Collaborative Filtering (Cont'd)

Harnessing quality judgments of other users

Thanks for source slides and material to: J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets

Recall: Three Approaches to Recommendation Systems

1) Content-based

- Use characteristics of an item
- Recommend items that have similar content to items user liked in the past
- Or items that match pre-defined attributes of the user

2) Collaborative filtering

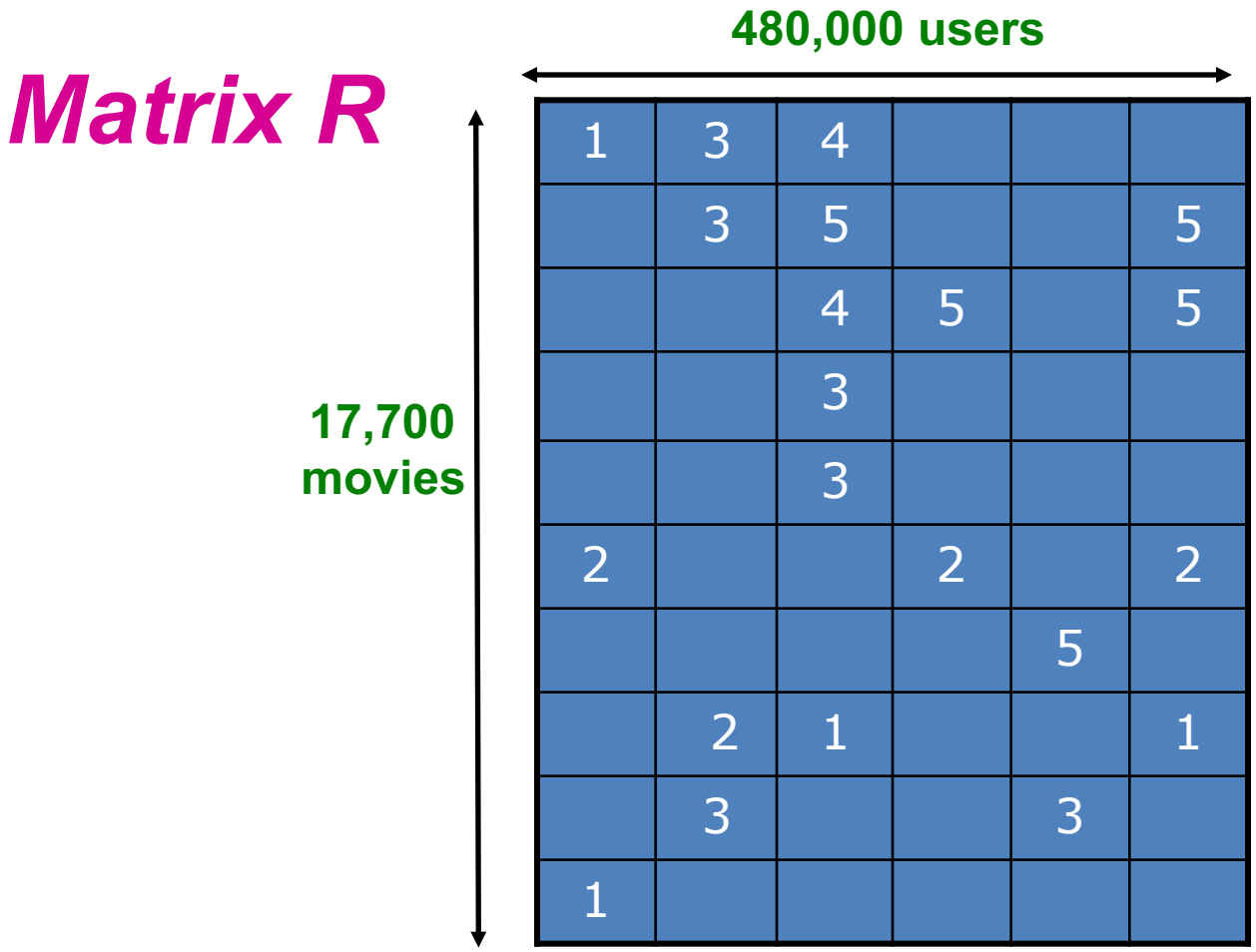
- Build a model from a user's past behavior (items previously purchased or rated) and similar decisions made by other users
- Use the model to predict items that the user may like
- Collaborative: suggestions made to a user utilize information across the entire user base

