Social Networks and Social Media

香港科技大學
HONG KONG
UNIVERSITY OF
SCIENCE &
TECHNOLOGY

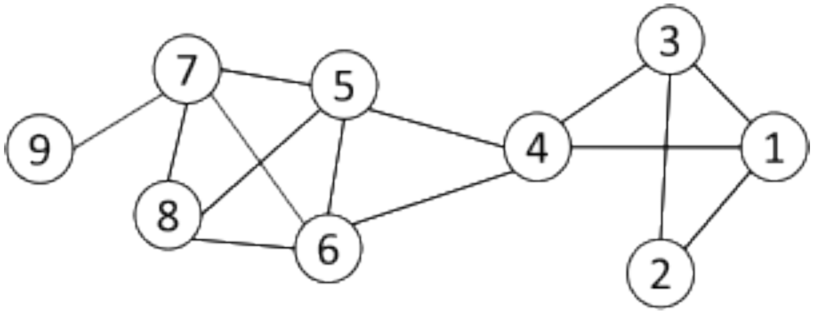
Why do statistics

- To understand the networks
  - Understand their topology and measure their properties
  - Study their evolution and dynamics
  - Create realistic models
  - Create algorithms that make use of the network structure

Networks and Representation

Social Network: A social structure made of nodes (individuals or organizations) and edges that connect nodes in various relationships like friendship, kinship etc.

- Graph Representation
- Matrix Representation



Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

Basic Concepts

- A: the adjacency matrix
- V: the set of nodes
- E: the set of edges
- v_i: a node v_i
- $e(v_i, v_j)$: an edge between node v_i and v_j
- N_i: the neighborhood of node v_i
- d_i: the degree of node v_i
- geodesic: a shortest path between two nodes
  – geodesic distance

Statistical Properties

- Static analysis
  - Static snapshots of graphs
- Dynamic analysis
  - A series of snapshots of graphs

Statistical Properties

- Static analysis
  - Static snapshots of graphs
- Dynamic analysis
  - A series of snapshots of graphs

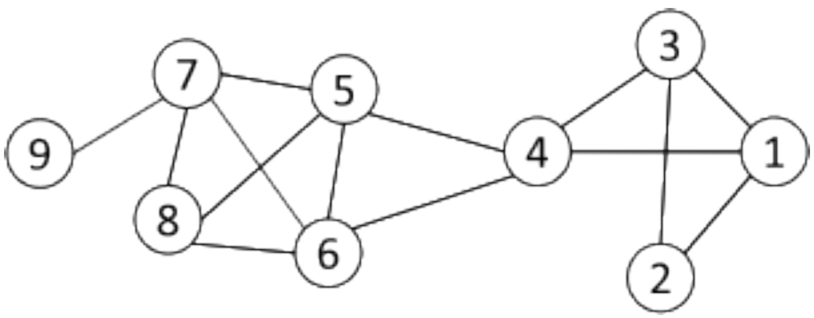
Why do statistics

- To understand the networks
  - Understand their topology and measure their properties
  - Study their evolution and dynamics
  - Create realistic models
  - Create algorithms that make use of the network structure

Networks and Representation

Social Network: A social structure made of nodes (individuals or organizations) and edges that connect nodes in various relationships like friendship, kinship etc.

- Graph Representation
- Matrix Representation



Node	1	2	3	4	5	6	7	8	9
1	-	1	1	1	0	0	0	0	0
2	1	-	1	0	0	0	0	0	0
3	1	1	-	1	0	0	0	0	0
4	1	0	1	-	1	1	0	0	0
5	0	0	0	1	-	1	1	1	0
6	0	0	0	1	1	-	1	1	0
7	0	0	0	0	1	1	-	1	1
8	0	0	0	0	1	1	1	-	0
9	0	0	0	0	0	0	1	0	-

Quality Estimation

- How to measure the discovering results?
  - Normalized Cut $$\frac{\sum_{i\in S, j\in \bar{S}} A(i,j)}{\sum_{i\in S} \deg ree(i)} + \frac{\sum_{i\in S, j\in \bar{S}} A(i,j)}{\sum_{i\in \bar{S}} \deg ree(i)}$$
  - Conductance $$\frac{\sum_{i\in S, j\in \bar{S}} A(i,j)}{\min(\sum_{i\in S} \deg ree(i), \sum_{i\in \bar{S}} \deg ree(i))}$$
  - Kernighan-Lin (KL) objective

$$\sum_{i\neq j} A(V_i, V_j) \ with \ |V_1| \equiv |V_2| \equiv \ldots \equiv |V_k|$$

Quality Estimation: Modularity

Q(division) = #(internal edges) - E(#(internal edges) in a RANDOM graph with same node degrees)

Trivial division: all vertices in one group ==> Q(trivial division) = 0

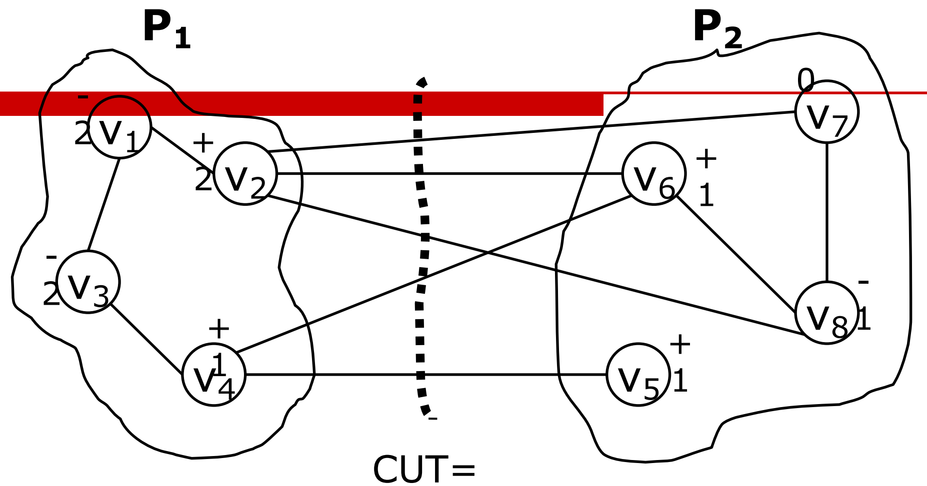
$$Q = \frac{1}{2m} \sum_{\ell=1}^{k} \sum_{i \in C_\ell, j \in C_\ell} \left(A_{ij} - d_i d_j / 2m \right)$$

