of recommendations
2. Collaborative filtering method:
    a. User-based
    b. Item-based
    c. Latent factor model (Matrix Factorization)
3. Content-based method
4. Hybrid method
5. Filtering technique
6. Scalability

Information Overload

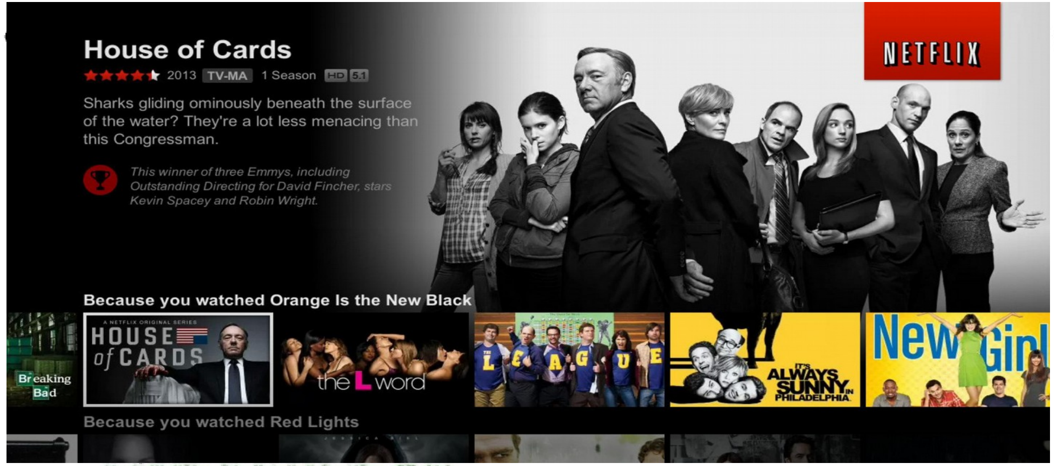
“People read around 10 MB worth of material a day, hear 400 MB a day, and see 1 MB of information every second” - The Economist, November 2006

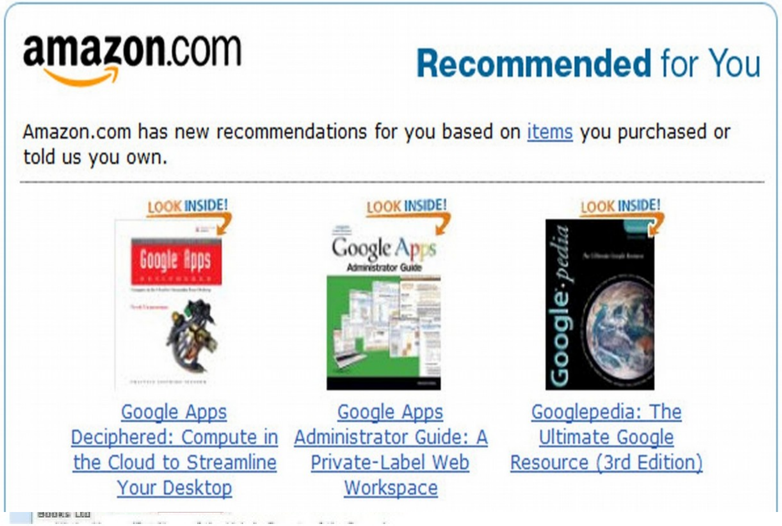
An End for the Age of Search

- **... long live the Age of Recommendation!**
- **Chris Anderson in “The Long Tail”**
  - *“We are leaving the age of information and entering the age of recommendation”*
- **CNN Money, “The race to create a 'smart' Google”:**
  - *“The Web, they say, is leaving the era of search and entering one of discovery. What's the difference? Search is what you do when you're looking for something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you.”*