3) Hybrid approaches

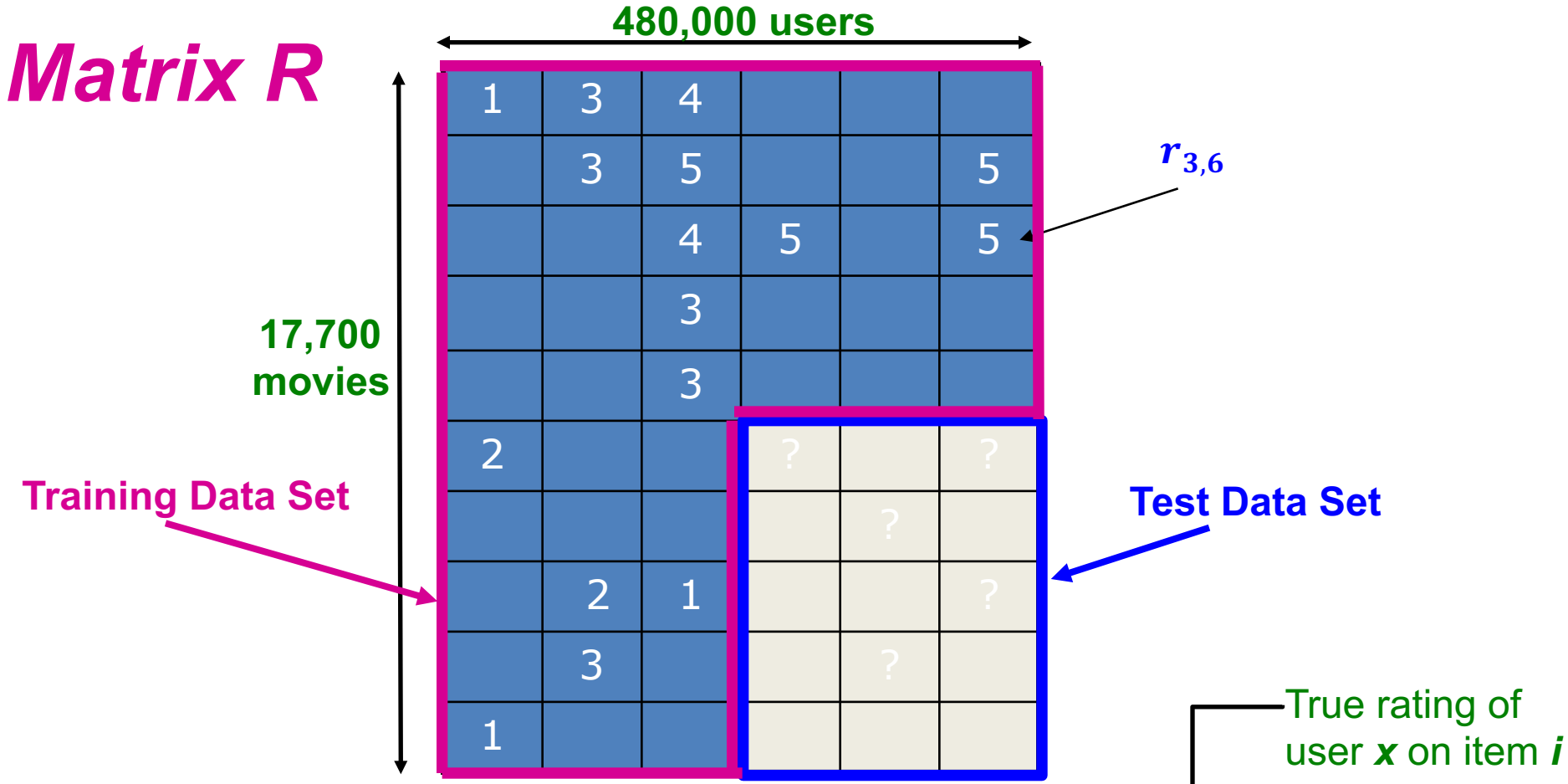
The Netflix Prize

- **Training data**
  - 100 million ratings, 480,000 users, 17,770 movies
  - 6 years of data: 2000-2005
- **Test data**
  - Last few ratings of each user (2.8 million)
  - **Evaluation criterion:** Root Mean Square Error (RMSE) $=$ $\frac{1}{|R|}\sqrt{\sum_{(i,x)\in R}(\hat{r}_{xi}-r_{xi})^2}$
  - **Netflix's system RMSE: 0.9514**
- **Competition**
  - 2,700+ teams
  - **$1 million** prize for 10% improvement on Netflix

