Unit 4 - Recommender Systems

Class notes: Vibha Masti

Feedback/corrections: vibha@pesu.pes.edu

Table of Contents

Unit 4 - Recommender Systems
- Table of Contents
- Goals of Rec System
- Types of Recommender Systems
  - Domain-Specific Challenges in RS
  - Cold Start Problem
  - Long Tail Phenomenon
- 1. Collaborative Filtering
  - Utility Function - Formal Model
  - Collaborative Filtering
  - 1.1 Memory-Based Methods/Neighborhood-based CF Algorithms
    - Similarity Measures
      - 1. Jaccard similarity
      - 2. Cosine similarity
      - 3. Centered cosine similarity
      - 4. Minkowski distance
      - 5. Mahalanobis distance
      - 6. Simple Matching coefficients for Binary Vectors
      - 7. Jaccard Matching for Binary Vectors
      - SMC vs Jaccard
      - 8. Extended Jaccard Coefficient (Tanimoto)
      - 9. Correlation coefficient
      - 10. Weighted similarity measures
      - 11. Density
    - 1.1.1 User-Based Collaborative Filtering
    - 1.1.2 Item-Based Collaborative Filtering
    - Item-Item vs User-User
      - Item-Item
      - User-User
      - Clustering
      - Eg: MovieLens Dataset
  - 1.2 Model-Based Methods
- 2. Knowledge-Based
  - User-Recommender Interactions
    - 1. Conversational systems
    - 2. Search-based systems
    - 3. Navigation-based systems
  - 2.1 Constraint-Based
  - 2.2 Case-Based

- Case-Based Reasoning
- (a) kNN - Instance-Based Learning (Lazy Learner)
  - Example problem: identify if a pattern is the work of Mondrian
- (b) Decision Trees (CART)
- Query Augmentation
- Tree-based case representation
  - CBR Containers
- 3. Other Methods
  - Ensemble Methods - Bagging and Boosting
    - (a) Bagging
    - (b) Boosting
  - SVM
  - ANN
  - Clustering
    - (a) K-means clustering
    - (b) Agglomerative clustering
    - (c) DBSCAN
      - Reachability and Connectivity
    - Cluster Cohesion
    - Cluster Separation
- 4. Content-Based
  - TF-IDF
    - Text to Numbers
    - Vector Space Model
    - Example problem
  - Text Classification
    - Feature Selection
    - Domains of Text Classification
    - Naive Bayes Classifier
    - Mixture Models
- Market Based Analysis (Frequent Itemset Mining)
  - Frequent Itemset
  - Apriori Principle
  - Association Rule Mining
  - Contingency Table for $X \to Y$
  - Rule Generation
    - Example
  - Handling of Categorical Attributes
  - Handling of Continuous Attributes
- Evaluation of Recommender Systems

Goals of Rec System

1. Prediction version of problem: predict the rating value for a user-item combination
2. Ranking version of problem: determination of the top-k

Types of Recommender Systems

1. Collaborative Filtering
    - Memory-based CF
        - User-based
        - Item-based
    - Model-based CF
        - Implicit/explicit ratings
        - Relationship with missing values
2. Knowledge-based
    - Constraint-based
    - Case-based
3. Content-based
4. Demographic
5. Hybrid and Ensemble

Domain-Specific Challenges in RS

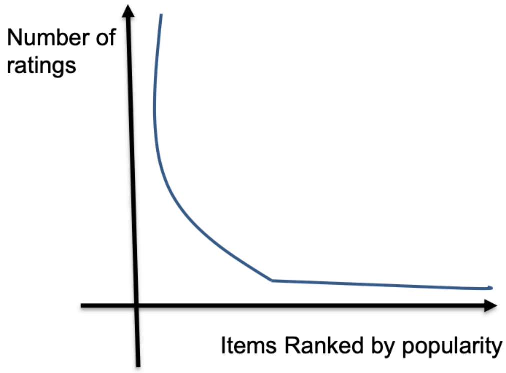
1. Context-based
    - Influenced by time, location, social data
    - Eg: clothing based on season and location
2. Time-sensitive
    - Evolve over time with community interests
    - Time of day, week, month, year, season
    - Eg: clothing based on season
3. Location-based
    - User-specific locality
    - Item-specific locality
4. Social
    - Structural rec of nodes and links
    - Product and content
    - Trustworthy
    - Leveraging Social Tagging Feedback

Cold Start Problem

- New items have very few ratings
- New users have no history

Long Tail Phenomenon

- Most products have low frequency of ratings
- Small fraction of products have high ratings



1. Collaborative Filtering

Utility Function - Formal Model

- Maps every pair of (customer, item)
- $U : C \times S \to R$
    - C: set of customers
    - S: set of items
    - R: set of ratings
- Utility matrix

	Avatar	KGF	Matrix	Bahubali
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

Collaborative Filtering

- Use the collaborative power of the ratings by multiple users to make recommendations
- Underlying ratings matrices are sparse
- Impute these ratings
- Observed ratings are highly correlated across various users and items
- Similar to missing values analysis

1.1 Memory-Based Methods/Neighborhood-based CF Algorithms

- Ratings of user-item combinations are predicted on the basis of their neighborhoods
- Memory-based techniques are easy to implement
- One of two ways:

1. **Prediction version of problem**
    - Predicting the rating value of a user-item combination
    - Missing rating r_{uj} value for user u and item j
2. **Ranking version of problem**
    - Determining the top-k items or top-k users
    - More common to find top k items
    - Items typically have less no of clusters

Similarity Measures

1. Jaccard similarity

- $sim(A,B) = \dfrac{|r_A \cap r_B|}{|r_A \cup r_B|}$

2. Cosine similarity

- $sim(A,B) = \dfrac{r_A \cdot r_B}{|r_A|\,|r_B|} = \cos(\theta_{AB})$

3. Centered cosine similarity

- mean-centered

4. Minkowski distance

- $dist(A,B) = \left(\sum_{k=1}^{n} |A_k - B_k|^r\right)^{\dfrac{1}{r}}$
    - n: number of dimensions
    - For $r = 1$: Manhattan distance, Hamming distance (binary vectors - number of differing bits), L_1 norm distance
    - For $r = 2$: Euclidean distance, L_2 norm distance
    - For $r = \infty$: Supremum distance, $L_{\max}$ norm distance, L_∞ norm distance
- Eg:

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L_1 **norm**	p_1	p_2	p_3	p_4
p_1	0	4	4	6
p_2	4	0	2	4
p_3	4	2	0	2
p_4	6	4	2	0

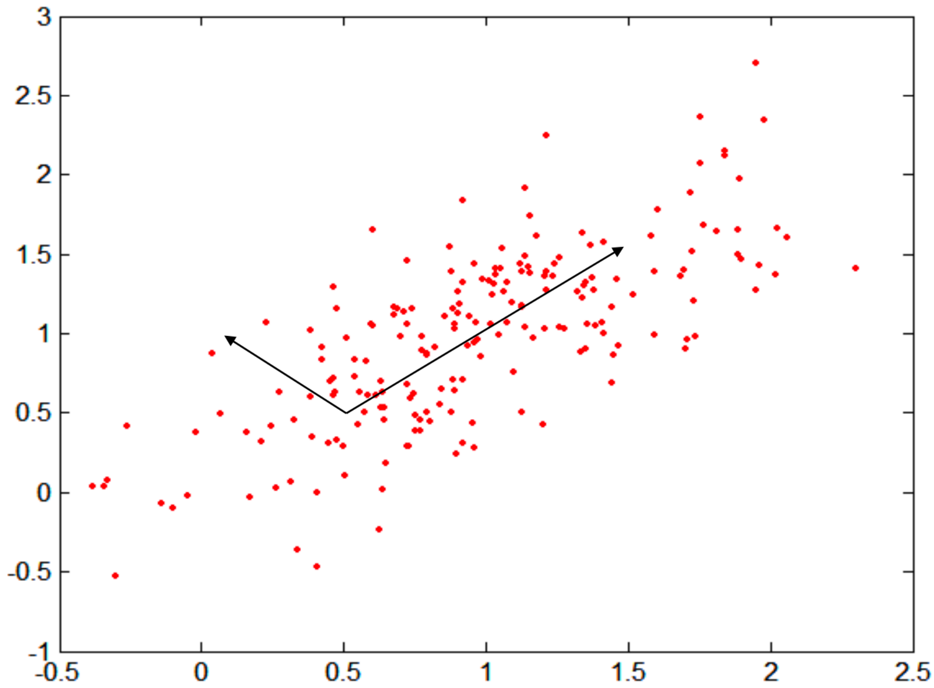
L_2 **norm**	p_1	p_2	p_3	p_4
p_1	0	2.828	3.162	5.099
p_2	2.828	0	1.414	3.162
p_3	3.162	1.414	0	2
p_4	5.099	3.162	2	0

- L_∞ norm: Maximum difference between any component of the vectors

L_∞ **norm**	p_1	p_2	p_3	p_4
p_1	0	2	3	5
p_2	2	0	1	3
p_3	3	1	0	2
p_4	5	3	2	0

5. Mahalanobis distance

- $dist(A, B) = \sqrt{(A - B)^T \, \Sigma^{-1}(A - B)}$



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

Mahal(A,C) = 4

- $A - B = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$
- $\begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}^{-1} = 20 \times \begin{bmatrix} 0.3 & -0.2 \\ -0.2 & 0.3 \end{bmatrix} = \begin{bmatrix} 6 & -4 \\ -4 & 6 \end{bmatrix}$
- $dist(A, B) = \sqrt{\begin{bmatrix} 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} 6 & -4 \\ -4 & 6 \end{bmatrix} \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}}$
- $dist(A, B) = \sqrt{5}$

6. Simple Matching coefficients for Binary Vectors

- $sim(A, B) = \frac{\text{number of matches}}{\text{number of attributes}}$
- $sim(A, B) = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$

7. Jaccard Matching for Binary Vectors

- $sim(A, B) = \frac{\text{number of 11 matches}}{\text{number of non-0 attributes}}$
- $sim(A, B) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$