m is the number of edges in the network

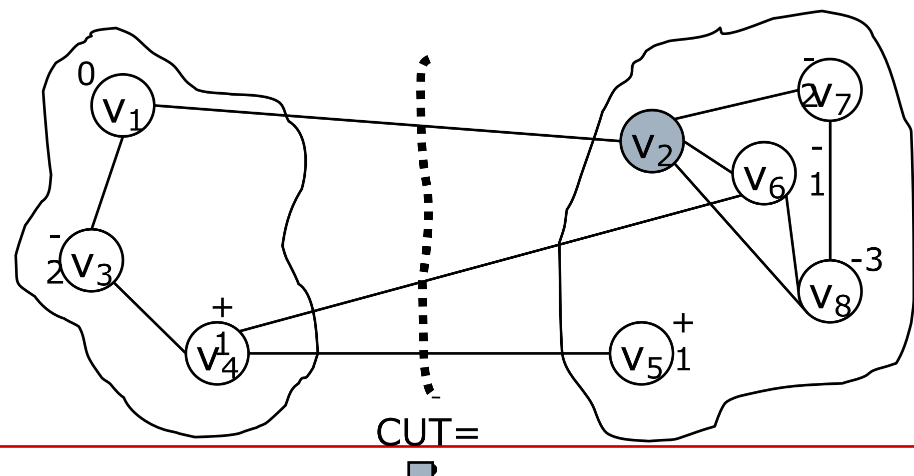
...

