COMP20008 Elements of Data Processing

THE UNIVERSITY OF MELBOURNE

Data Pre-Processing and Cleaning: Recommender Systems and Missing Values

Plan today

- Complete section on outlier detection
- Recommender systems and collaborative filtering
- Types of similarity for imputation of missing values
  - Item-Item
  - User-User
- Question to consider during lecture: Are we doing cleaning or prediction?

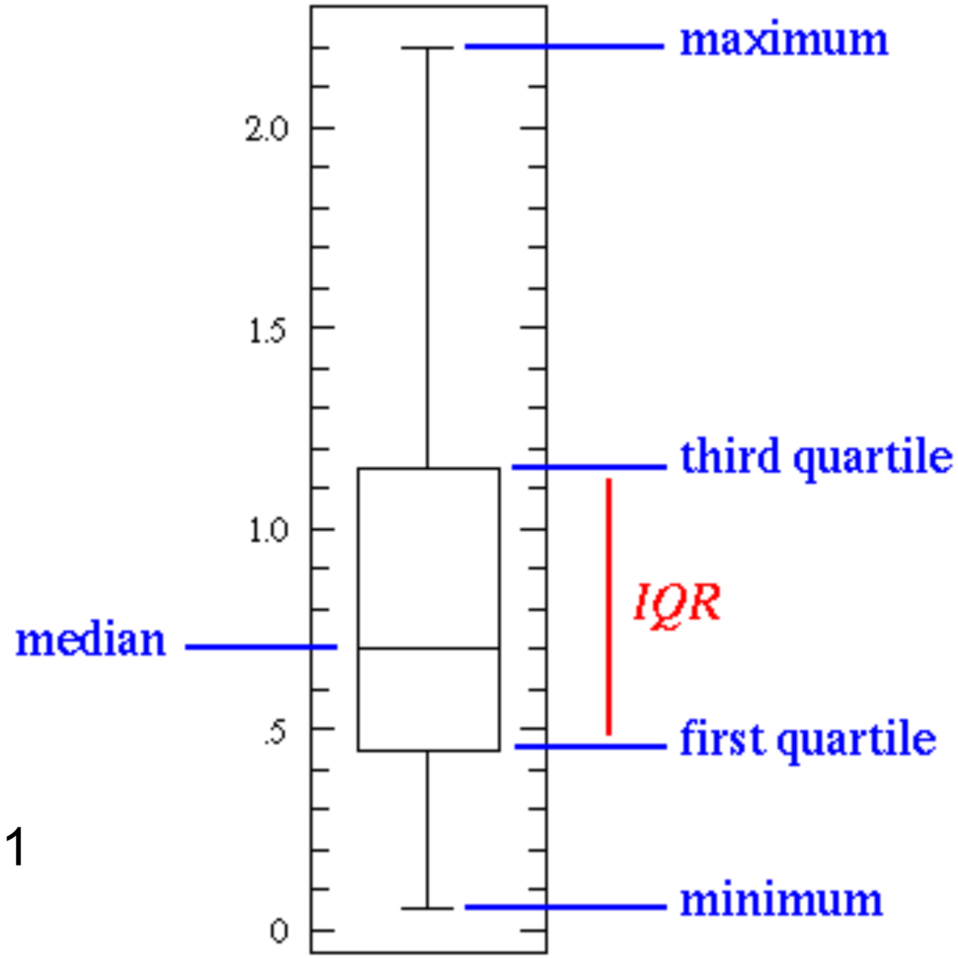
How to detect outliers

- 1-D data
  - Boxplot
  - Histogram
  - Statistical tests
- 2-d Data: Scatter plot and eyeball
- 3-D data: Can also use scatter plot and eyeball
- >3-D data: Statistical or algorithmic methods

Box and whisker plot (diagram from http://www.physics.csbsju.edu/stats/box2.html)

From sample compute

- Minimum and maximum (the whiskers)
- Median
- First quartile(Q1): middle number between median and minimum
- Third quartile(Q3): middle number between median and maximum
- IQR=interquartile range =Q3-Q1



Outliers and Tukey Boxplots(diagram from http://www.physics.csbsju.edu/stats/box2.html)

Whiskers

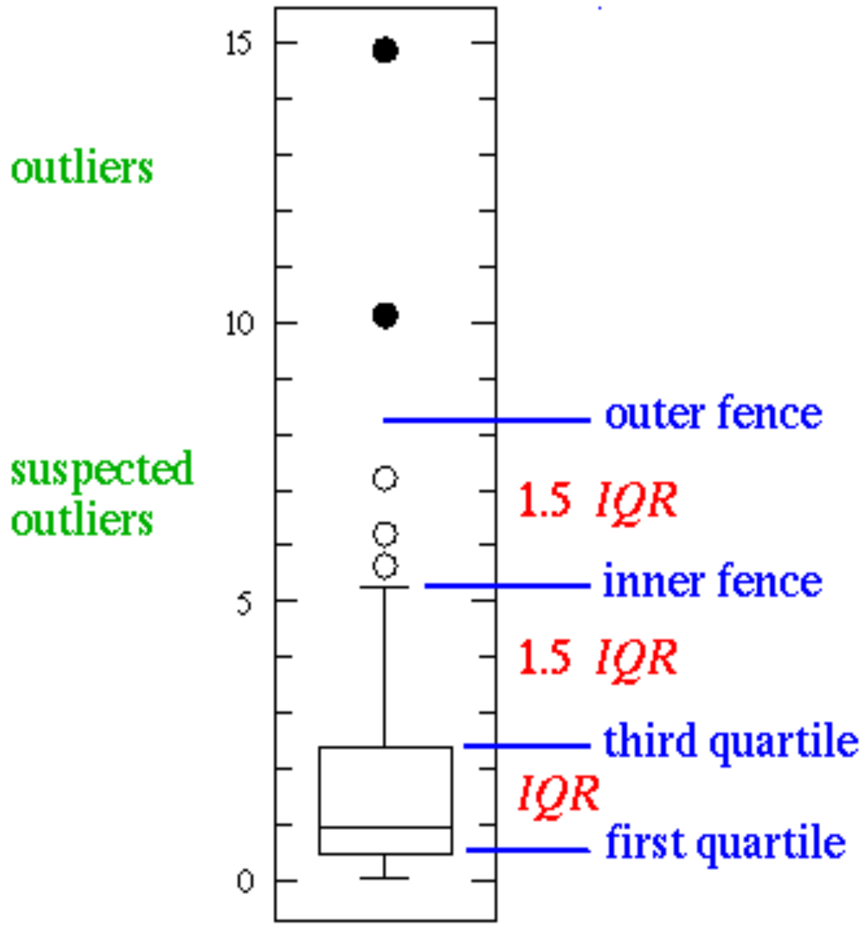
- Lowest point still within 1.5IQR o lower quartile
- Highest point still within 1.5 IQR of upper quartile