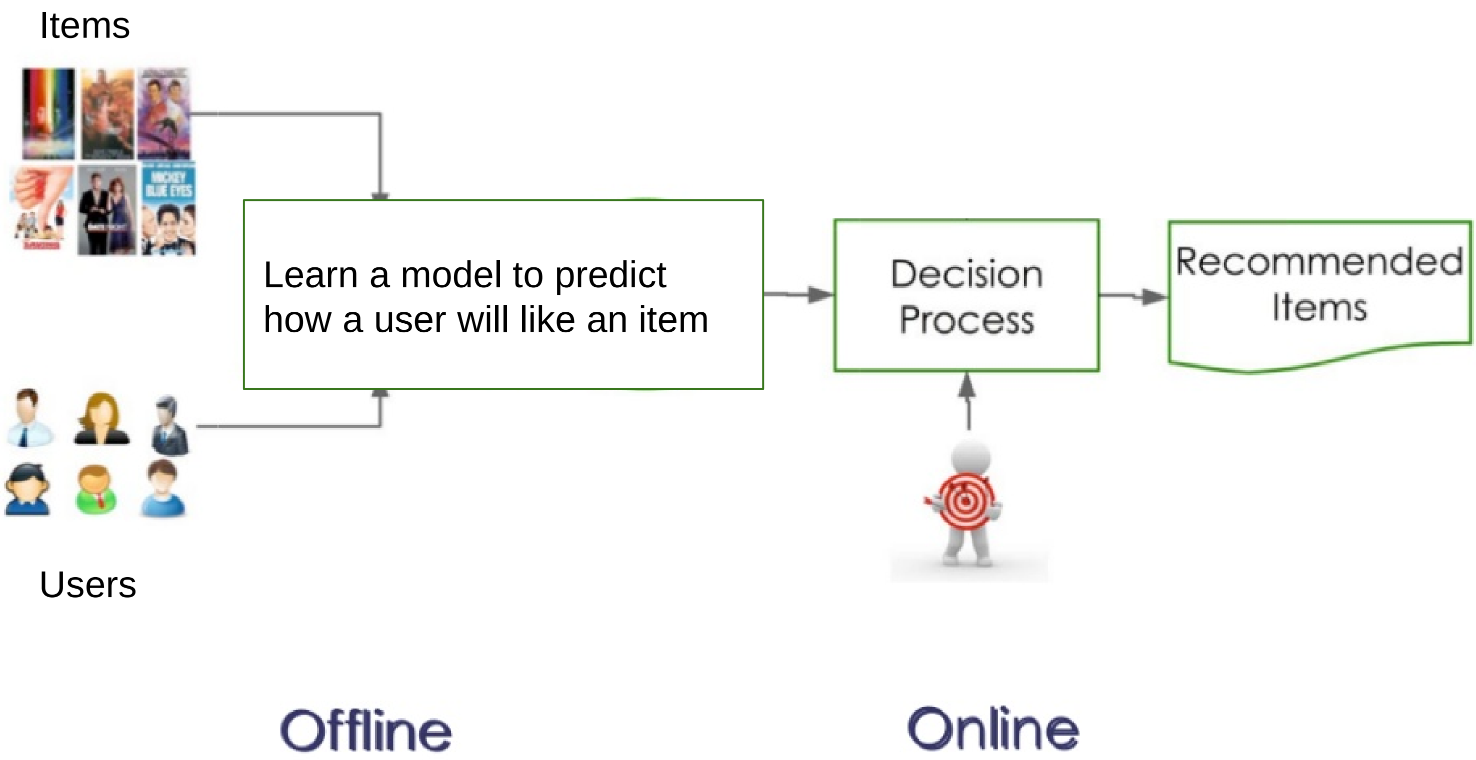
The Values of Recommendations

- Netflix: 2/3 of the movies watched are recommended
- Google News: recommendations generate 38% more clickthrough
- Amazon: 35% sales from recommendations