<u>*Optimizing any of these objectives is NP-Hard*</u>

The Kernighan-Lin (KL) algorithm

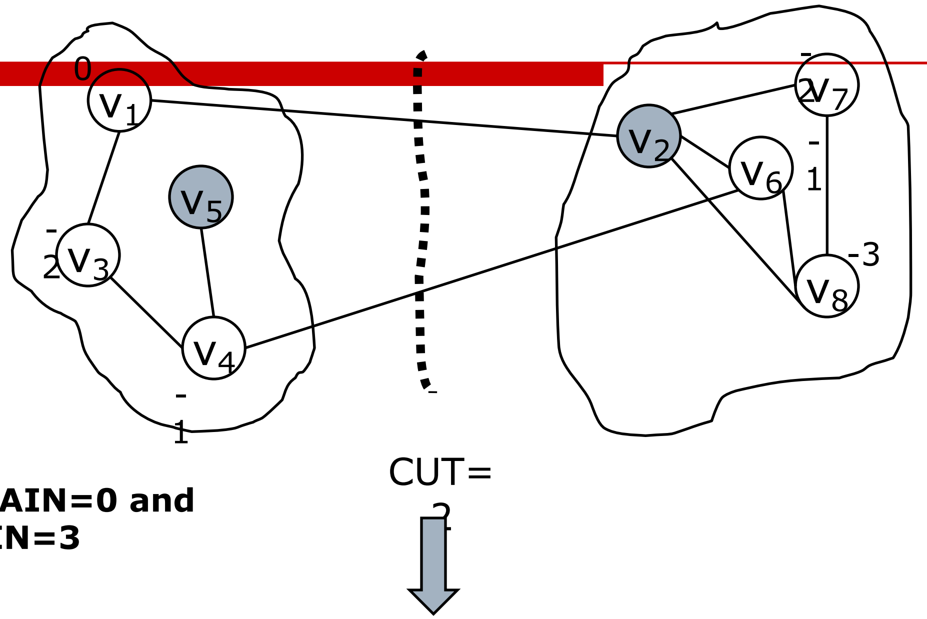


IF V_2 MOVES GAIN=2 and TOT_GAIN=2



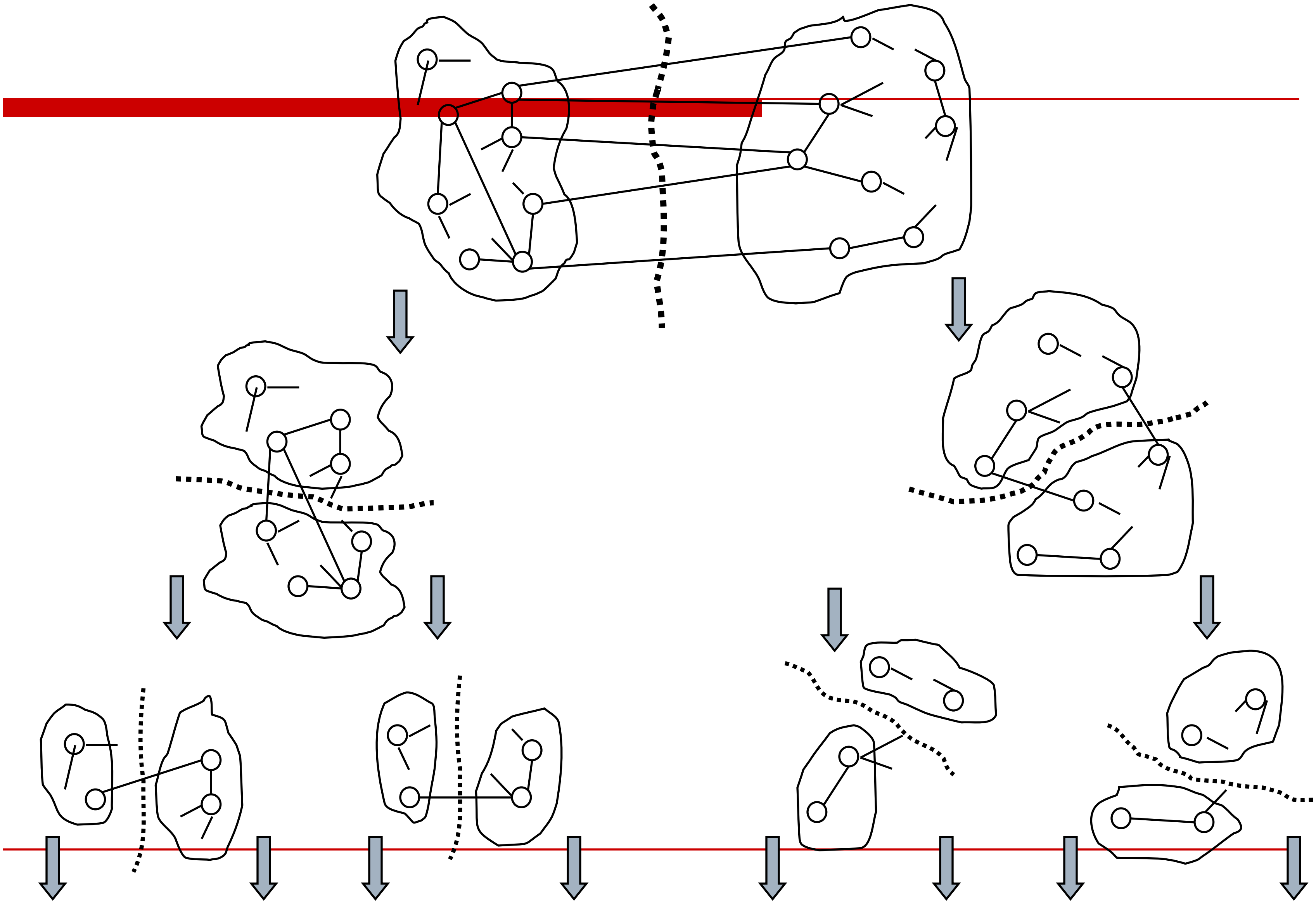
IF V_5 MOVES GAIN=1 and TOT_GAIN=3

The Kernighan-Lin (KL) algorithm con’t



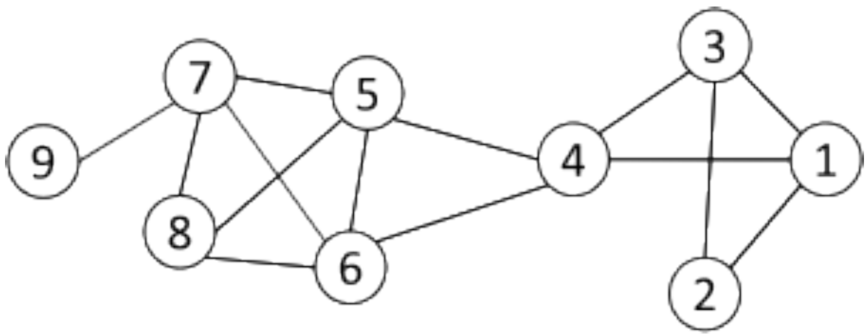
IF V1 MOVES GAIN=0 and TOT_GAIN=3

The Kernighan-Lin (KL) algorithm con't



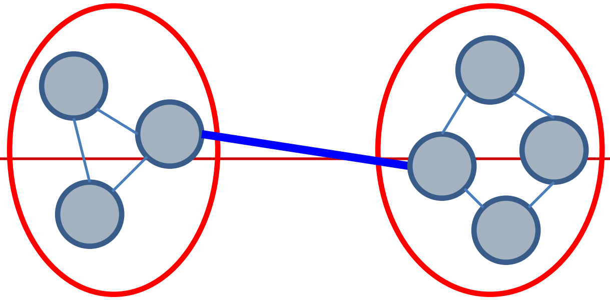
Edge Betweenness Method

- The strength of a tie can be measured by *edge betweenness*
- *Edge betweenness*: the number of shortest paths that pass along with the edge



The ***edge betweenness*** of e(1, 2) is 4 (=6/2 + 1), as all the shortest paths from 2 to {4, 5, 6, 7, 8, 9} have to either pass e(1, 2) or e(2, 3), and e(1,2) is the shortest path between 1 and 2

- The edge with higher betweenness tends to be the bridge between two communities.



Edge Betweenness Method

- Basic idea
  1. Calculate betweenness score for all edges
  2. Find the edge with the highest score and remove it from the network
  3. Recalculate betweenness for all remaining edges
  4. Repeat from step 2

Edge Betweenness Method con’t

Initial betweenness value

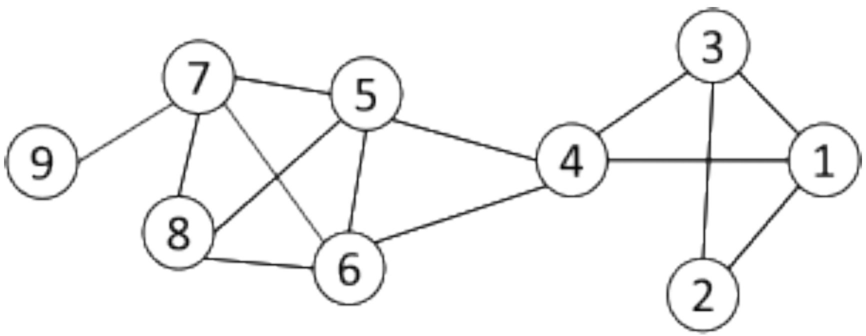
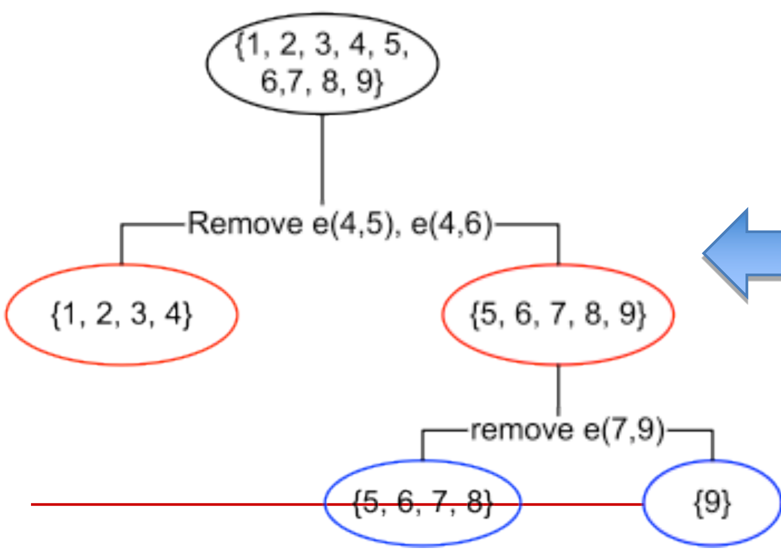