The Netflix Utility Matrix *R*



Utility Matrix *R*: Evaluation



$$\text{RMSE} = \frac{1}{|R|}\sqrt{\sum_{(i,x)\in R}(\hat{r}_{xi} - r_{xi})^2}$$

True rating of user ***x*** on item ***i***

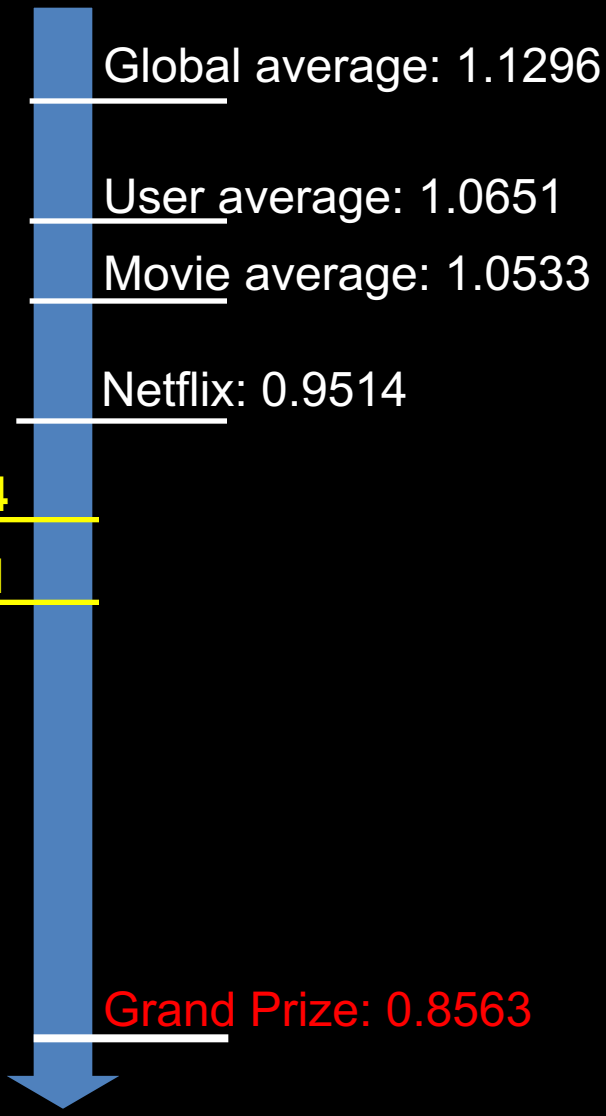
Predicted rating

Evaluation

- **Goal:** Make good recommendations
  - Quantify goodness using **RMSE:**
    Lower RMSE ⇒ better recommendations
  - Want to make good recommendations on items that user has not yet seen. Can't really do this!

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

Performance of Various Methods
Netflix prize



Recall: Collaborative Filtering Overview

CF works by **collecting user feedback**: **ratings for items**

- Exploit similarities in rating behavior among users in determining recommendations

Two classes of CF algorithms:

1. **Neighborhood-based or Memory-based approaches**
   - User-based CF
   - Item-based CF
2. **Model-based approaches**
   - Estimate parameters for statistical models for user ratings
   - Latent factor and matrix factorization models.

MODEL-BASED CF

Model-based Collaborative Filtering

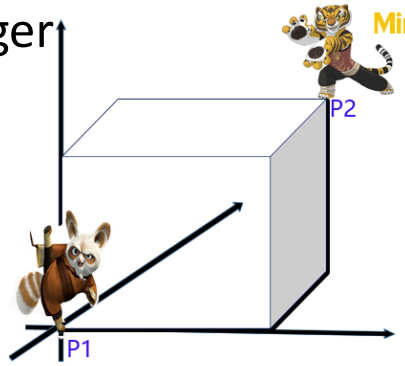
- Provide recommendations by estimating parameters of statistical models for user ratings
- Design and development of models can allow system to learn to recognize complex patterns
  - Based on training set – supervised learning
- Then make intelligent predictions for CF tasks based on the **learned models**
- Example models:
  - Bayesian models
  - **Clustering models**
  - Dependency networks
  - Classification algorithms (if users ratings are in categories)
  - Regression models and SVD (singular value decomposition) methods for numerical ratings.

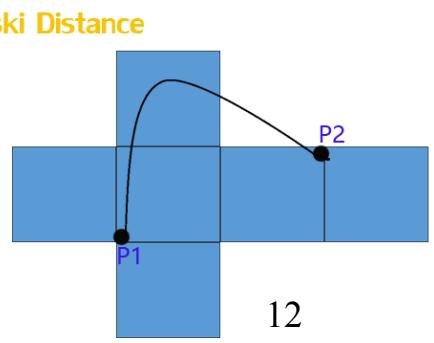
CLUSTERING CF

Clustering CF Algorithms

- **Cluster = collection of data objects that are:**
  - Similar to one another within the same cluster
  - Dissimilar to objects in other clusters
- **Measurement of similarity between objects uses:**
  - Pearson correlation
  - Cosine similarity (it's important to study your data! E.g., http://grouplens.org/blog/similarity-functions-for-user-user-collaborative-filtering/)
  - Minkowski distance
    - Two objects $X = (x_1, x_2, \ldots, x_n)$, $Y = (y_1, y_2, \ldots, y_n)$
    - Where q is a positive integer
    - If q=2: Euclidean distance.

$$d(X,Y) = \sqrt[q]{\sum_{i=1}^{n} |x_i - y_i|^q},$$





Clustering Algorithms

- Common clustering method
  - K-Means
  - Hierarchical Clustering
  - Mean-Shift
  - More on Clustering in Chapter 7.

Clustering Algorithms (Cont'd)

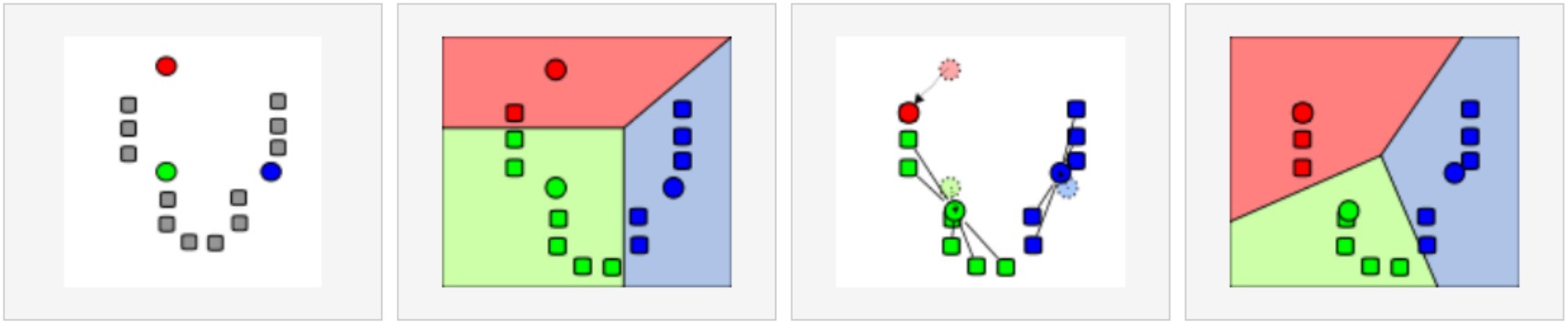
- **Key operation:** **Repeatedly combine two nearest clusters**
- **(1) How to represent a cluster of many points?**
  - **Key problem:** As you merge clusters, how do you represent the "location" of each cluster, to tell which pair of clusters is closest?
- **Euclidean case:** each cluster has a *centroid* = average of its (data)points
- **(2) How to determine "nearness" of clusters?**
  - Measure cluster distances by distances of centroids.