SMC vs Jaccard

- Eg:

x = 1 0 0 0 0 0 0 0 0 0

y = 0 0 0 0 0 0 1 0 0 1

f_{01} = 2 (the number of attributes where **x** was 0 and **y** was 1)

f_{10} = 1 (the number of attributes where **x** was 1 and **y** was 0)

f_{00} = 7 (the number of attributes where **x** was 0 and **y** was 0)

f_{11} = 0 (the number of attributes where **x** was 1 and **y** was 1)

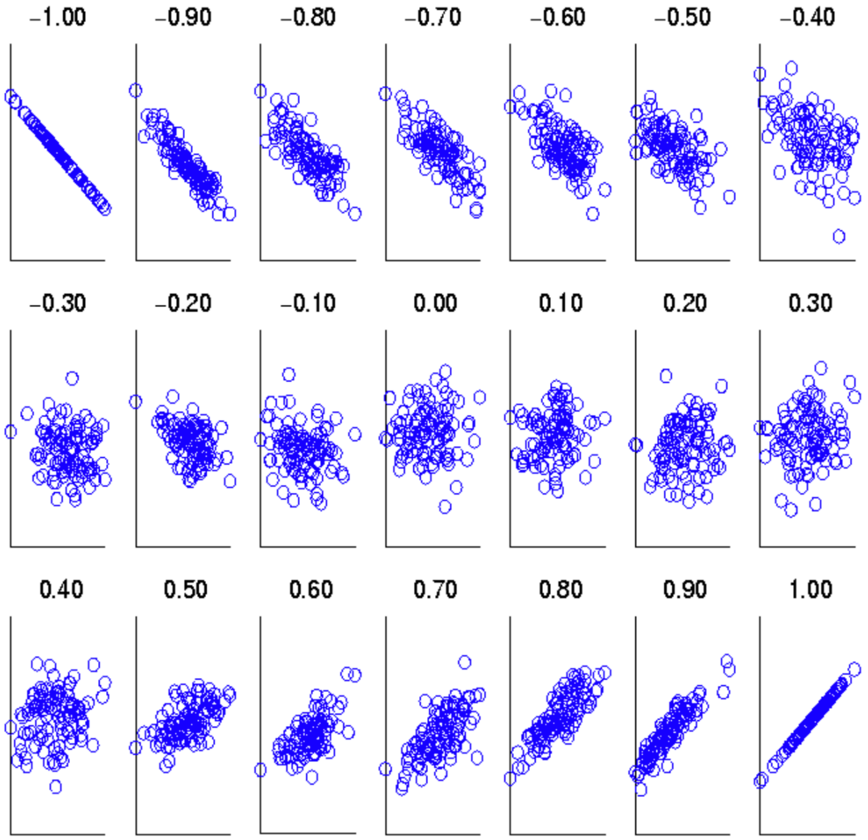
- $\text{SMC} = \frac{7}{10} = 0.7$
- $\text{Jaccard} = \frac{0}{10} = 0$

8. Extended Jaccard Coefficient (Tanimoto)

- $sim(A, B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B}$
- For continuous or count attributes
- Reduces to Jaccard for binary attributes

9. Correlation coefficient

- $sim(A, B) = \frac{\text{covariance}(A, B)}{\text{Standard deviation}(A) \times \text{Standard deviation}(B)}$
- $sim(A, B) = \frac{S_{AB}}{S_A \times S_B}$

- 

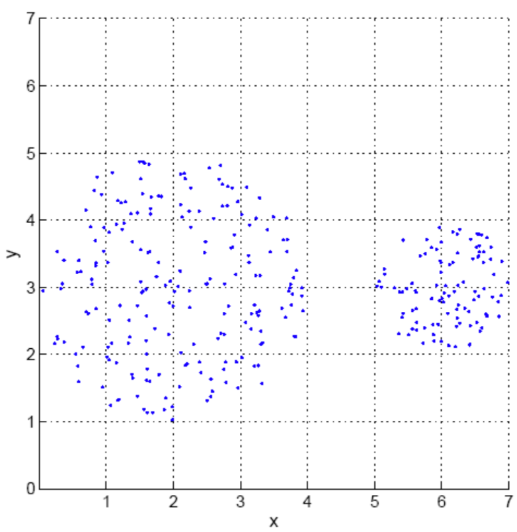
Scatter plots showing the similarity from –1 to 1.

10. Weighted similarity measures

- Use non-negative weights

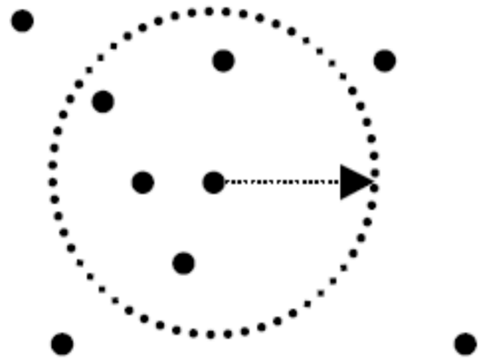
11. Density

- Euclidean density = number of points per unit volume
- **Grid-based Approach**
    - Divide region into a number of rectangular cells of equal volume
    - Number of points per cell



0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

- **Centre-based Approach/Euclidean Density**
    - Number of points within a specified radius of the point