Table 3.3: Edge Betweenness

	1	2	3	4	5	6	7	8	9
1	0	4	1	9	0	0	0	0	0
2	4	0	4	0	0	0	0	0	0
3	1	4	0	9	0	0	0	0	0
4	9	0	9	0	10	10	0	0	0
5	0	0	0	10	0	1	6	3	0
6	0	0	0	10	1	0	6	3	0
7	0	0	0	0	6	6	0	2	8
8	0	0	0	0	3	3	2	0	0
9	0	0	0	0	0	0	8	0	0



After remove e(4,5), the betweenness of e(4, 6) becomes 20, which is the largest;

After remove e(4,6), the edge e(7,9) has the largest betweenness value 4, and should be removed.

Idea: progressively removing edges with the highest betweenness

Modularity Matrix

Q = ∑(Aij - ki*kj/M | i,j in the same group)

- Modularity matrix:

$$B = A - \mathbf{d}\mathbf{d}^T/2m \qquad (B_{ij} = A_{ij} - d_i d_j/2m)$$

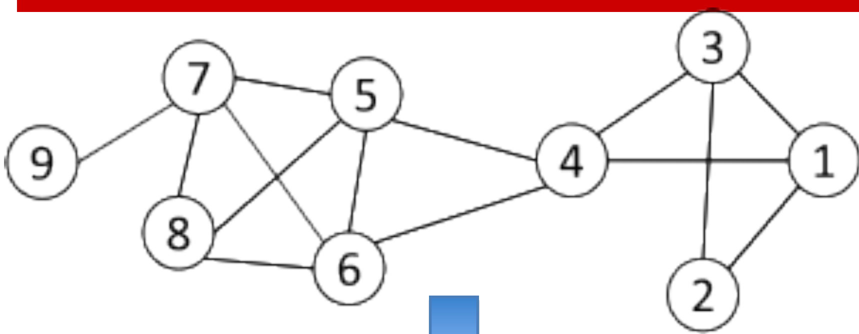
- Modularity maximization can be reformulated as

$$\max Q = \frac{1}{2m} Tr(S^T B S) \qquad s.t.\ S^T S = I_k$$

- Optimal solution: top eigenvectors of the modularity matrix
- Apply k-means to S as a post-processing step to obtain community partition

$$Q = \frac{1}{2m} \sum_{\ell=1}^{k} \sum_{i \in C_\ell, j \in C_\ell} (A_{ij} - d_i d_j/2m)$$

Modularity Maximization Example



Two Communities:

{1, 2, 3, 4} and {5, 6, 7, 8, 9}

k-means

$$B = \begin{bmatrix} -0.32 & 0.79 & 0.68 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\ 0.79 & -0.14 & 0.79 & -0.29 & -0.29 & -0.29 & -0.29 & -0.21 & -0.07 \\ 0.68 & 0.79 & -0.32 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\ 0.57 & -0.29 & 0.57 & -0.57 & 0.43 & 0.43 & -0.57 & -0.43 & -0.14 \\ -0.43 & -0.29 & -0.43 & 0.43 & -0.57 & 0.43 & 0.43 & 0.57 & -0.14 \\ -0.43 & -0.29 & -0.43 & 0.43 & 0.43 & -0.57 & 0.43 & 0.57 & -0.14 \\ -0.43 & -0.29 & -0.43 & -0.57 & 0.43 & 0.43 & -0.57 & 0.57 & 0.86 \\ -0.32 & -0.21 & -0.32 & -0.43 & 0.57 & 0.57 & 0.57 & -0.32 & -0.11 \\ -0.11 & -0.07 & -0.11 & -0.14 & -0.14 & -0.14 & 0.86 & -0.11 & -0.04 \end{bmatrix}$$

0.4384	-0.2709
0.3809	0.2671
0.4384	-0.2709
0.1716	0.6063
-0.2861	-0.3487
-0.2861	-0.3487
-0.3754	0.3355
-0.3421	0.1855
-0.1396	-0.1552

Modularity Matrix

Spectral Clustering

- Both ratio cut and normalized cut can be reformulated as

$$\min_{S \in \{0,1\}^{n \times k}} Tr(S^T \widetilde{L} S)$$

- Where $\widetilde{L} = \begin{cases} D - A & \text{graph Laplacian for ratio cut} \\ I - D^{-1/2} A D^{-1/2} & \text{normalized graph Laplacian} \end{cases}$

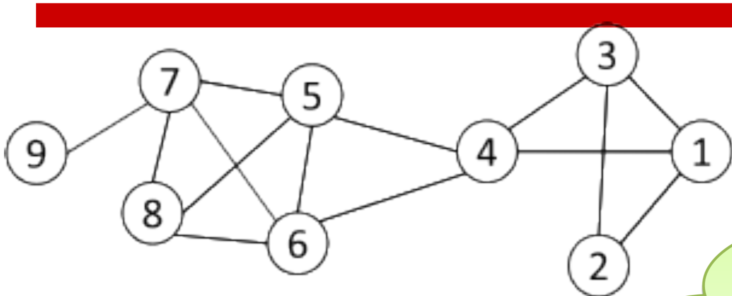
$$D = diag(d_1, d_2, \cdots, d_n)$$ A diagonal matrix of degrees