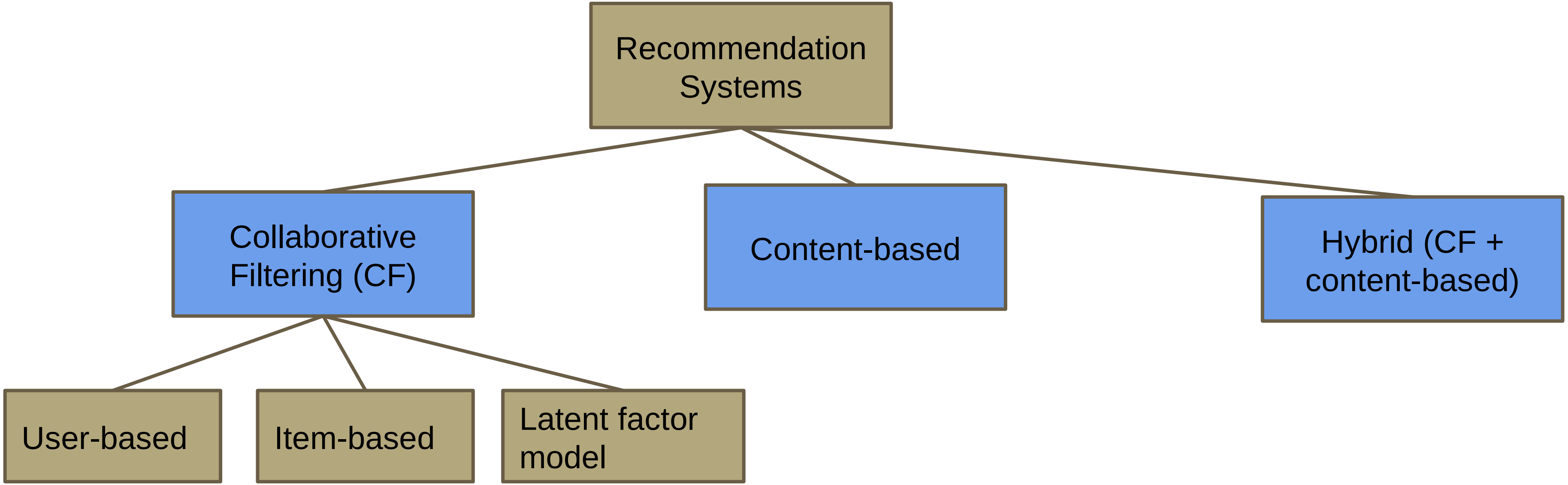




Recommendation Systems Workflow



Approaches


Recommendation Systems
Collaborative Filtering (CF)
Content-based
Hybrid (CF + content-based)
User-based
Item-based
Latent factor model

Collaborative Filtering (CF) Method

- Filter information by using the recommendations of other people
- Assumption: people agreed in their evaluation of certain items in the past are likely to agree again in the future

Explicit Rating

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

Implicit Rating

View

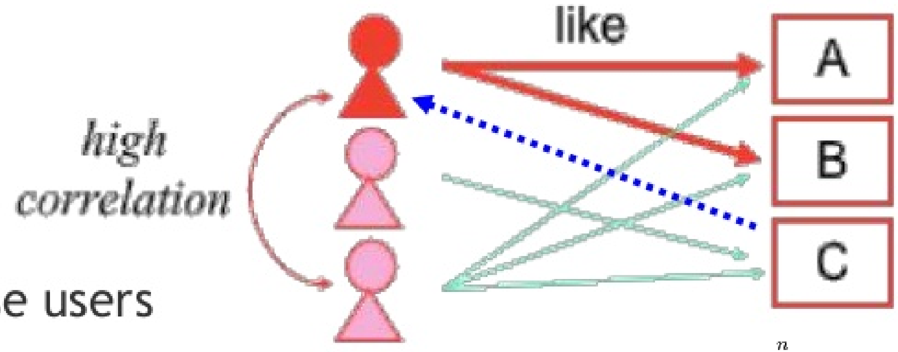
Add to cart

Purchase

→ Combination to create ratings

User-based CF

- Use user-item rating matrix
- Make user-to-user correlations
- Find highly correlated users
- Recommend items preferred by those users



$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum\limits_{i=1}^{n} A_i \times B_i}{\sqrt{\sum\limits_{i=1}^{n} (A_i)^2} \times \sqrt{\sum\limits_{i=1}^{n} (B_i)^2}}$$

Pearson Correlation :