K–means Algorithm(s)

- Assumes Euclidean space/distance
- Start by picking ***k***, the number of clusters
- Initialize clusters by picking one point per cluster
  - **Example:** Pick one point at random, then ***k*-1** other points, each as far away as possible from the previous points.

Example: Standard K-Means in this case K-3

Demonstration of the standard algorithm



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

Source: https://en.wikipedia.org/wiki/K-means_clustering

Populating Clusters

1) For each point, place it in the cluster whose current centroid it is nearest

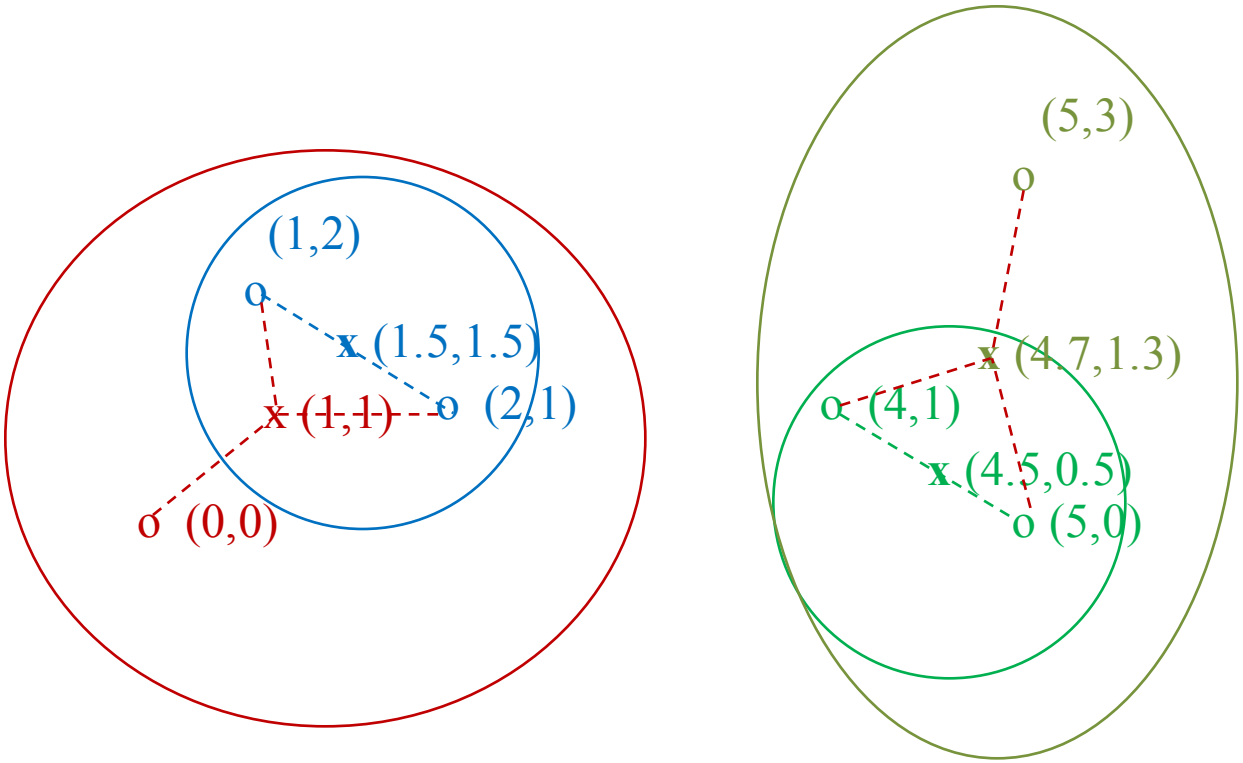
2) After all points are assigned, update the locations of centroids of the ***k*** clusters

3) Reassign all points to their closest centroid

Sometimes moves points between clusters

- **Repeat 2 and 3 until convergence**
  - **Convergence:** Points don't move between clusters and centroids stabilize.