- Spectral relaxation:

$$\min_S Tr(S^T \widetilde{L} S) \quad s.t.\ S^T S = I_k$$

- Optimal solution: top eigenvectors with the smallest eigenvalues

Spectral Clustering Example



Two communities:

{1, 2, 3, 4} and {5, 6, 7, 8, 9}

The 1st eigenvector means all nodes belong to the same cluster, no use

k-means

$$D = diag(3, 2, 3, 4, 4, 4, 4, 3, 1)$$

$$\widetilde{L} = D - A = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} 0.33 & -0.38 \\ 0.33 & -0.48 \\ 0.33 & -0.38 \\ 0.33 & -0.12 \\ 0.33 & 0.16 \\ 0.33 & 0.16 \\ 0.33 & 0.30 \\ 0.33 & 0.24 \\ 0.33 & 0.51 \end{bmatrix}$$

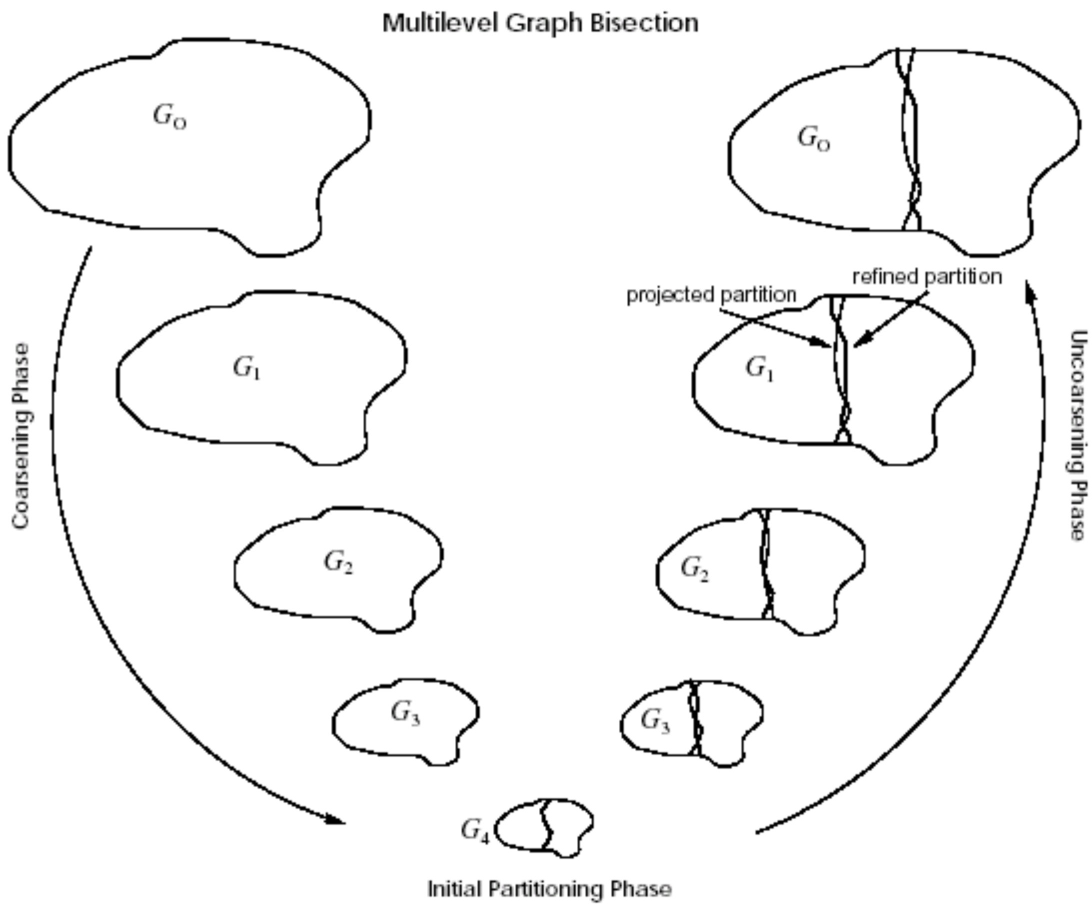
Centered matrix

Multi-level Graph Partitioning

- Logic flow
  1. Produce a smaller graph that is similar to the original graph
  2. A partitioning of the coarsest graph is performed.
  3. the partitioning of the coarser graph is projected back to the original graph. The partition is further refined.

Multi-level Graph Partitioning

- *3 Phases*
  - *Coarsen*
  - *Partition*
  - *Uncoarse*



Other works

- Community Discovery in Dynamic Networks
  - How should community discovery algorithms be modified to dynamic networks?
  - How do communities get formed?
  - How persistent and stable are communities and their members?
  - How do they evolve over time?
- Community discovery in Heterogeneous Networks
- Coupling Content Relationship Information for Community Discovery