1.1.1 User-Based Collaborative Filtering

- Ratings provided by the **like-minded users** of a target user A are used in order to make the recommendations for A
- Similarity matrix for users

- Eg: Users A, B, C, D and movies HP1, HP2, KGF, BB1, BB2, BB3

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- **Jaccard similarity**: $sim(A, B) = \frac{|r_A \cap r_B|}{|r_A \cup r_B|}$
    - $sim(A, B) = \frac{count(HP1)}{count(HP1, KGF, BB1, HP2, HP3)}$
    - $sim(A, B) = \frac{1}{5} = 0.2$
    - $sim(A, C) = \frac{count(KGF, BB1)}{count(HP1, KGF, BB1, BB2)}$
    - $sim(A, C) = \frac{2}{4} = 0.5$
    - Using Jaccard, $sim(A, B) < sim(A, C)$
    - **Flaw**: ignores rating values

- **Cosine similarity**: $sim(A, B) = \frac{r_A \cdot r_B}{|r_A|\,|r_B|} = \cos(\theta_{AB})$
  - $sim(A, B) = \frac{4 \times 5 + 0}{\sqrt{4^2 + 5^2 + 1^2}\sqrt{5^2 + 5^2 + 4^2}}$
  - $sim(A, B) = \frac{20}{\sqrt{42}\sqrt{66}} = 0.3799$
  - $sim(A, C) = \frac{5 \times 2 + 1 \times 4 + 0}{\sqrt{4^2 + 5^2 + 1^2}\sqrt{2^2 + 4^2 + 5^2}}$
  - $sim(A, C) = \frac{14}{\sqrt{42}\sqrt{45}} = 0.3220$
  - Using cosine, $sim(A, B) > sim(A, C)$ (only slightly)
  - **Flaw**: ignores missing values

- **Centered cosine similarity**: normalise rows by subtracting row mean
  - Missing ratings treated as average
  - **Pearson correlation**
  - $sim(A, B) = \frac{\frac{2}{9}}{\sqrt{\frac{26}{3}}\sqrt{\frac{2}{3}}}$
  - $sim(A, B) = 0.0925$
  - $sim(A, C) = \frac{-\frac{32}{9}}{\sqrt{\frac{26}{3}}\sqrt{\frac{14}{3}}}$
  - $sim(A, C) = -0.5591$
  - Using centered cosine, $sim(A, B) > sim(A, C)$

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4-10/3=2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

1.1.2 Item-Based Collaborative Filtering

- Determine a **set S of items** that are most similar to target item B by user A
- Similar items are identified to a target item
- User’s own ratings on those similar items are used to extrapolate the ratings of the target
- Item-based methods provide more relevant recommendations
- Estimate rating of item i based on similar items
  - $r_{xi} = \frac{\sum_{j \in N(i;x)} S_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} S_{ij}}$
  - r_{xi} : rating of user x on item i
  - r_{xj} : rating of user x on item j
  - S_{ij} : similarity of item i and item j
  - $N(i, x)$: set of k nearest items rated by user x similar to item i

- Eg: Users 1 to 10, movies 1 to 6

Users

Movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Unknown Rating

Rating between 1to 5

- Estimate rating of movie 1 by user 5

- **Pearson correlation similarity**
  - Subtract mean

Movies \ Users	1	2	3	4	5	6	7	8	9	10	11	12	Sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

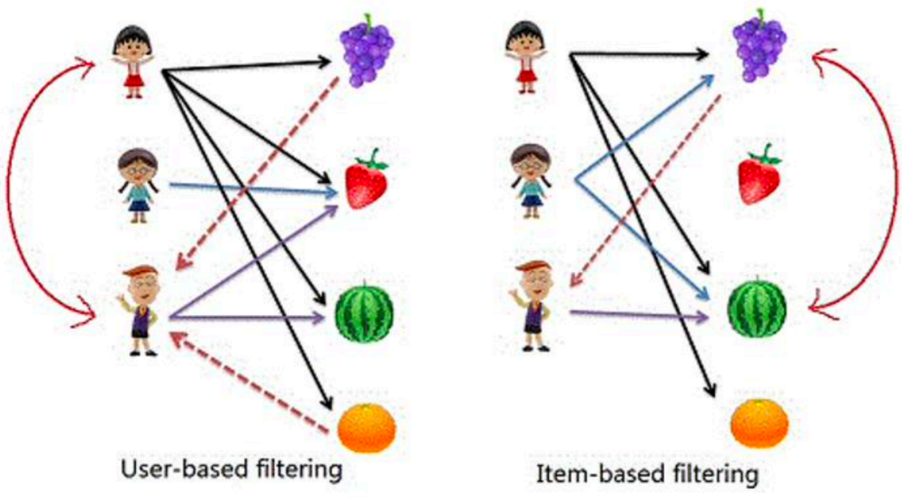
- Compute $sim(1, m)$ for $m = 1$ to $m = 6$ for normalised values of movie ratings (compute similarities of movies, not users)
- Taking $k = 2$ nearest neighbours for user 5 we get movie 6 with $sim(1, 6) = 0.59$ and movie 3 with $sim(1, 3) = 0.41$