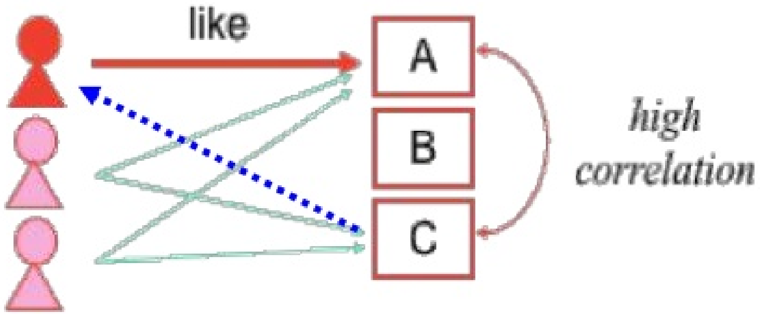
$$userSim(u,n) = \frac{\sum_{i \subset CR_{u,n}} (r_{ui} - \bar{r}_u)(r_{ni} - \bar{r}_n)}{\sqrt{\sum_{i \subset CR_{u,n}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \subset CR_{u,n}} (r_{ni} - \bar{r}_n)^2}}$$

Prediction Function :

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n \subset neighbors(u)} userSim(u,n) \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \subset neighbors(u)} userSim(u,n)}$$

Item-based CF

- Use user-item ratings matrix
- Make item-to-item correlations
- Find items that are highly correlated
- Recommend items with highest correlation



Similarity Metric :

$$itemSim(i,j) = \frac{\sum_{u \subset RB_{i,j}} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \subset RB_{i,j}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \subset RB_{i,j}} (r_{uj} - \bar{r}_u)^2}}$$

Prediction Function :

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} itemSim(i,j) \cdot r_{uj}}{\sum_{j \in ratedItems(u)} itemSim(i,j)}$$

User-based vs Item-based

	User-based	**Item-based**
Scalability	Bad when user size is large	Bad when item size is large
Cold start	Bad for new users	Bad for new items

A user profile usually contains fewer ratings than an item profile

→ Better to compute item similarity than user similarity

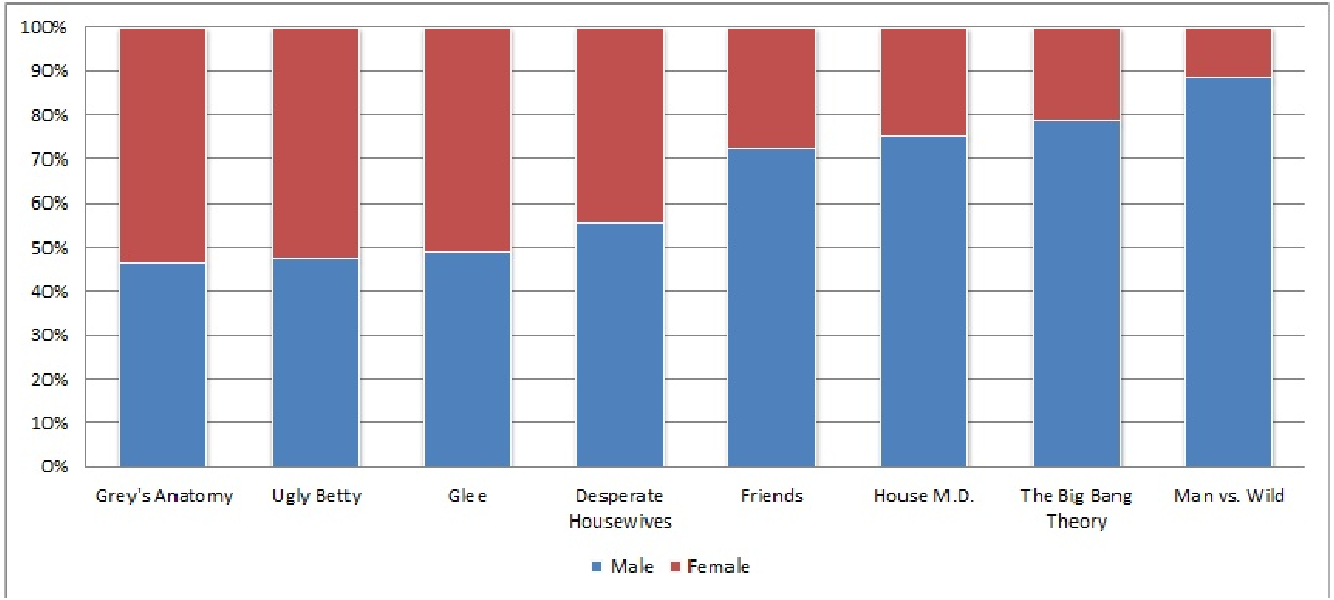
Example use case: Item-based method is used by Amazon