Link Prediction in Social Networks

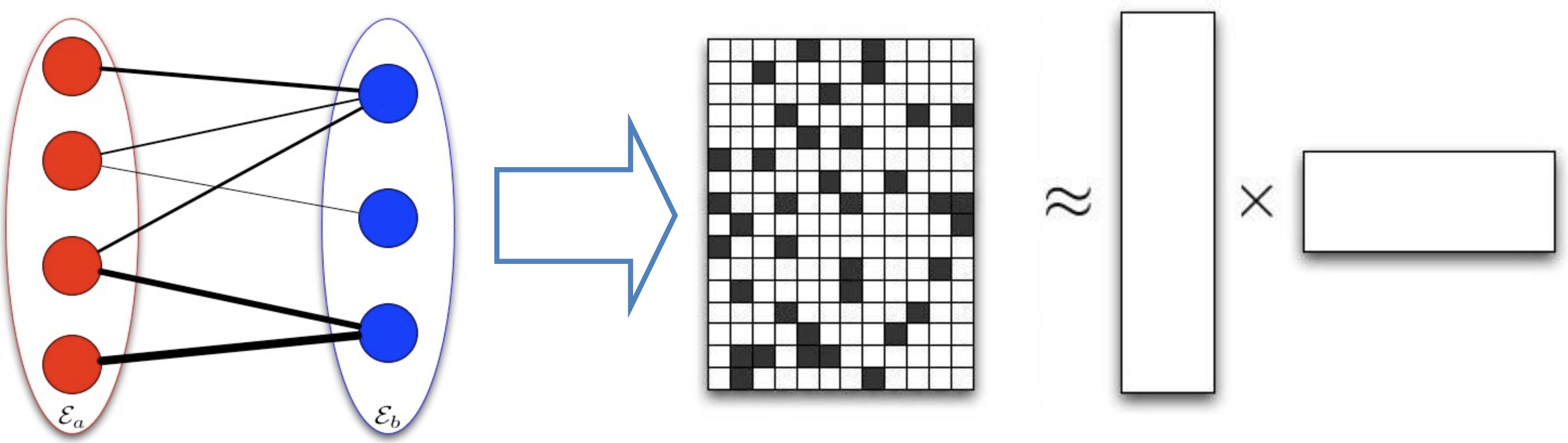
香港科技大學
HONG KONG
UNIVERSITY OF
SCIENCE &
TECHNOLOGY

Outline

- Link Prediction Problems
  - Social Network
  - Recommender system
- Algorithms of Link Prediction
  - Supervised Methods
  - Collaborative Filtering
- Recommender System and The Netflixprize
- References

Link Prediction using Collaborative Filtering

- Find the background model that can generate the link data



Challenges in Link Prediction

- Data!!!
- Cold Start Problem
- Sparsity Problem

Link Prediction using Collaborative Filtering

- Memory-based Approach
  - User-base approach [Twitter]
  - item-base approach [Amazon & Youtube]
- Model-based Approach
  - Latent Factor Model [Google News]
- Hybrid Approach

Memory-based Approach

- Few modeling assumptions
- Few tuning parameters to learn
- Easy to explain to users
  - Dear Amazon.com Customer, We've noticed that customers who have purchased or rated *How Does the Show Go On: An Introduction to the Theater* by Thomas Schumacher have also purchased *Princess Protection Program #1: A Royal Makeover (Disney Early Readers)*.

Algorithms: User-Based Algorithms (Breese et al, UAI98)

- $v_{i,j}$= vote of user *i* on item *j*
- I_i = items for which user *i* has voted
- Mean vote for *i* is

$$\overline{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$$

- Predicted vote for "active user" *a* is weighted sum

$$p_{a,j} = \overline{v}_a + \kappa \sum_{i=1}^{n} \underbrace{w(a,i)}_{\text{weights of } n \text{ similar users}} (v_{i,j} - \overline{v}_i)$$

normalizer

Algorithms: User-Based Algorithms (Breese et al, UAI98)

- K-nearest neighbor

$$w(a,i) = \begin{cases} 1 & \text{if } i \in \text{neighbors}(a) \\ 0 & \text{else} \end{cases}$$

- Pearson correlation coefficient (Resnick ’94, Grouplens):

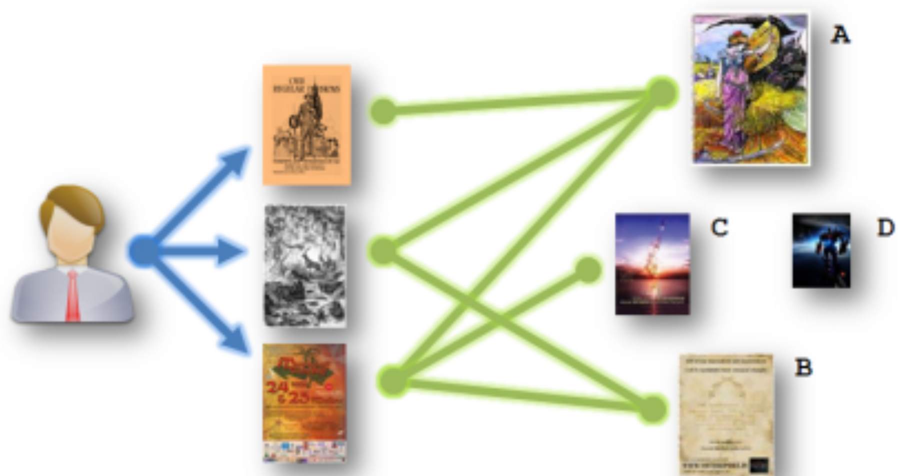
$$w(a,i) = \frac{\sum_j (v_{a,j} - \overline{v}_a)(v_{i,j} - \overline{v}_i)}{\sqrt{\sum_j (v_{a,j} - \overline{v}_a)^2 \sum_j (v_{i,j} - \overline{v}_i)^2}}$$

- Cosine distance (from IR)

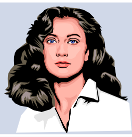
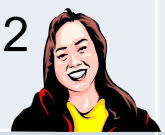
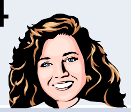
$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

Algorithm: Amazon's Method

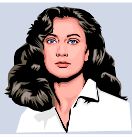
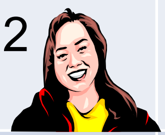
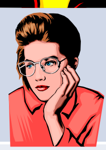
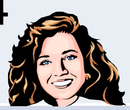
- Item-based Approach
  - Similar with user-based approach but is on the item side



Item-based CF Example: infer (user 1, item 3)

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	?
User 6 	8	3	8	3	7

How to Calculate Similarity (Item 3 and Item 5)?