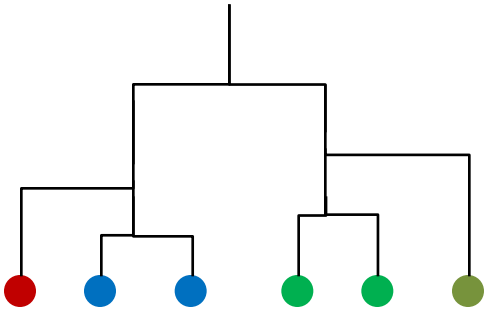
Example: Hierarchical clustering



Data:

o … data point

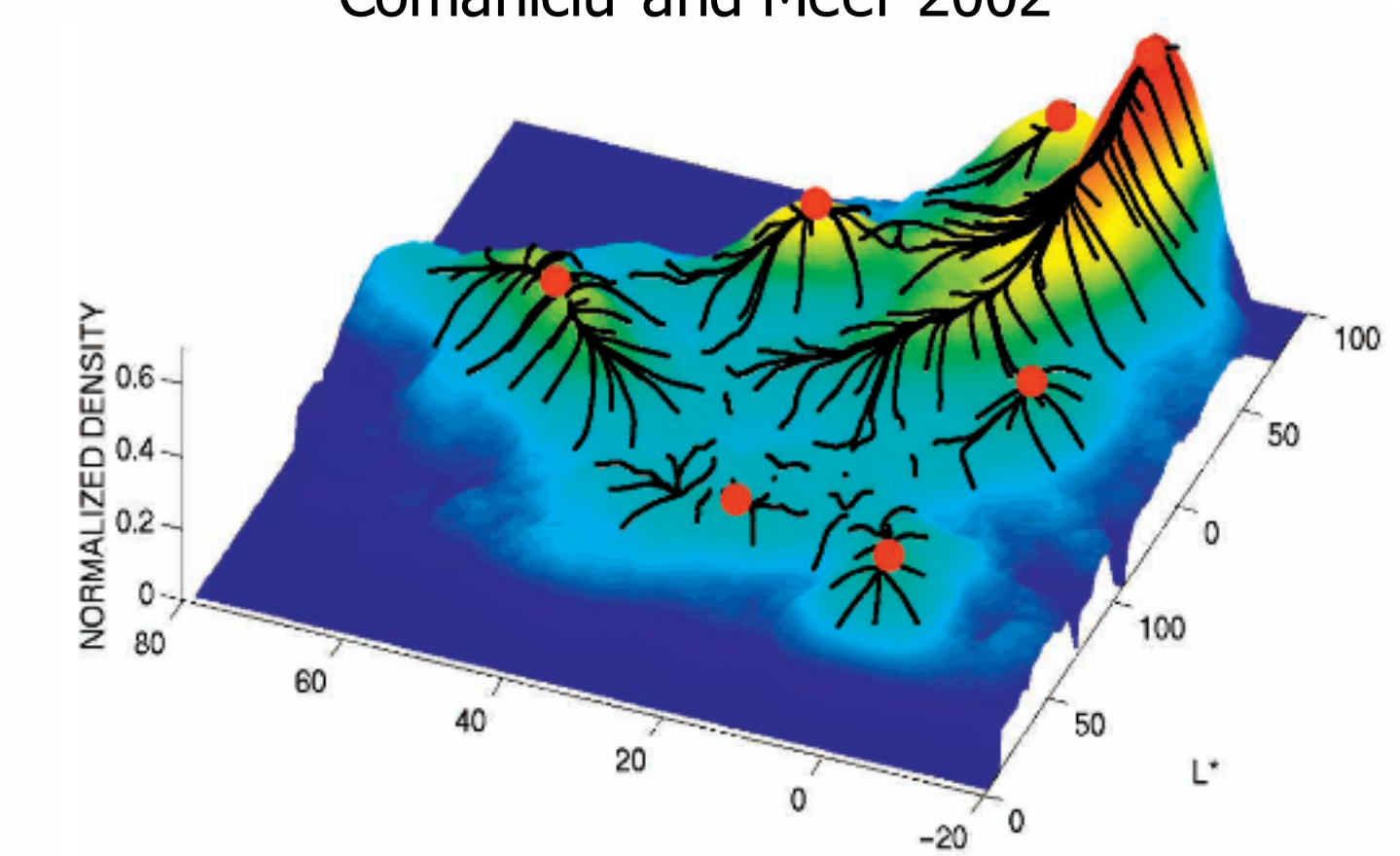
x … centroid



Dendrogram

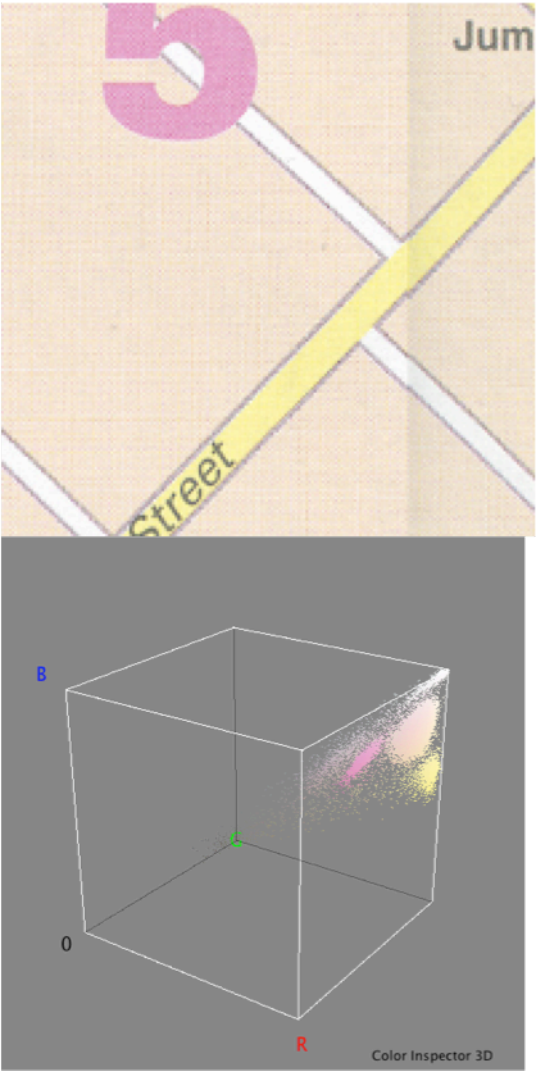
Example: Mean-Shift clustering

Comaniciu and Meer 2002

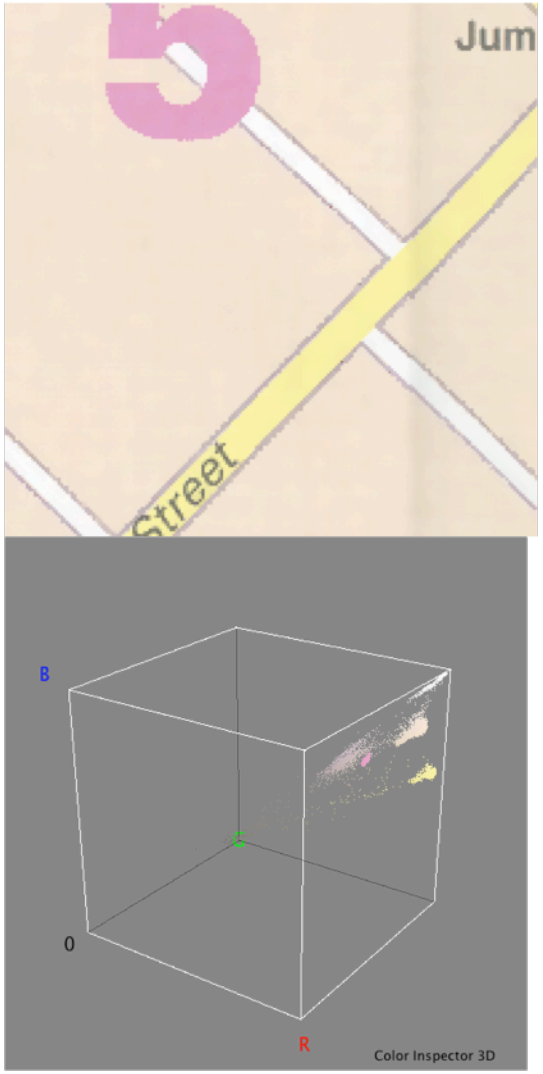


Mean shift is a powerful but computationally expensive method for nonparametric clustering and optimization. It iteratively moves each data point to its local mean until convergence.

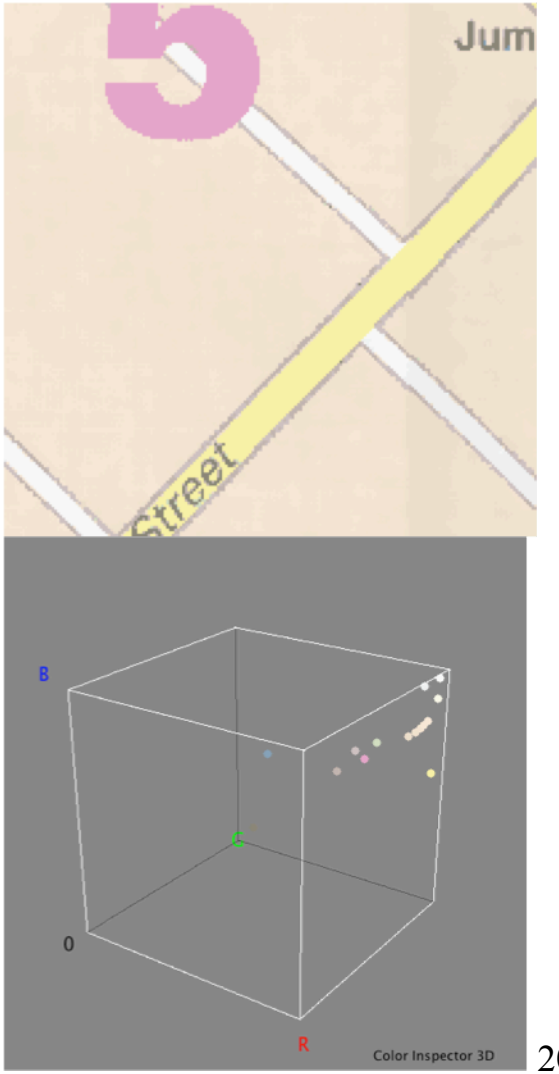
Example: Hybrid Clustering



original



Mean-shift



Mean-shift->K-Means

Clustering CF Algorithms

- **Clustering is an intermediate step**
- Resulting clusters used for further analysis or processing
  - For classification and other tasks
  - Example: **partition data into clusters; then use memory-based CF algorithm like Pearson correlation to make predictions within each cluster**
- Clustering algorithms have **better scalability than typical CF methods** because they **make predictions on smaller clusters rather than whole customer base**
- **Complex and expensive clustering computation run offline**
- **Recommendation quality is generally low**
- Optimal clustering over large data sets is impractical
  - Most applications use greedy cluster generation techniques.