User Cold Start

- How to recommend items to new users?
  - Non-personalization recommendation
    - Most popular items
    - Highly Rated items
  - Using user register profile (Age, Gender, …)

User Cold Start

- Example: Gender and TV shows



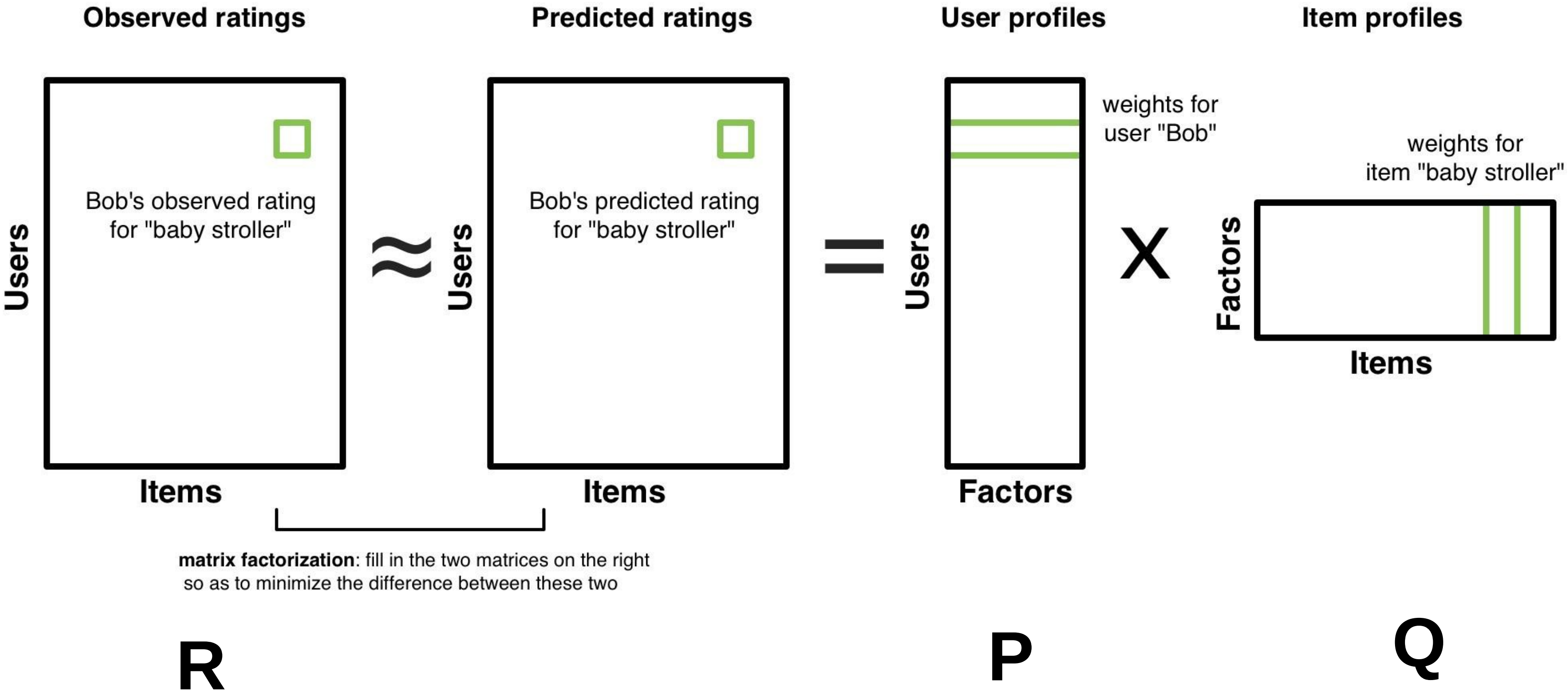
Data comes from IMDB : http://www.imdb.com/title/tt0412142/ratings