Outliers (filled black)

- >3*IQR above third quartile, or
- >3*IQR below 1st quartile

Suspected outliers (open black)

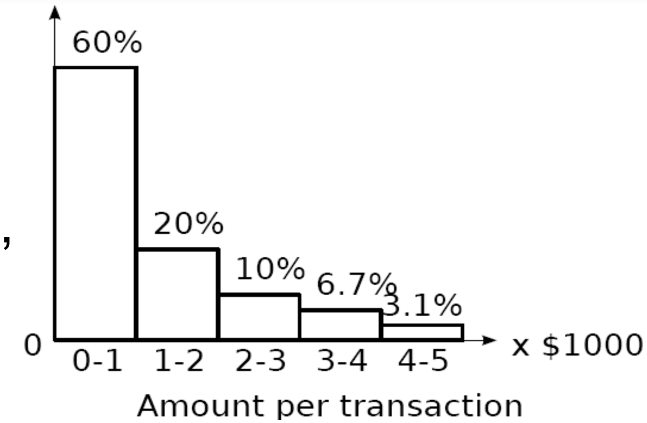
- >1.5*IQR above third quartile, or
- >1.5*IQR below 1st quartile



Example of box plot outlier detection

- Example from
  - http://www.alcula.com/calculators/statistics/box-plot
  - 10,20,30,40,50,60,70,80,90,100,120,130,140,150,160,180,999

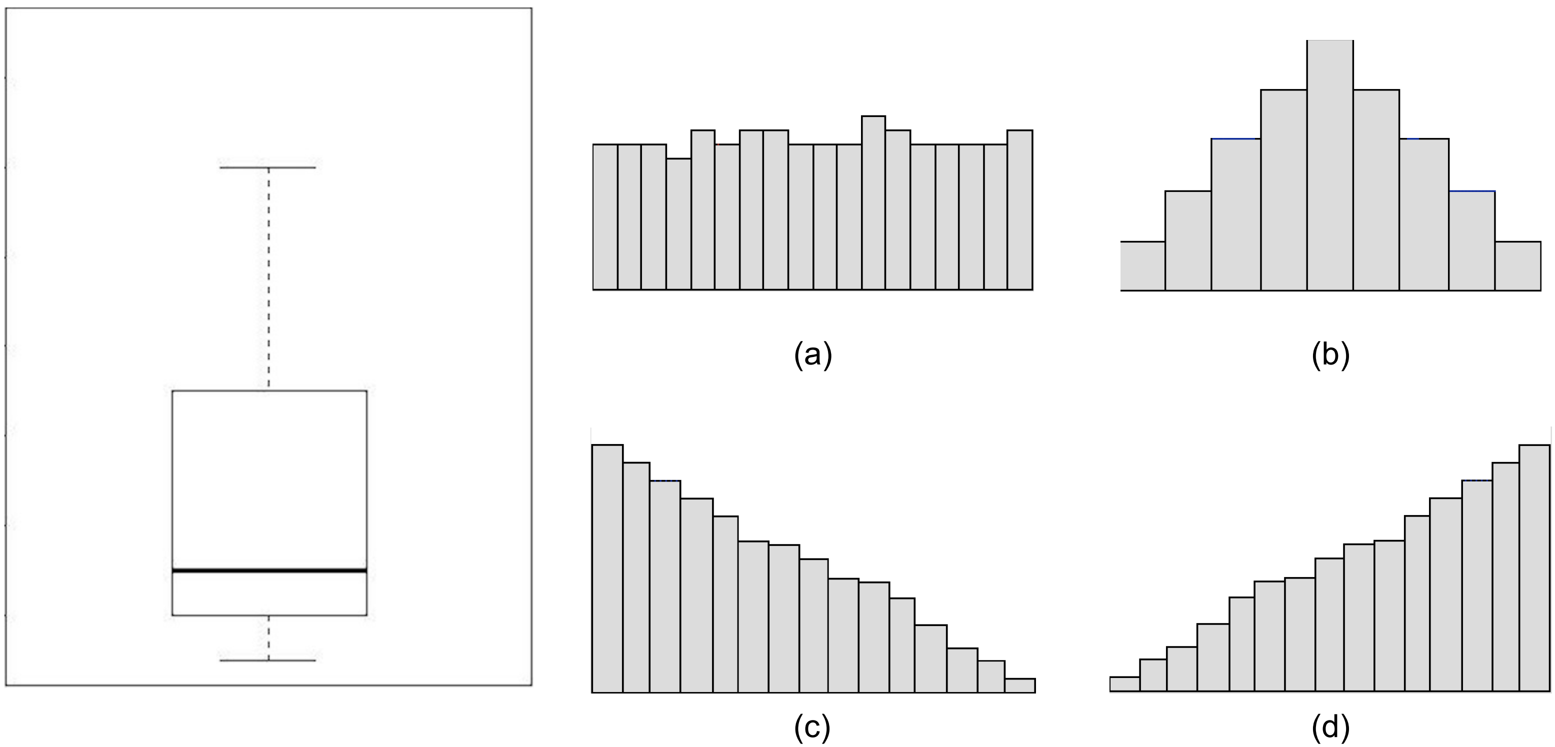
Detection Using Histogram



- The model of normal data is learned from the input data without any *a priori* structure.
- Often makes fewer assumptions about the data, and thus can be applicable in more scenarios
- Outlier detection using histogram:
  - Figure shows the histogram of purchase amounts in transactions
  - A transaction in the amount of $7,500 is an outlier, since only 0.2% transactions have an amount higher than $5,000
- Problem: Hard to choose an appropriate bin size for histogram
  - Too small bin size → normal objects in empty/rare bins, false positive
  - Too big bin size → outliers in some frequent bins, false negative

Exercise

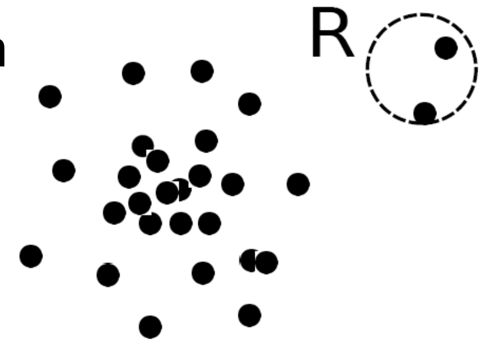
- Which histogram is the best representation of the boxplot?