REGRESSION-BASED CF

Regression-based CF Algorithms

- For memory-based CF algorithms: **in some cases, two rating vectors may have:**
  - large Euclidean distance
  - but have very high similarity using vector cosine or Pearson correlation measures **- noise**
- Numerical ratings are common in real recommender systems
- **Regression methods: good at making predictions for numerical values**
- **Uses an approximation of the ratings to make predictions based on a regression model.**

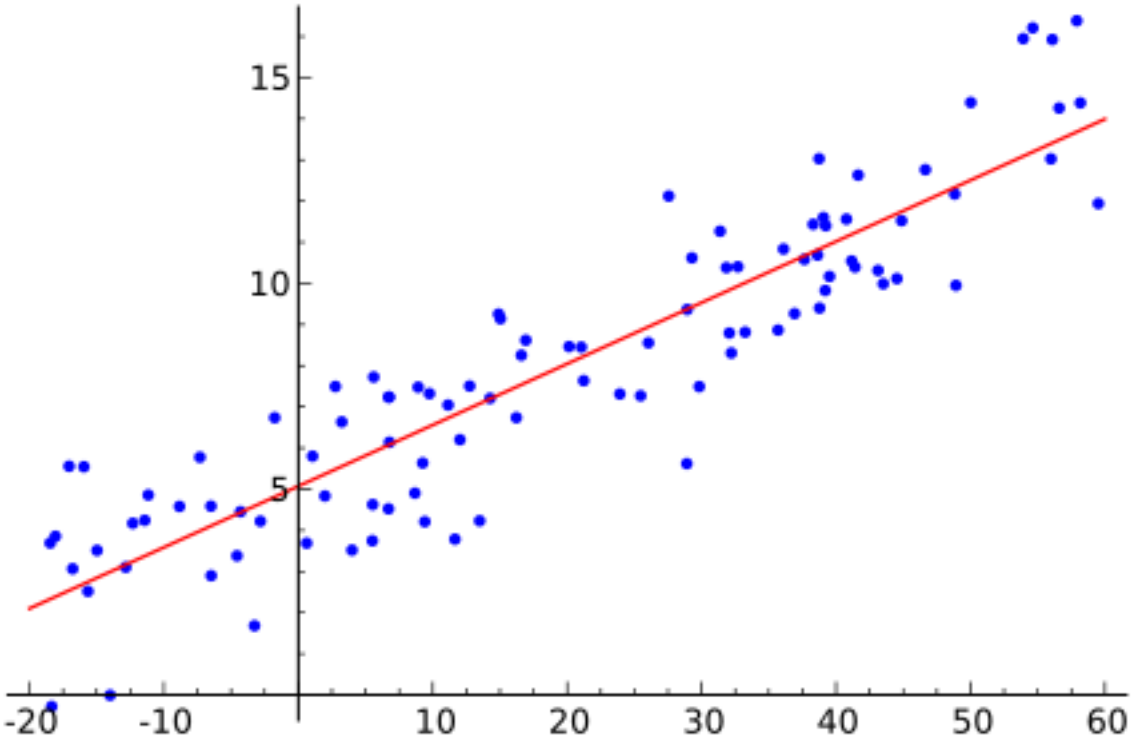
Linear Regression

- Data are modeled using linear predictor functions, and unknown model parameters are estimated from the data
- Such models are called *linear models*
- If the goal is prediction, forecasting, or reduction, linear regression can be used to **fit a predictive model to an observed data set of *y* and *X* values**
- After developing such a model, if an additional value of *X* is then given without its accompanying value of *y*
  - the fitted model can be used to **make a prediction of the value of *y*.**

Regression method

- Let **X = (X1, X2, …, XN)** be a random variable representing **user's preference** on different items
- Linear regression method:
  - **Y = ΛX + N**
  - Where **Λ** is an **n x k** matrix
  - **N = (N_1, …., N_n)** is a random variable representing **Noise** in user choices
  - **Y** is an **n x m** matrix where **Y_{ij} is rating of user i on item** j (typically very sparse)
  - **X** is a **k x m** matrix with each column as estimate of the value of the random variable X (user's rating in k-dimensional rating space for one user).

Example: Linear Regression



Source: https://en.wikipedia.org/wiki/Linear_regression

CHARACTERISTICS AND CHALLENGES OF COLLABORATIVE FILTERING

Characteristics and Challenges of Collaborative Filtering

- **Data Sparsity**
  - Many commercial recommender systems are used with very large product sets
  - Most users do not rate most items: User-item matrix is extremely sparse
  - For CF: reduces probability of finding set of users with similar ratings
- **Approaches:**
  - **Dimensionality reduction techniques**
    - **Singular Value Decomposition (SVD):** remove unrepresentative or insignificant users or items to reduce size of user-item matrix
    - **Latent semantic Indexing:** similarity between users is determined by representation of users in reduced space
    - **Principle Component Analysis.**

Characteristics and Challenges of Collaborative Filtering

- **Data Sparsity (cont.)**
  - **Dimensionality reduction techniques**
  - **When users or items are discarded:**
    - **Useful information for recommendations may be lost**
    - **Recommendation quality may be degraded.**

Challenges (cont.)