Movies \ Users	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- Computed weighted average of ratings of k nearest neighbours to find the rating of movie 1 with user 5
- $r_{15} = \dfrac{sim(1,6) \times r_{65} + sim(1,3) \times r_{35}}{sim(1,6) + sim(1,3)}$
- $r_{15} = \dfrac{0.59 \times 3 + 0.41 \times 2}{0.59 + 0.41} = 2.59$

Item-Item vs User-User

- Item-item outperforms user-user in many use cases
- Items belong to a small set of genres, users have varied tastes (more similar)



Item-Item

- Scalability and performance are achieved by creating the expensive similar-items table offline
- Scales independently of the number of customers
- Fast for large datasets
- Recommends highly correlated similar items
- Performs well with limited user data

User-User

- Minimal offline computation
- Impractical on large datasets
- Dim reduction reduces rec quality

Clustering

- Much of the computation offline
- Quality poor

Eg: MovieLens Dataset

	User Based	Model Based	Item Based
Model Construction Time (sec.)	730	254	170
Prediction Time (sec.)	31	1	3
MAE	0.6688	0.6736	0.6382

1.2 Model-Based Methods

- ML and data mining methods used
- Predictive models
- Eg: Decision trees, Rule-based models, Bayesian methods and latent factor models

2. Knowledge-Based

- Customers want to explicitly specify their requirements (interactivity)
- Difficult to obtain ratings for a specific type of item

User-Recommender Interactions

1. Conversational systems

- User preferences in feedback loop
- Iterative conversational system
- Critiquing recommender systems - case based

2. Search-based systems

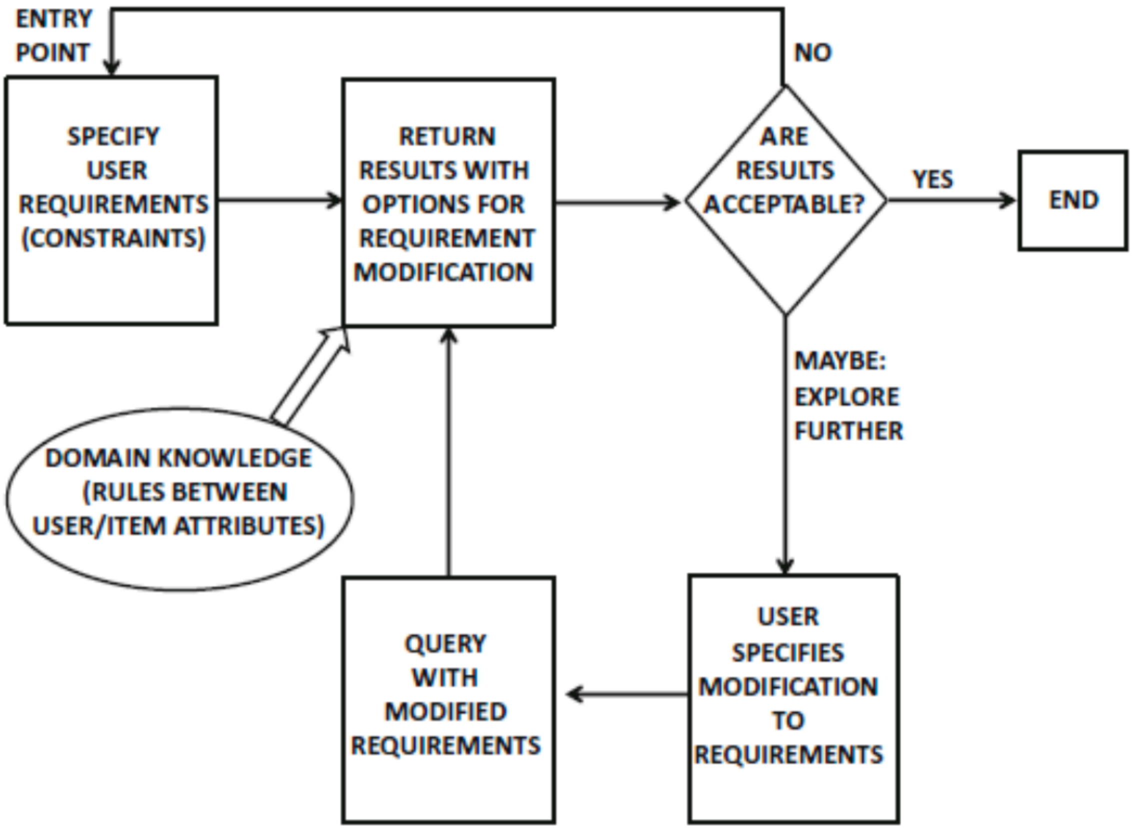
- User preferences from answers to questions
- Eg: "Do you prefer a house in a suburban area or within the city?"
- Can be for constraint based

3. Navigation-based systems

- User specifies a number of change requests to item being currently recommended
- Iterative set of change requests
- Eg: "I would like a similar house about 5 miles west of the currently recommended house"
- Critiquing recommender systems - case based