Outlier Detection (1): Statistical Methods

- Statistical methods assume that the normal data follow some statistical model
  - The data not following the model are outliers.

- Example (right figure): First use Gaussian distribution to model the normal data
  - For each object y in region R, estimate $g_D(y)$, the probability of y fits the Gaussian distribution
  - If $g_D(y)$ is very low, y is unlikely generated by the Gaussian model, thus an outlier



- Effectiveness of statistical methods: highly depends on whether the assumption of statistical model holds in the real data
- There are rich alternatives to use various statistical models

Univariate case: Grubb's Test

- Univariate outlier detection: Detect one outlier at a time and repeat.
  - Compute the following statistic where x_i is a data instance

$$\frac{\max_{i=1,\ldots,N} |x_i - \mu|}{\sigma}$$

where μ is the sample mean and

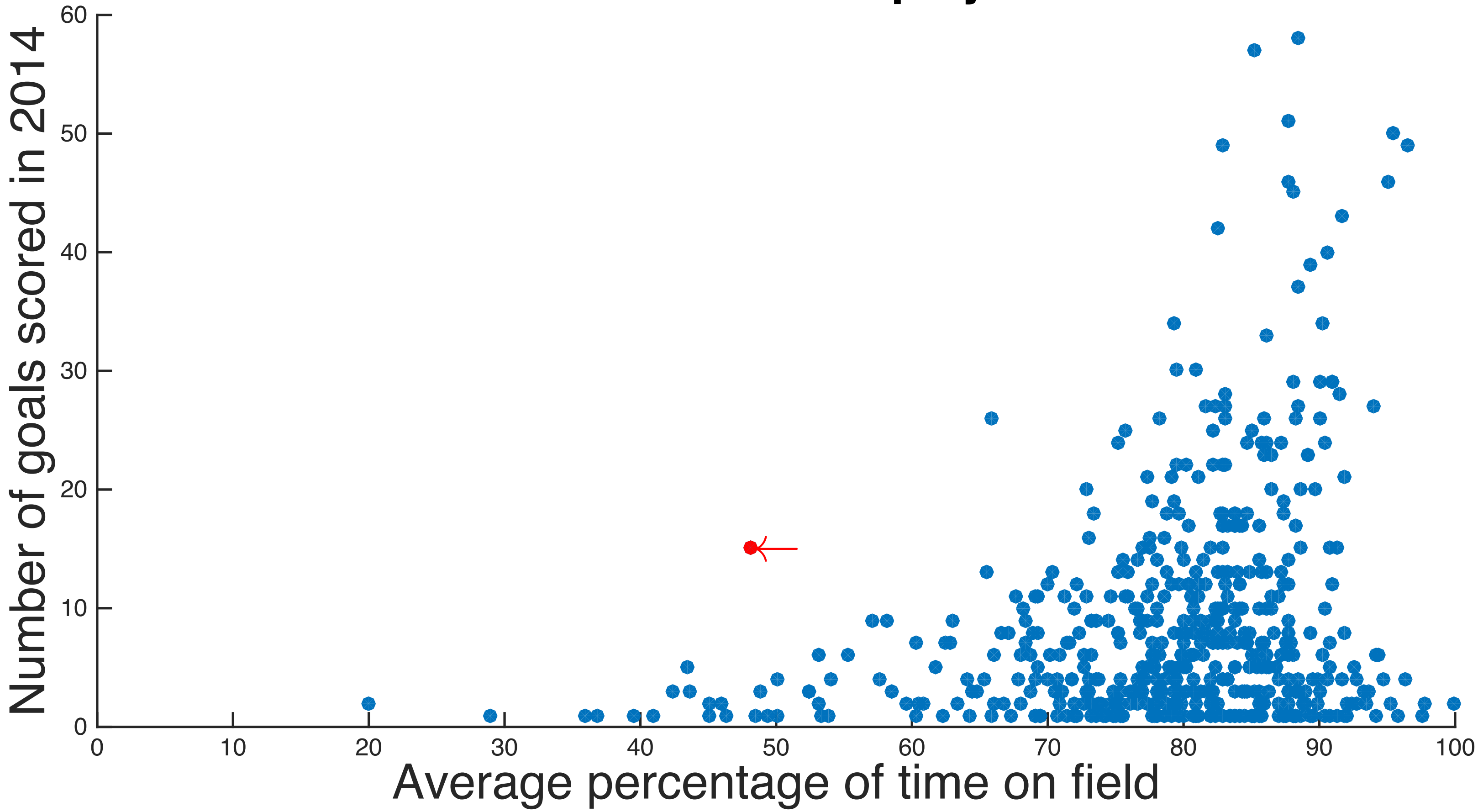
σ is the sample standard deviation

Then assume population is normally distributed and do a statistical hypothesis test (Python package outlier_utlis). Farthest point is an outlier if unlikely to have occurred under normal distribution assumption. Throw away outlier if test indicates that instance is "too far" from the mean.