- **Cold start problem**
  - **When a new user or item has just entered the system**
    - Hard to find similarities: not enough information to make good recommendations
  - **New item problem:** can't be recommended until some users rate it
    - Also applies to obscure items
    - Also **called "first-rater problem"**
  - **New users**: not given good recommendations because of lack of rating or purchase history
- **Approaches:**
  - **Content-based systems** do not rely on ratings from other users
  - **Hybrid CF (content-boosted CF)**: external content information can be used to produce predictions for new users or new items
  - Research on **effectively selecting items to be rated by a user to rapidly improve recommendation performance.**

Amazon Vine


amazon
Try Prime
All
Shop by Department
Your Amazon.com
Today's Deals
Gift Cards
Sel

What is Amazon Vine?

Amazon Vine invites the most trusted reviewers on Amazon to post opinions
rank, which is a reflection of the quality and helpfulness of their reviews as ju
independent opinions of the Vine Voices. The vendor cannot influence, modif
Customer review from the Amazon Vine Program.

Why do you have the Amazon Vine program?

The program was created to provide customers with more information includ

How can I join the program?

Amazon Vine is an invitation-only program. Vine Voices are selected based o
in the program. Customers who consistently write helpful reviews and develo

How are Vine Voices selected?

We want the Voice program to reflect the best of our growing body of custom
their reviews, factoring in the number of reviews they have written. More we

How To Become An Amazon Vine Reviewer & Get Free Stuff


Written by Bakari Chavanu
August 4, 2010

If you're an Amazon.com customer who regularly reads product reviews by other customers, you may have noticed that some reviewers have a little badge (one of several) under their name identifying them as a ï»¿"Vine Voice" reviewer.

These Amazon Vine reviewers were invited by Amazon to receive a monthly newsletter of pre- and recently-released books and other products that Vine members can select from and review. Selected products are sent to Amazon Vine reviewers for free. So how do you become a Vine reviewer?

I'm not a prolific reviewer on Amazon, but about six months ago I was invited to become a member of its "exclusive club of influential Amazon voices."

Getting free items such as a 1 TB Western Digital External Hard Drive, a Duracell iPhone charger, a very expensive photography bag, computer software, and numerous books in exchange for honest reviews of these products has been a pretty good deal in my view.

Challenges (cont.)

- **Synonyms**
  - **Same or very similar items that have different names or entries**
  - Most recommender systems are unable to discover this latent association
  - Treat these products differently
  - **Synonyms decrease recommendation performance of CF systems**
- Approaches
  - **Automatic term expansion** or construction of thesaurus
    - Some added terms may have different meanings than intended
  - **SVD techniques: Latent Semantic Indexing (LSI)**: construct a semantic space where terms and documents that are closely associated are placed close to each other.

Example: Latent Semantic Indexing (LSI)

Searches related to usc

usc **next coach**

university of southern california notable alumni

usc **address**

usc **clothing**

usc **tuition**

usc **ranking**

usc **jobs**

usc **athletics**

Searches related to ucla

ucla **vs stanford predictions**

ucla **football**

ucla **requirements**

ucla **tuition**

ucla **ranking**

ucla **admissions**

ucla **majors**

ucla **medical center**

Challenges (cont.)

- **Scalability**
  - **Traditional CF systems suffer scalability problems at very large scale**
  - With tens of millions of customers (M), millions of catalog items (N): and ***O(n)* algorithm is too large**
- Approaches
  - **Dimensionality reduction (SVD)** can scale and quickly produce good recommendations
    - But have to do expensive matrix factorization
  - Memory-based CF algorithms (e.g., **item-based Pearson correlation CF algorithm**) have good scalability
    - Instead of calculating similarities between all pairs of items, **calculate similarity only between pairs of co-rated items by a user.**

Example SVD

$$\begin{bmatrix} 5 & 2 & 4 & 4 & 3 \\ 3 & 1 & 2 & 4 & 1 \\ 2 & & 3 & 1 & 4 \\ 2 & 5 & 4 & 3 & 5 \\ 4 & 4 & 5 & 4 & \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \\ u_{41} & u_{42} \\ u_{51} & u_{52} \end{bmatrix} \times \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} \end{bmatrix}$$

Figure 9.9: UV-decomposition of matrix M

Challenges (cont.)

- **Gray Sheep**
  - Users whose opinions **do not consistently agree or disagree with any group of people**
  - Do not benefit from collaborative filtering
- Approaches:
  - **Hybrid approach combining content-based and CF recommendations**
  - Base prediction on **weighted average of content-based prediction and CF prediction**
  - Weights are determined on a per-user basis
  - System determines optimal mix of content and CF-based recommendations
- **Black sheep**
  - **Idiosyncratic tastes make recommendations nearly impossible: considered an acceptable failure**

Challenges (cont.)

- **Shilling attacks**
- Shill definition
  - **Noun:** an accomplice of a hawker, gambler, or swindler who acts as an enthusiastic customer to entice or encourage others
  - **Verb:** act or work as a shill
- **In systems where anyone can provide ratings (Yelp, Amazon):**
  - People may give many positive ratings for their own materials
  - Negative recommendations for competitors
- CF systems want to discourage this phenomenon
- Approaches (research)
  - **Item-based less affected than user-based**
  - **Hybrid CF systems provide partial solutions.**

Pros/Cons of Collaborative Filtering

+ Works for any kind of item

No feature selection needed

- Cold Start:

Need enough users in the system to find a match

- Sparsity:

The user/ratings matrix is sparse

Hard to find users that have rated the same items

- First rater:

Cannot recommend an item that has not been previously rated

New items, Esoteric items

- Popularity bias:

Cannot recommend items to someone with unique taste

Tends to recommend popular items.