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	?
User 6 	8	3	8	3	7

Similarity between Items

Item 3	Item 4	Item 5
?	2	7
5	7	5
7	4	7
7	3	8
4	6	?
8	3	7

- How similar are items 3 and 5?
  - How to calculate their similarity?

Similarity between items

Item 3	Item 5
?	7
5	5
7	7
7	8
4	?
8	7

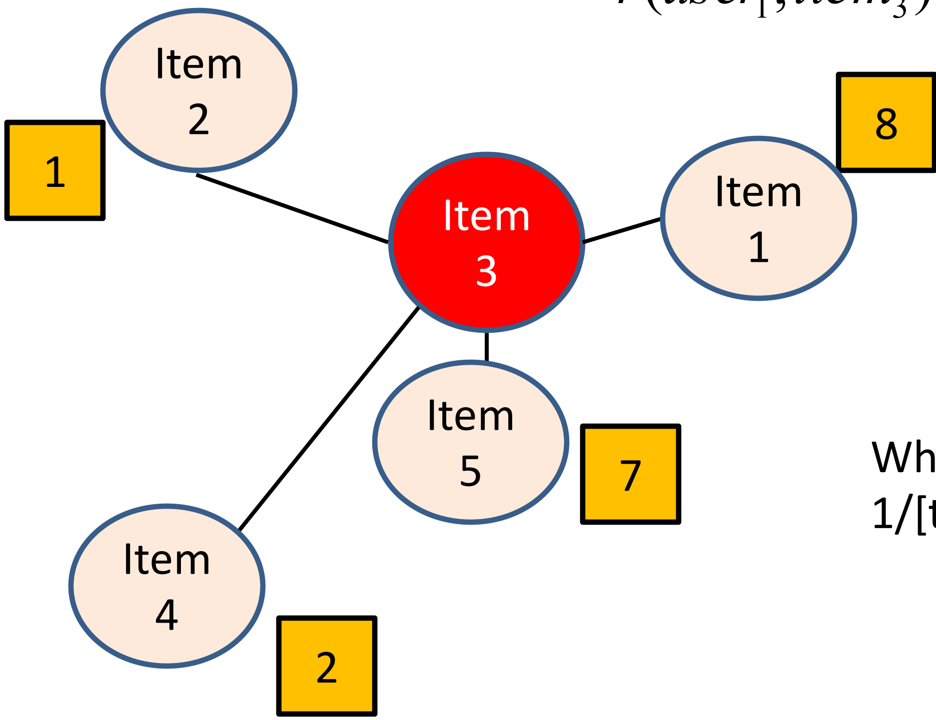
- Only consider users who have rated both items
- For each user:
  Calculate difference in ratings for the two items
- Take the average of this difference over the users

sim(item 3, item 5) = cosine((5, 7, 7), (5, 7, 8))

= (5*5 + 7*7 + 7*8)/(sqrt($5^2+7^2+7^2$)* sqrt($5^2+7^2+8^2$))

- Can also use Pearson Correlation Coefficients as in user-based approaches

Prediction: Calculating ranking r(user1,item3)



$$r(user_1,item_3)=\alpha*\{r(user_1,item_1)sim(item_1,item_3)$$
$$+r(user_1,item_2)sim(item_2,item_3)$$
$$+r(user_1,item_4)sim(item_4,item_3)$$
$$+r(user_1,item_5)sim(item_5,item_3)\}$$

Where α is a normalization factor, which is 1/[the sum of all sim($item_i$,$item_3$)].