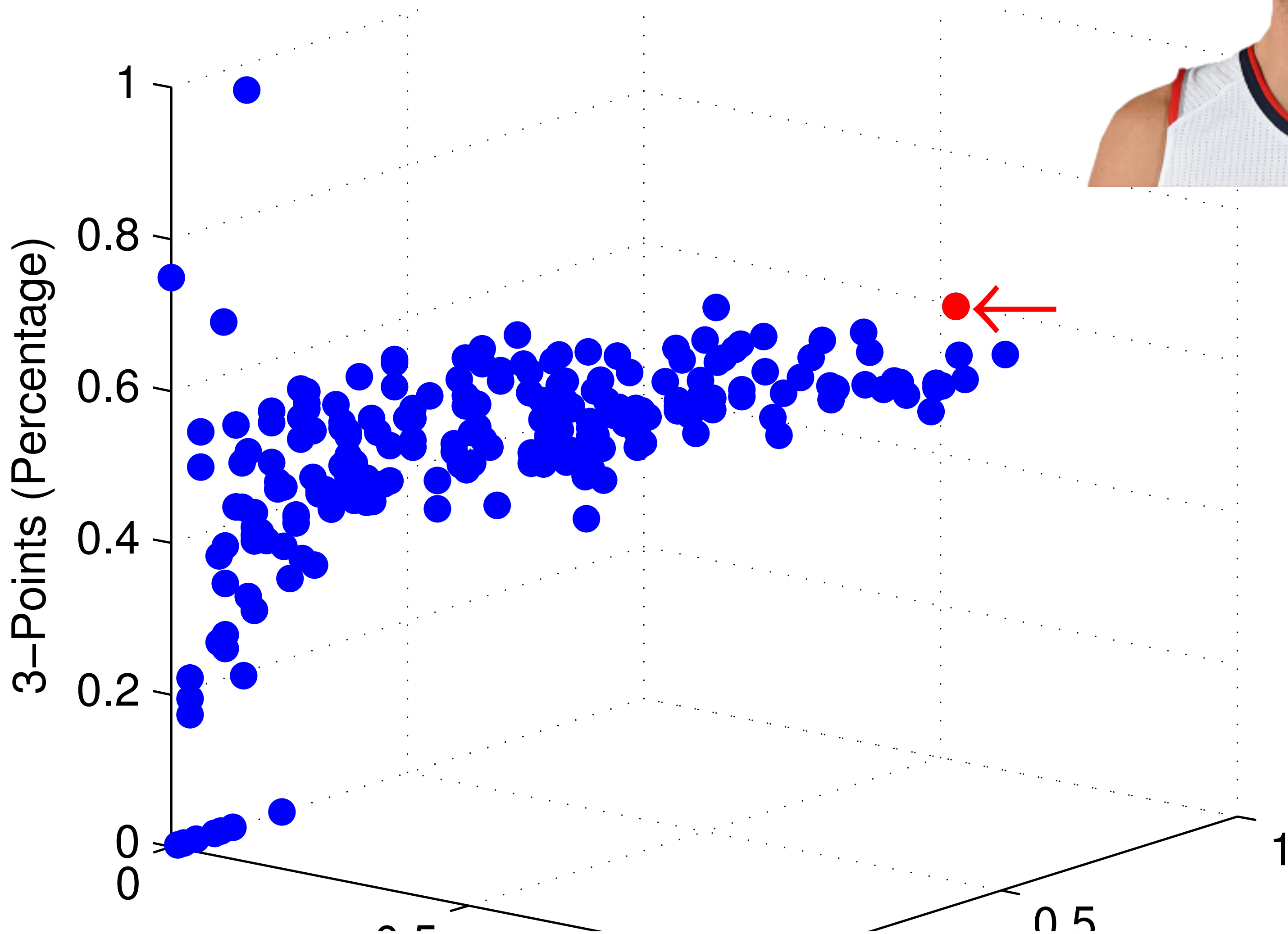
2D scatter plot

- Daniel Giansiracusa

Outlyingness of Daniel Giansiracusa (see arrow) versus 626 other players



3D scatter plot: Kyle Korver
3 points: made, attempted, percentage



Acknowledgements

– Data Mining Concepts and Techniques. Han, Kamber and Pei. 3rd edition (chapter 3 and 12). Available through library as ebook.
– Data analysis using regression and multilevel hierarchical models. Gelman and Hill (chapter 25), 2006.