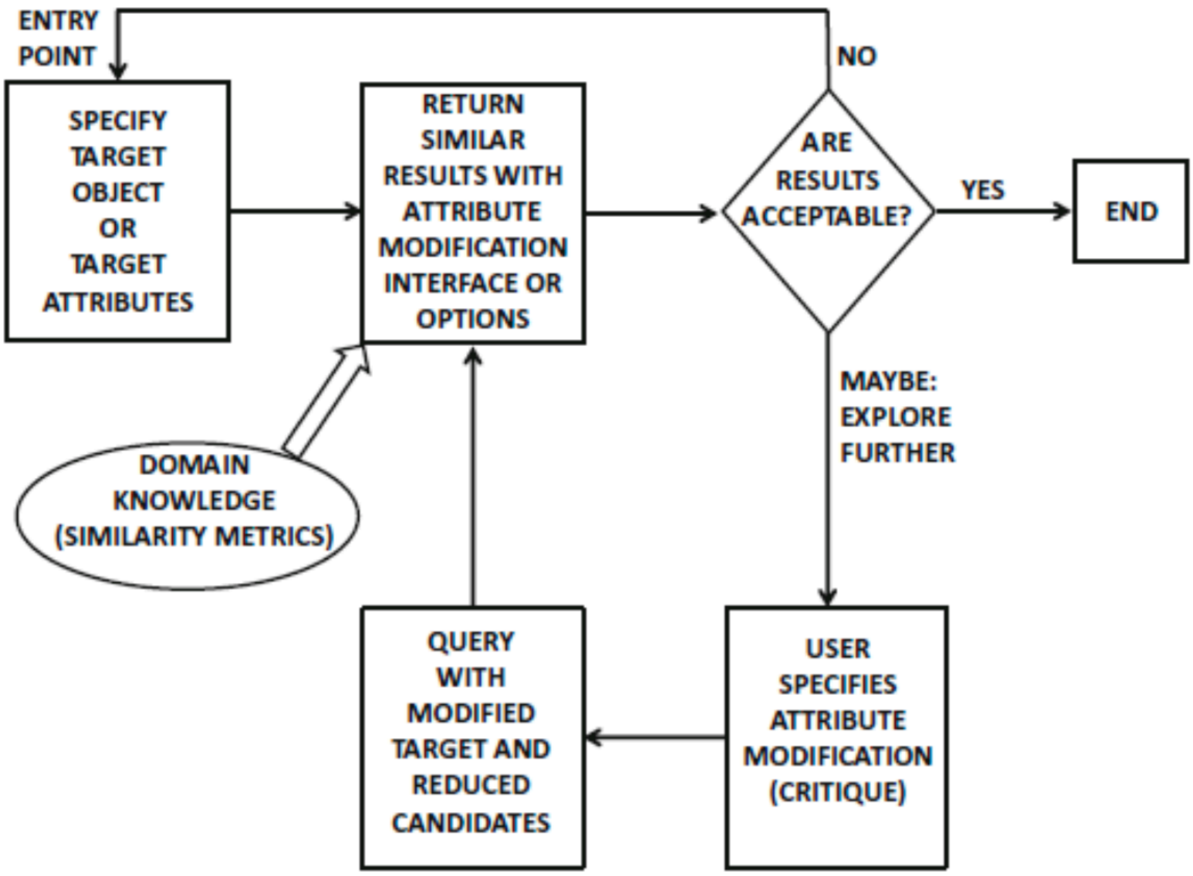
2.1 Constraint-Based

- Users specify requirements or constraints on item attributes
- Domain knowledge: mapping user requirements to item attributes
- Original query modified by addition, deletion, modification or relaxation of original requirements
- Complex problem domain