Item Cold Start

- Not recommend
- Use item content (content-based method)

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

Latent Factor Model (Matrix Factorization)



Example of Factorization

items

users

1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

~

~

users

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

•

items

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

Here we assume that 3 factors are sufficient

Learn P, Q

$$\text{Min}_{p*,q*} \sum_{\text{known } r_{ui}} \left(r_{ui} - p_u^T q_i\right)^2 + \underbrace{\lambda\left(\|p_u\|^2 + \|q_i\|^2\right)}_{\text{regularization}}$$

p_u - user-factors of u

q_i - item-factors of i

r_{ui} - rating by u for i

Predict the rating of a new user

$$\hat{r}_{ui} = \sum_k p_{uk} q_{ik}$$

Gradient Descent

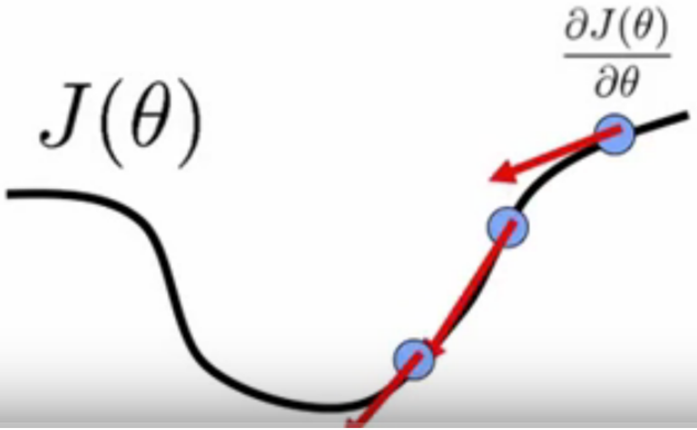
- Initialization
- Step size
  - Can change as a function of iteration
- Gradient direction
- Stopping condition

Initialize θ

Do {

$\theta \leftarrow \theta - \alpha \nabla_{\theta} J(\theta)$

} while ($\alpha \|\nabla J\| > \epsilon$)



Gradient Descent

Perform till convergence:

- For each training example r_{ui} :
  - Compute prediction error: $e_{ui} = r_{ui} - p_u^T q_i$
  - Update item factor: $q_i \leftarrow q_i + \gamma(p_u e_{ui} - \lambda q_i)$
  - Update user factor: $p_u \leftarrow p_u + \gamma(q_i e_{ui} - \lambda p_u)$

- The parameters are modified by a magnitude proportional to γ in the opposite direction of the gradient (of the function that we want to minimize)
- Two constants to tune: γ (step size) and λ (regularization)

Adding Biases

- **Much of the observed variation in rating values is due to effects associated to either users or items (individually)**
- Example: certain users give higher ratings and certain items are widely perceived as better
- First order approximation of the bias involved in rating r_{ui} is:
  - $b_{ui} = \mu + b_u + b_i$
  - Where μ is the overall average rating
- **Example:** If μ=3.7, if Titanic is a movie that tends to be rated 0.5 better than an average movie, and Marius is a critical user who tends to rate 0.3 stars lower than the average
  - $b_{marius,\ titanic} = 3.7 - 0.3 + 0.5 = 3.9$

Adding Biases

- The rating prediction function is now
  - $\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i$
- And the corresponding new error function that we must minimize is:

$$\min_{p*,q*,b*} \sum (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2)$$

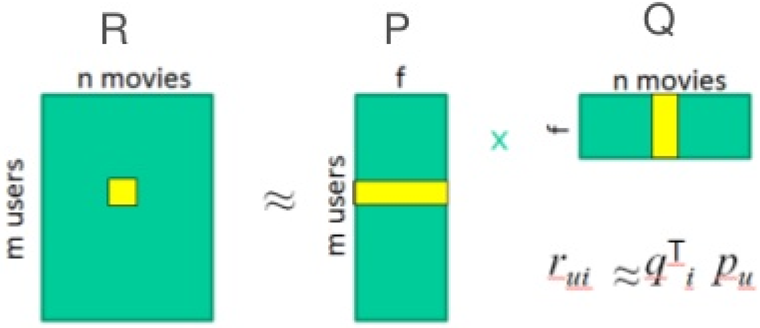
Latent Factor Model

Pros:

- Good performance (e.g., in Netflix Prize 2006 - 2009)
- Fast prediction: only need one vector multiplication

Cons:

- Hard to interpret the models
- Need to keep updating the models



Collaborative Filtering Method

- **Pros:** require minimal knowledge engineering efforts (knowledge poor)
  - Users and products are symbols without any internal structure or characteristics
- **Cons:**
  - Requires a large number of **explicit** and **reliable** “rates” to bootstrap
  - Requires products to be standardized (users should have bought **exactly** the same product)
  - Assumes that **prior behavior determines current behavior** without taking into account “contextual” knowledge (session-level)
  - Does not provide information about products or explanations for the recommendations
  - Does not support sequential decision making or recommendation of “good bundling”, e.g., a travel package.

Content-based Method

- What is the **content** of an item?
- It can be explicit **attributes** or **characteristics** of the item. For example for a film:
  - Genre: Action / adventure
  - Feature: Bruce Willis
  - Year: 1995
- It can also be **textual content** (title, description, table of content, etc.)
  - Several techniques to compute the distance between two textual documents
  - Can use NLP techniques to extract content features
- Can be extracted from the signal itself (audio, image)

Content-based Method

- **Item representation**

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

- **User profile**

Title	Genre	Author	Type	Price	Keywords
…	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

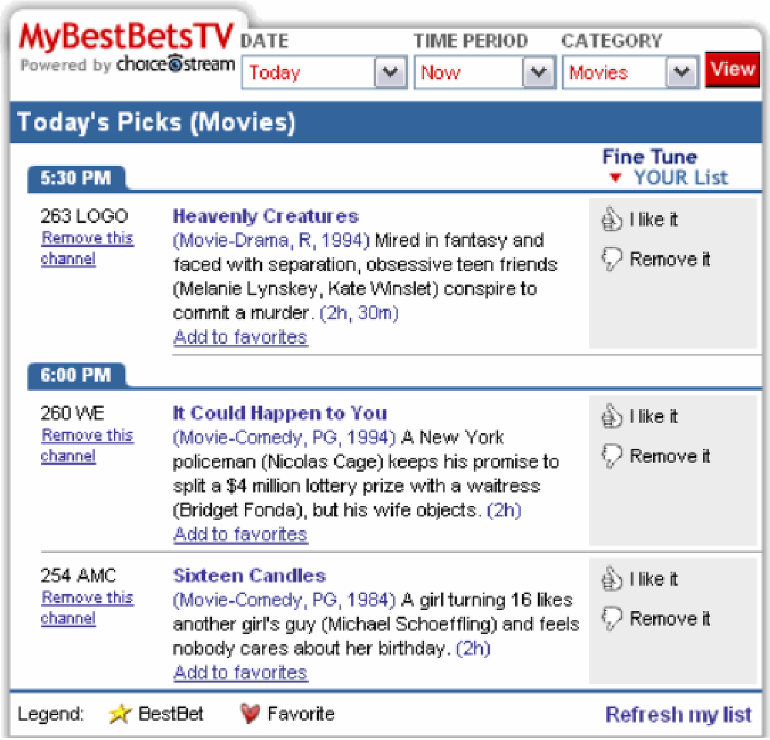
$keywords(b_j)$ describes Book b_j with a set of keywords

- Approach
  - Compute the similarity of an unseen item with the user profile based on the keyword overlap

$$\frac{2 \times |\boldsymbol{keywords}(\boldsymbol{b_i}) \cap \boldsymbol{keywords}(\boldsymbol{b_j})|}{|\boldsymbol{keywords}(\boldsymbol{b_i})| + |\boldsymbol{keywords}(\boldsymbol{b_j})|}$$

Content-based Using Machine Learning

For example, based on a set of webpages rated as “relevant” or “irrelevant” by users, build a classifier to classify unrated webpages



Content-based Method

Pros:

- No cold start problem
- Able to recommend users with unique taste
- Able to recommend unpopular items
- Can provide explanations of recommended items (use their features)

Cons:

- Require contents to be encoded as meaningful features
- Require technology from information retrieval, natural language processing, image processing
- Unable to exploit quality judgement from other users

Hybrid Method

$$\hat{r}_{ui} = p_u^\top \left(q_i + \boxed{\mid N(\theta, i) \mid^{-\frac{1}{2}} \sum_{j \in N(\theta, i)} \theta_{ij} y_j}\right)$$

$N(\theta, i)$	Neighbors of item i, estimated by content-based similarity
y_j	Item vectors of these neighbors
θ_{ij}	Content-based similarity between i and j

Lu, Z.; Dou, Z.; Lian, J.; Xie, X.; and Yang, Q. 2015. Content-based collaborative filtering for news topic recommendation. In AAAI.

Filtering technique

- Multiple processing layers vs. combined hybrid method
  - Fallback concept
- Customer segmentation
  - Age, gender, location, season…
- Where to put filters?
  - Input before recommendation
  - During recommendation
  - Output after recommendation

Scalability

Example use case: Matrix factorization method

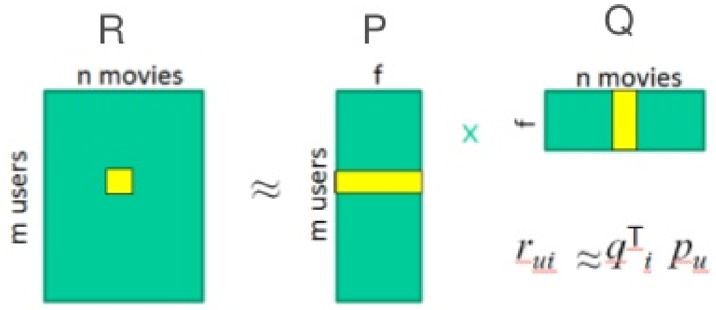
1. **Offline process - Training:**

- Parallel algorithm to learn P, Q matrix by "alternating least squares" algorithm(*)

  Fix Q and minimize over P and vise versa

$$\min_{\mathbf{p}_u} \sum_{i \in \{R_u\}} (R_{ui} - \mathbf{p}_u^t \mathbf{q}_i)^2 + \lambda \|\mathbf{p}_u\|^2$$

  (optimize for all $\mathbf{p}_u$ in parallel)



- Use map-reduce/spark for parallelization

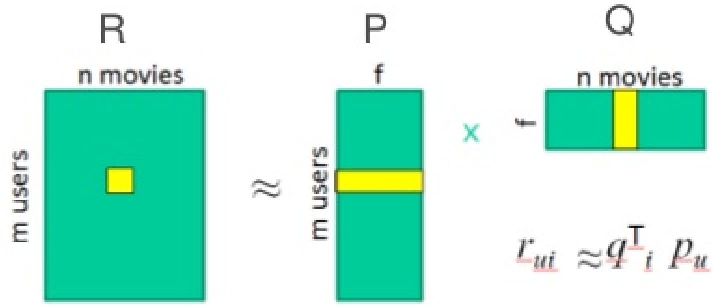
- Incremental learning: Update U, V by new ratings only

(*) Parallel Matrix Factorization for Recommender Systems Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit S. Dhillon Department of Computer Science, The University of Texas at Austin, Austin, TX 78712, USA

Scalability

Example use case: Matrix factorization method

2. Online process



- Store matrices P, Q by indexing on user id, item id
- Multiply vectors in parallel by map-reduce/spark (multiplying $\mathbf{p}_u$ by all $\mathbf{q}_i$)
- Use cache to store past results

Conclusions

- Collaborative filtering (user-based, item-based, latent factor model): suitable for systems with many ratings (or user activities) available
    - Item-based is generally better than user-based (item profile is richer than user profile)
    - Latent factor model is becoming more popular in literature
- Content-based method: use when ratings are limited, new users/items
- Hybrid method: best of both worlds, state-of-the-art
- Filtering: customer segmentation, multiple processing layers
- Scalability: precomputation, parallel processing in training and test