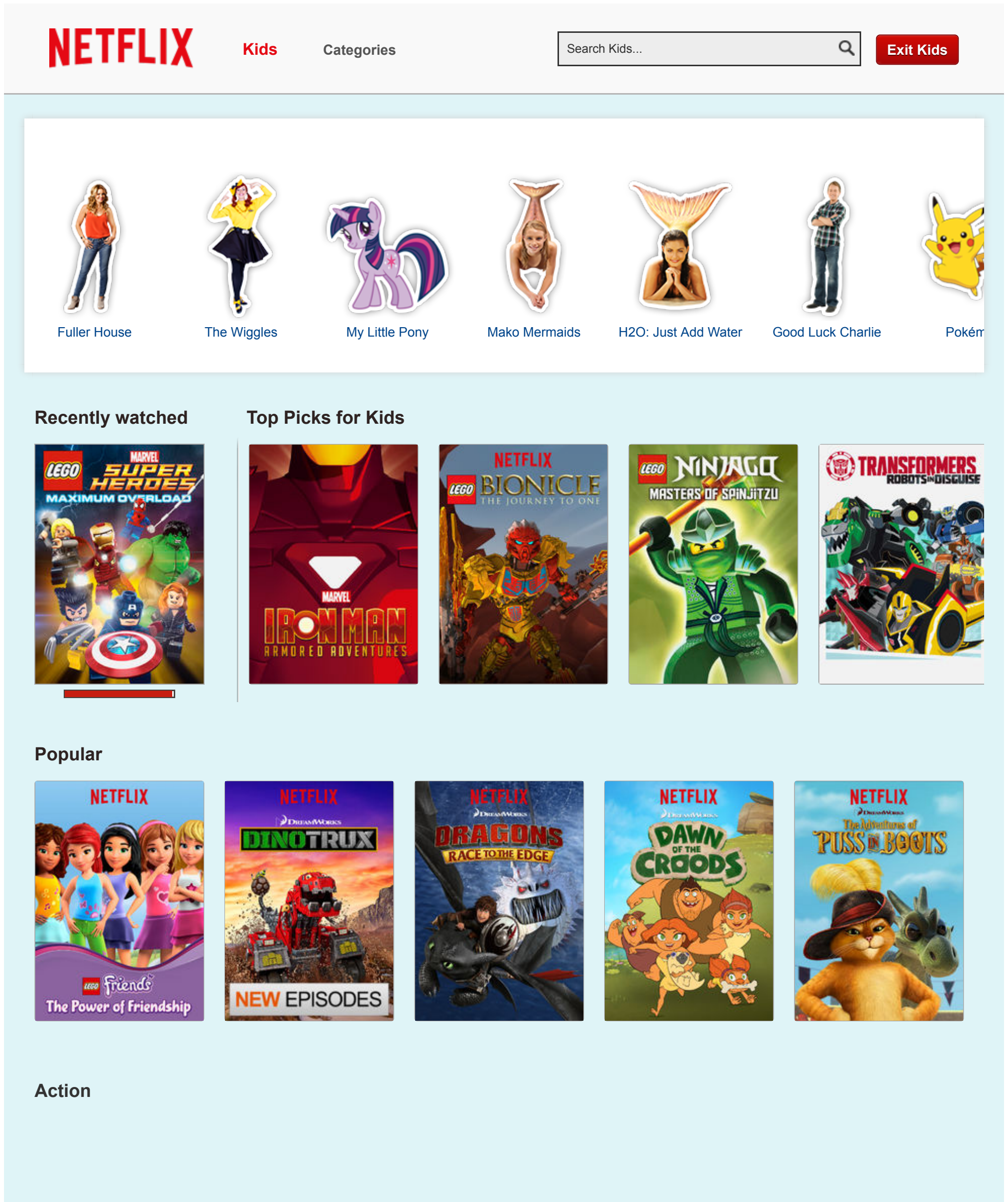
Recommender systems: missing data

- Movie Recommender systems

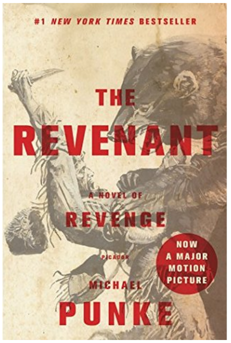
Person	Star Wars	Batman	Jurassic World	The Martian	The Revenant	Lego Movie	Selma	….
James	3	2	-	-	-	1	-	
John	-	-	1	2	-	-	-	
Jill	1	-	-	3	2	1	-	

Users and movies
Each user only rates a few movies (say 1%)
Netflix wants to predict the missing ratings for each user

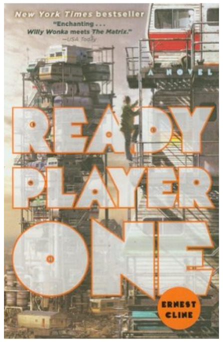
Netflix


NETFLIX
Kids
Categories
Search Kids...
Exit Kids
Fuller House
The Wiggles
My Little Pony
Mako Mermaids
H2O: Just Add Water
Good Luck Charlie
Recently watched
Top Picks for Kids
LEGO MARVEL SUPER HEROES MAXIMUM OVERLOAD
MARVEL IRON MAN ARMORED ADVENTURES
NETFLIX LEGO BIONICLE THE JOURNEY TO ONE
LEGO NINJAGO MASTERS OF SPINJITZU
TRANSFORMERS ROBOTS IN DISGUISE
Popular
NETFLIX LEGO Friends The Power of Friendship
NETFLIX DreamWorks DINOTRUX NEW EPISODES
NETFLIX DreamWorks DRAGONS RACE TO THE EDGE
NETFLIX DreamWorks DAWN OF THE CROODS
NETFLIX DreamWorks The Adventures of PUSS IN BOOTS
Action

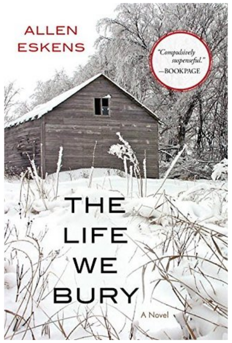
Amazon.com: Customers who bought this item also bought …..



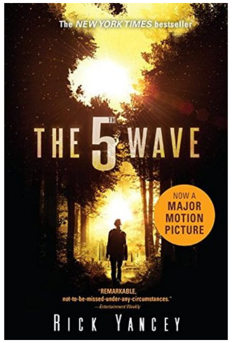
The Revenant: A Novel of Revenge
› Michael Punke
1,250
Paperback
$9.52



Ready Player One: A Novel
› Ernest Cline
9,210
Paperback
$8.37



The Life We Bury
› Allen Eskens
1,896
Paperback
$8.75



The 5th Wave: The First Book of the 5th Wave Series
› Rick Yancey
2,006
Paperback
$6.70

- “75% of what people watch is from some sort of recommendation” (Netflix)

- “If I have 3 million customers on the web, I should have 3 million stores on the web.” (Amazon CEO)

Other Examples of Recommender Systems

- IMDb
- Online dating
- Twitter: “Who to Follow”, what to retweet
- Spotify, youtube: music recommendation
- LinkedIn/Facebook: who to add as a contact, jobs of interest, news of interest
- Tourist attraction apps

- University subjects … ? Subject discussion forums … ?