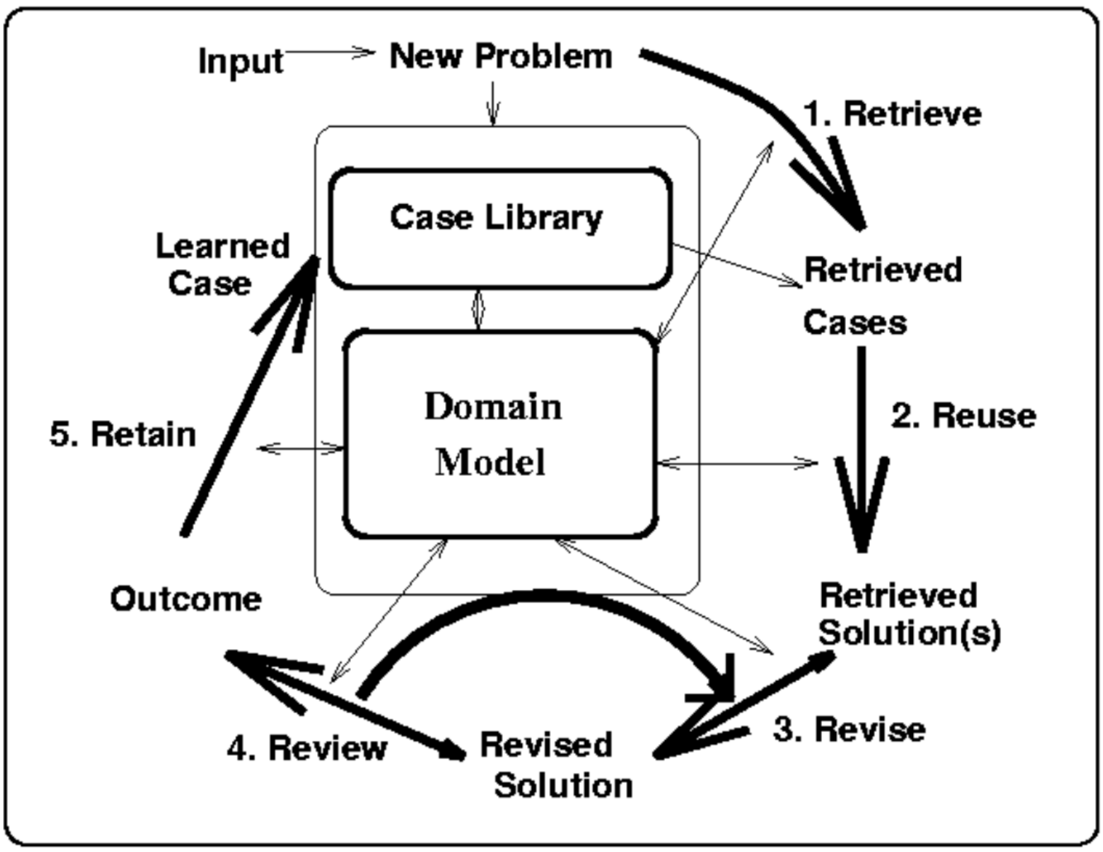
2.2 Case-Based

- Specific cases are specified by the user as targets or anchor points
- Similarity metrics on item attributes to retrieve similar items
- Query modified through user interaction or pruning
- Conversational style of critiquing



Case-Based Reasoning

- Store previous experiences (cases) in memory



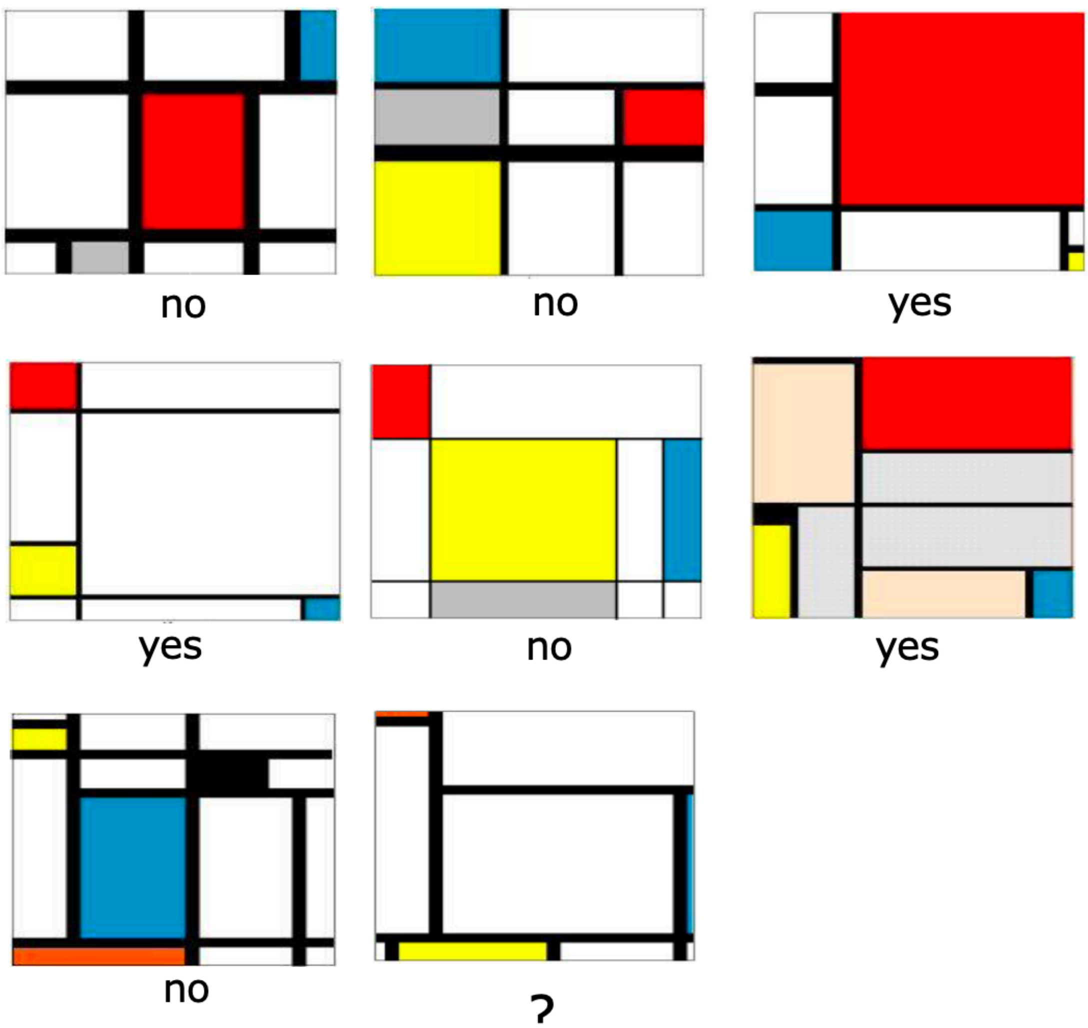
- Assumption: new problem can be solved by retrieving similar problems and adapting retrieved solutions
- Eg: Compiling solutions: "Patient N's heart symptoms can be explained in the same way as previous patient D's"