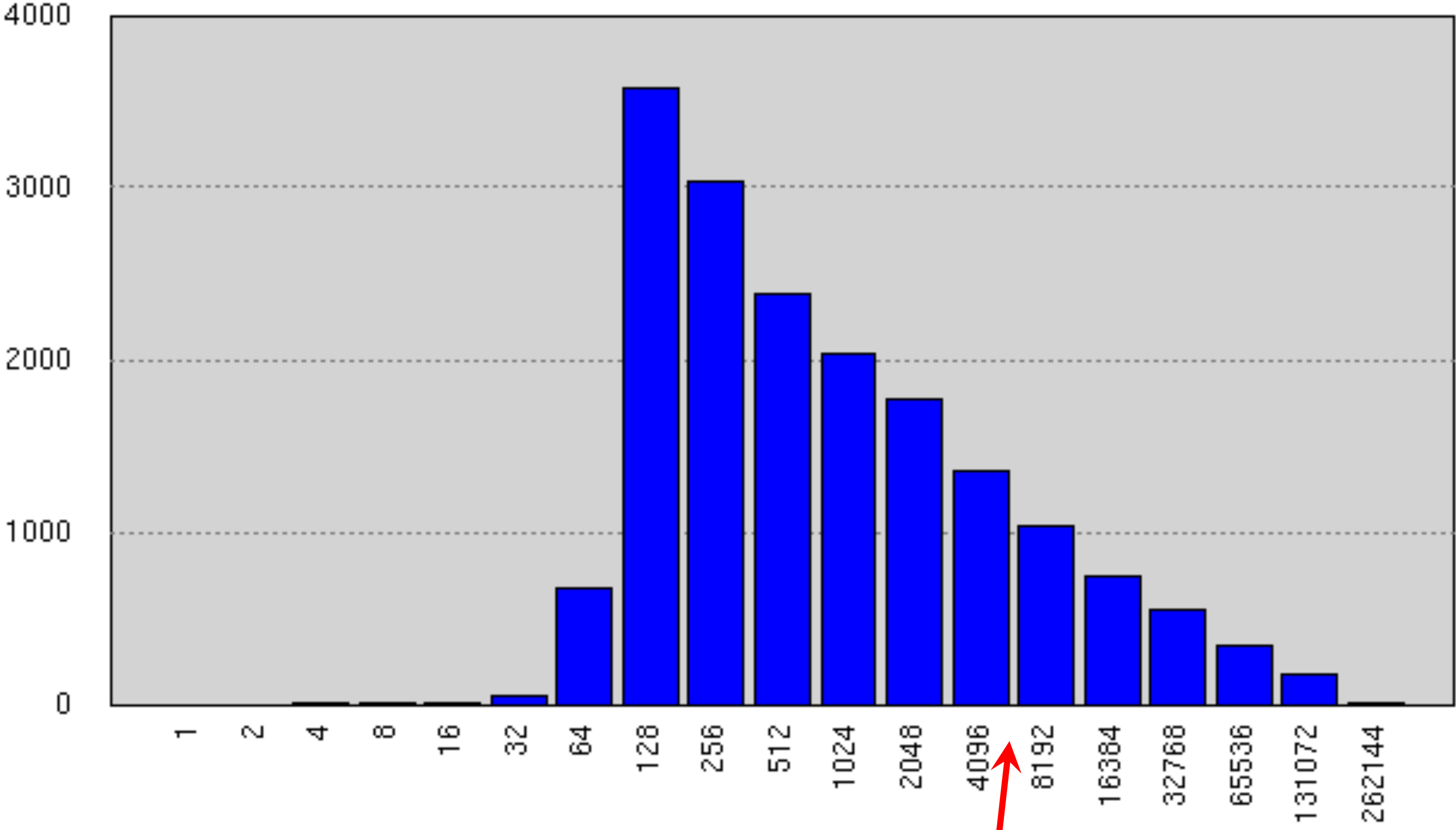
Major Challenges

1. Size of data
   - Places premium on efficient algorithms
   - Stretched memory limits of standard PCs
2. 99% of data are missing
   - Eliminates many standard prediction methods
   - Certainly *not* missing at random
3. Training and test data differ systematically
   - Test ratings are later
   - Test cases are spread uniformly across users

Major Challenges (cont.)

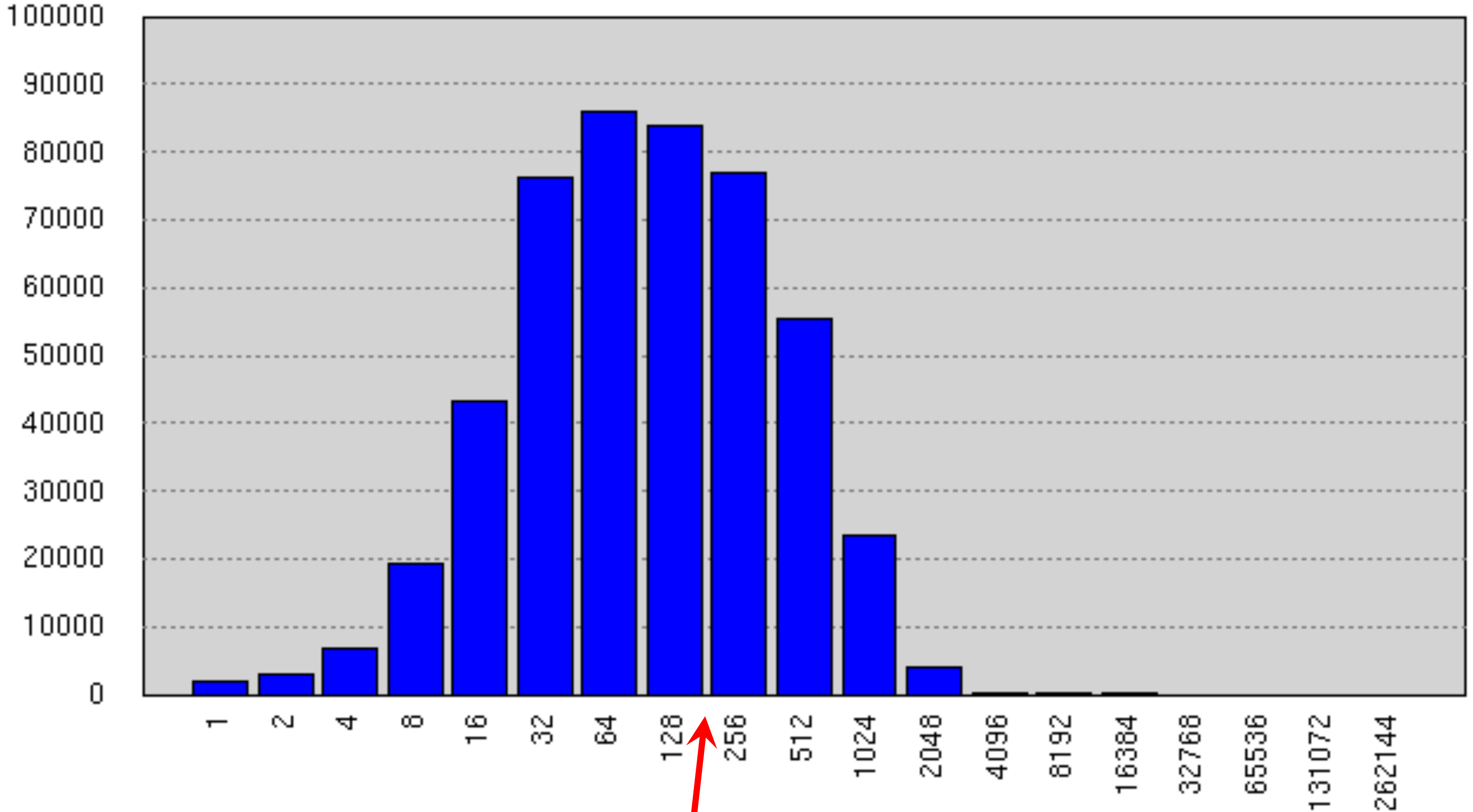
4. Countless factors may affect ratings
   - Genre, movie/TV series/other
   - Style of action, dialogue, plot, music et al.
   - Director, actors
   - Rater’s mood
5. Large imbalance in training data
   - Number of ratings per user or movie varies by several orders of magnitude
   - Information to estimate individual parameters varies widely

Ratings per Movie in Training Data



Avg #ratings/movie: 5627

Ratings per User in Training Data



Avg #ratings/user: 208