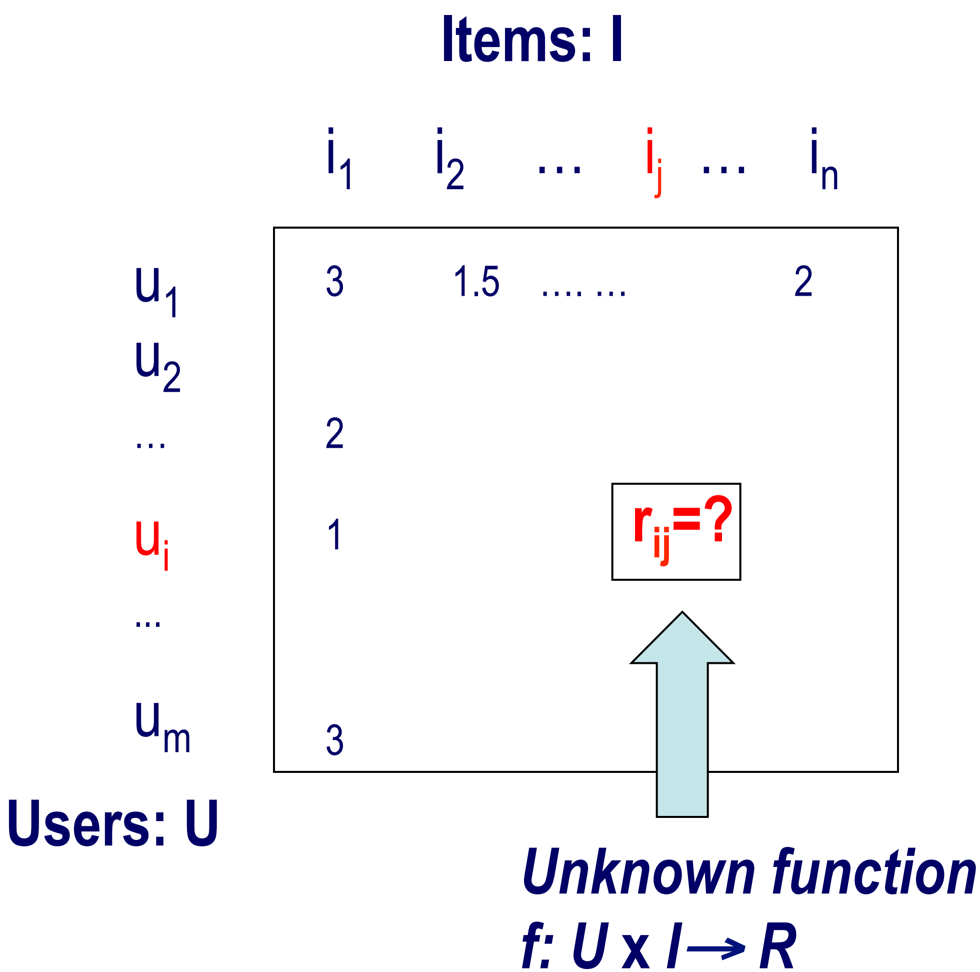
How it works

- Each user has a profile
- Users rate items
  - Explicitly: Give a score
  - Implicitly: web usage mining
    - Time spent in viewing the item
    - Navigation path
    - Etc…
- System does the rest, How?

Collaborative filtering

- *Collaborative Filtering: Make predictions about a user's missing data according to the behaviour of many other users*
  - Look at users collective behavior
  - Look at the active user history
  - Combine!

Collaborative Filtering: A Framework



The task:

Q1: Find Unknown ratings?

Q2: Which items should we recommend to this user?

.

.

.

Collaborative Filtering Approaches

- User based methods
  - Identify like-minded users
- Item based methods
  - Identify similar items
- Model (matrix) based methods
  - Solve an optimization problem and identify latent factors

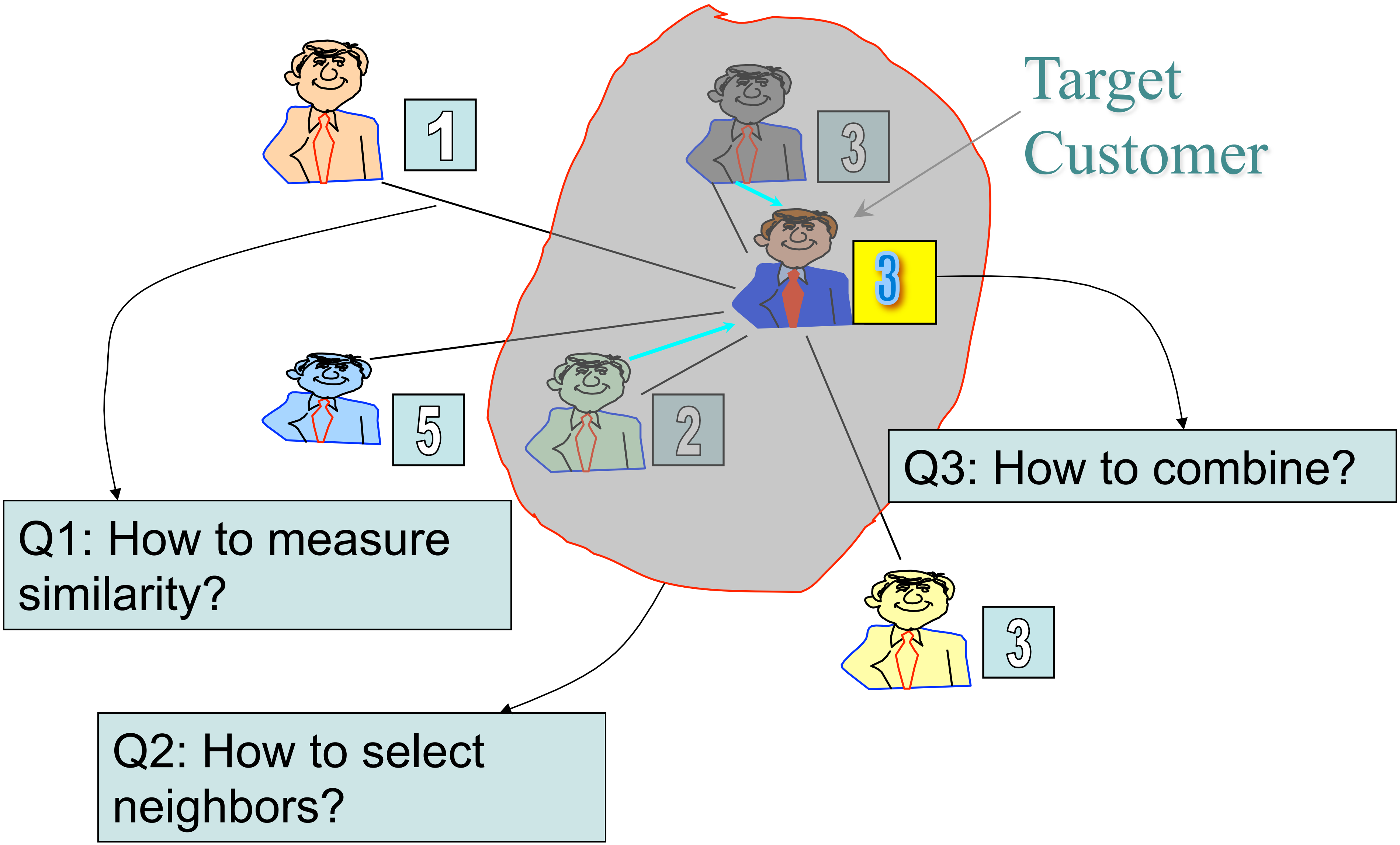
Ratings of items by users: Fill in cell ????

Items

Users

	Item1	Item2	Item3	Item4	Item5	Item6
User1	17	-	20	18	17	18.5
User2	8	-	**????**	17	14	17.5
User3	-	-	17	18	18.5	17.5
User4	-	-	-	18	17.5	18
User5	17	-	18	19	15.5	-
User6	-	-	17.5	-	16	-
User7	15	17.5	-	17	-	17
User8	18	-	-	-	17	16.5
User9	18	17	-	-	18.5	17
User10	19	17	-	-	-	16.5
User11	17	18.5	19	19	-	-
User12	14	19	17	-	-	15.5
User13	-	16	-	-	17	-
User14	20	18.5	-	18	-	18

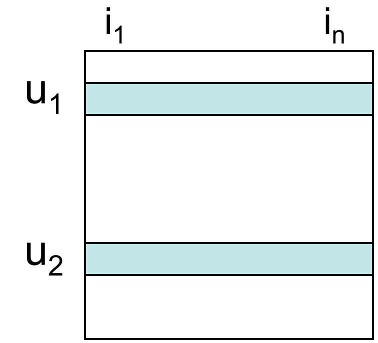
User-User Similarity: Intuition



THE UNIVERSITY OF MELBOURNE

How to Measure Similarity? Method 1

U1	**17**	**-**	**20**	**18**	**17**	**18.5**
U2	**8**	**-**	**-**	**17**	**14**	**17.5**



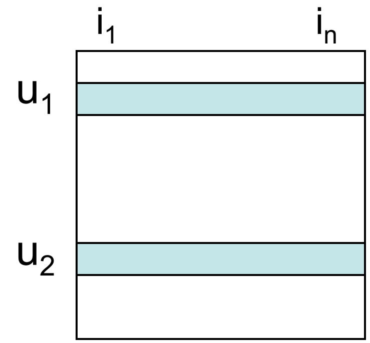
$$SIM(U1, U2) =$$

$$((17-8)^2 + (18.1-14.1)^2 + (20-14.1)^2 + (18-17)^2 + (17-14)^2 + (18.5-17.5)^2$$

- Compute mean value for User1's missing values (18.1)
- Compute mean value for User2's missing values (14.1)
- Compute squared Euclidean distance between resulting vectors

How to Measure Similarity? Method 2

User1	**17**	**-**	**20**	**18**	**17**	**18.5**
User2	**8**	**-**	**-**	**17**	**14**	**17.5**



$$Sim(User1, User2) = \frac{6}{6-2}((17-8)^2 + (18-17)^2 + (17-14)^2 + (18.5-17.5)^2$$

- Compute squared Euclidean distance between vectors, summing only pairs without missing values
- 2 out of the 6 pairs have at least one missing value
- Scale the result, according to percentage of pairs with a missing value

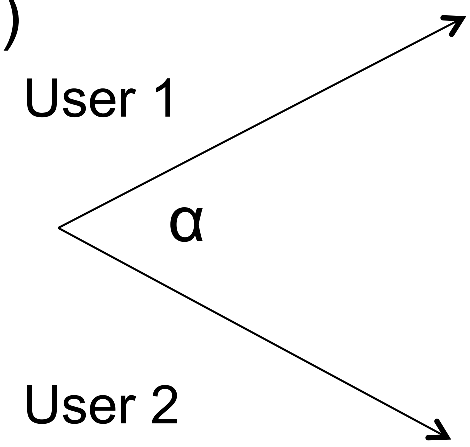
Practice Example

User1	**12**	**2.5**	**20**	**-**	**17**	**-**	**3.5**
User2	**13**	**-**	**-**	**17**	**14**	**17.5**	**4.5**

Using Method 2, SIM(User1,User2)=?

Similarity

- Instead of Euclidean distance can also use other measures to assess similarity, e.g.
  - Correlation (we will look at later in subject)
  - Cosine similarity (angle between user profile vectors)



Selecting neighbors and making prediction

- At runtime
  – Need to *select* users to compare to
  – Could choose the top-k most similar users
  – *Combining*: Prediction of rating is the (weighted) average of the values from the top-k similar users
- Can make more efficient by computing clusters of users offline
  – At runtime find nearest cluster and use the centre of the cluster as the rating prediction
  – Faster (more scalable) but a little less accurate

User based methods summary

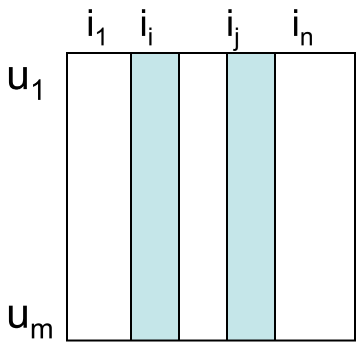
- Achieve good quality in practice
- The more processing we push offline, the better the method scale
- However:
  - User preference is dynamic
    - High update frequency of offline-calculated information
  - No recommendation for new users
    - We don’t know much about them yet

Item based methods: Intuition

- Search for similarities among items
- All computations can be done offline
- Item-Item similarity is more stable that user-user similarity
  - No need for frequent updates

Item Based Methods

- Same as in user-user similarity but on item vectors
  - Find similar items to the one whose rating is missing
  - E.g. For item i_i compute its similarity to each other item i_j



Item based similarity

- Offline phase. For each item
  - Determine its k-most similar items
  - Can use same type of similarity as for user-based
- Online phase:
  - Predict rating r_{aj} for a given user-item pair as a weighted sum over k-most similar items that they rated

$$r_{aj} = \frac{\sum_{i \in \text{k-similar items}} sim(i,j) \times r_{ai}}{\sum_{\in \text{k-similar items}} sim(i,j)}$$

User a	**8**		**r_{aj}**		**9**	**15**

Item j

Items

Users

	Item1	Item2	Item3	Item4	Item5	Item6
User1	17	-	20	18	17	18.5
User2	8	-	????	17	14	17.5
User3	-	-	17	18	18.5	17.5
User4	-	-	-	18	17.5	18
User5	17	-	18	19	15.5	-
User6	-	-	17.5	-	16	-
User7	15	17.5	-	17	-	17
User8	18	-	-	-	17	16.5
User9	18	17	-	-	18.5	17
User10	19	17	-	-	-	16.5
User11	17	18.5	19	19	-	-
User12	14	19	17	-	-	15.5
User13	-	16	-	-	17	-
User14	20	18.5	-	18	-	18

Matrix Based Techniques

- Treat the User-Item Rating table R as a matrix
  - Use matrix factorisation of this Rating Table

Rating Table R

Items

Users

	Item1	Item2	Item3	Item4	Item5	Item6
User1	17	-	20	18	17	18.5
User2	8	-	-	17	14	17.5
User3	-	-	17	18	18.5	17.5
User4	-	-	-	18	17.5	18
User5	17	-	18	19	15.5	-
User6	-	-	17.5	-	16	-
User7	15	17.5	-	17	-	17
User8	18	-	-	-	17	16.5
User9	18	17	-	-	18.5	17
User10	19	17	-	-	-	16.5
User11	17	18.5	19	19	-	-
User12	14	19	17	-	-	15.5
User13	-	16	-	-	17	-
User14	20	18.5	-	18	-	18

Factorisation

- We are familiar with factorisation of numbers

  15=3*5

  99=3*33

  1000=10*100

  We can also do approximate factorisation

  17 ≈ 6*2.8 (RHS= 16.8, an error of 0.2)

  167 ≈ 17*9.8 (RHD=166.6, an error of 0.4)

Matrix Factorization

Given a matrix R, we can find matrices U and V such that when U and V are multiplied together

$$R \approx UV$$

- R is m*n, U is m*k and V is k*n
  - k is the "number of latent factors"

For example, suppose R is a 4*4 matrix

$$R = \begin{bmatrix} 5 & 2 & 3 & 6 \\ 4 & 4 & 6 & 11 \\ 3 & 19 & 2 & 7 \\ 3 & 8.5 & 4 & 2 \end{bmatrix}$$

Example: m=4, n=4, k=2

$$\begin{bmatrix} 5 & 2 & 3 & 6 \\ 4 & 4 & 6 & 11 \\ 3 & 19 & 2 & 7 \\ 3 & 8.5 & 4 & 2 \end{bmatrix} \approx \begin{bmatrix} 0.34776 & 1.97802 \\ 0.71609 & 3.13615 \\ 4.27876 & 0.58287 \\ 1.88074 & 0.56923 \end{bmatrix} \begin{bmatrix} 0.58367 & 4.40189 & 0.44605 & 1.04492 \\ 1.52915 & 0.26346 & 1.75046 & 3.09976 \end{bmatrix}$$

$$= \begin{bmatrix} 3.22769 & 2.05196 & 3.61758 & 6.49480 \\ 5.21363 & 3.97844 & 5.80912 & 10.46959 \\ 3.3887 & 18.98823 & 2.92886 & 6.27777 \\ 1.96819 & 8.42882 & 1.83534 & 3.72973 \end{bmatrix}$$

We can compute the error (squared distance between R and UV). The smaller it is, the better the fit of the factorisation.

$$(5 - 3.22769)^2 + (2 - 2.05196)^2 + (3 - 3.61758)^2 + \ldots$$
$$(4 - 1.83534)^2 + (2 - 3.72973)^2$$

How to factorise

- *Details of how to compute the matrix factorisation are beyond the scope of our study.*
- Intuitively, factorisation algorithms search over lots of choices for U and V, with the aim of making the error as low as possible
- If there are missing values in R, ignore these when computing the error.

Factorisation and missing values

$$\begin{bmatrix} 5 & - & - & 6 \\ - & 4 & 6 & 11 \\ - & 19 & 2 & 7 \\ 3 & 8.5 & - & - \end{bmatrix} \approx \begin{bmatrix} 1.51261 & 1.65457 \\ -0.0474 & 3.56317 \\ 3.88351 & 1.50482 \\ 1.76637 & 0.56005 \end{bmatrix} \begin{bmatrix} 1.07179 & 4.42771 & -0.13516 & 0.60378 \\ 2.01538 & 1.18272 & 1.67926 & 3.08647 \end{bmatrix}$$

$$= \begin{bmatrix} 4.95572 & 8.65430 & 2.57402 & 6.02008 \\ 7.13025 & 4.00394 & 5.98995 & 10.96899 \\ 7.19512 & 18.97488 & 2.00210 & 6.98942 \\ 3.02190 & 8.48338 & 0.70173 & 2.79509 \end{bmatrix}$$

$$\text{Error} = (5 - 4.95572)^2 + (6 - 6.02008)^2 + (4 - 4.00394)^2 + (6 - 5.98995)^2 + \ldots$$

The product of the two factors U and V, has no missing values. We can use this to predict our missing entries. E.g. R_{12}=8.65430

Using k=2 for factorisation

Items

Users

	Item1	Item2	Item3	Item4	Item5	Item6
User1	17	-	20	18	17	18.5
User2	8	-	**13.48**	17	14	17.5
User3	-	-	17	18	18.5	17.5
User4	-	-	-	18	17.5	18
User5	17	-	18	19	15.5	-
User6	-	-	17.5	-	16	-
User7	15	17.5	-	17	-	17
User8	18	-	-	-	17	16.5
User9	18	17	-	-	18.5	17
User10	19	17	-	-	-	16.5
User11	17	18.5	19	19	-	-
User12	14	19	17	-	-	15.5
User13	-	16	-	-	17	-
User14	20	18.5	-	18	-	18

- Real answer for (User 2, Item 3) is 13.5
  - Matrix technique predicts 13.48. Low error for this cell.
- Real answer for (User 13, Item 1) is 17.
  - Matrix technique predicts 15.3. Error is a little higher for this cell.
- In general, the prediction error varies across the cells, but taking all missing cells as a whole, the method aims to make predictions with low average error

Commerical Recommender Systems

- Commercial recommender systems (Netflix, Amazon) use variations of matrix factorisation.
- In 2009, Netflix offered a prize of $USD 1,000,000 in a competition to see which algorithms were most effective for predicting user-movie ratings.
  - Anonymised training data released to public: 100 million ratings by 480k users of 17.8k movies
  - Won by “BellKor’s Pragmatic Chaos” team
- *A followup competition was cancelled due to privacy concerns* … [We will elaborate when we get to topic on privacy]

Other issues

- Many challenging issues in deployment of recommendations
  - Interpretability of recommendations?
  - How to be fair to rare items?
  - How to avoid only recommending popular items?
  - How to handle new users?

References

- See
  - Matrix Factorization Techniques for Recommender Systems. Koren, Bell and Volinsky. IEEE Xplore, Vol 42, 2009. Available on the LMS in Week 3 section.

- Some slides based on “Data Mining Concepts and Techniques”, Han et al, 2^{nd} edition 2006.