(a) kNN - Instance-Based Learning (Lazy Learner)

- Idea: store **all** training examples
  - When test instance comes, compute with all training instances
  - Find closest match (or k closest matches)
- Distance Measure: can use any

Example problem: identify if a pattern is the work of Mondrian

- Piet Mondrian was a Dutch painter and art theoretician
- Created unique pieces of artwork



- Training data (extract features like number of colours, number of lines, thickness of lines, number of rectangles)

Training data

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	6	1	10	4	No
2	4	2	8	5	No
3	5	2	7	4	Yes
4	5	1	8	4	Yes
5	5	1	10	5	No
6	6	1	8	6	Yes
7	7	1	14	5	No

- Test instance

Test instance

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	7	2	9	4	

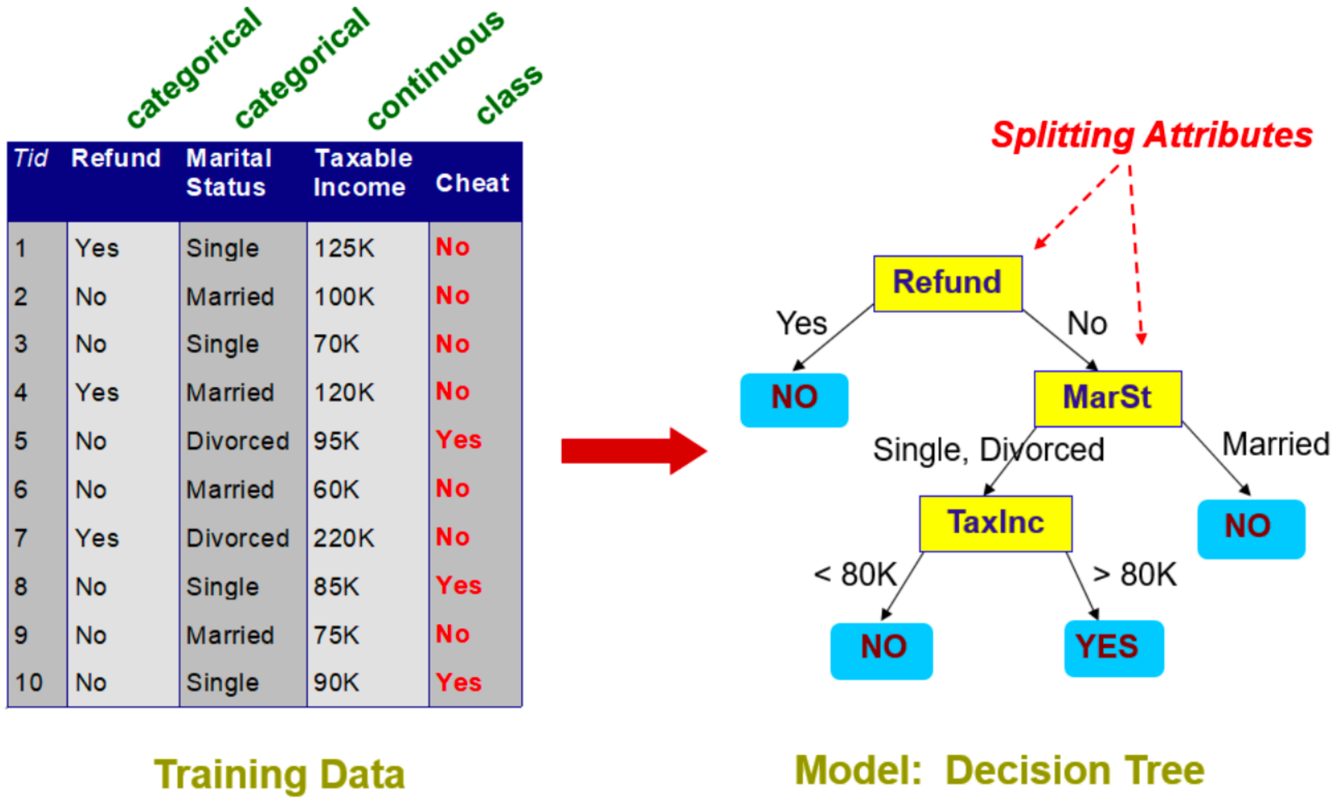
- Normalise features and find nearest neighbours using distance measure (check MI unit 2)

(b) Decision Trees (CART)

- Supervised learning
- Classification and Regression Tree
- Criteria to develop the tree
  1. Splitting criteria
  2. Merging criteria
  3. Stopping criteria (pruning)
- Impurity measures:
  - Gini index $(0-0.5)$
    - $I_G = 1 - \sum_{j=1}^{c} p_j^2$
    - p_i : proportion of samples that belong to class c for a particular node
  - Entropy $(0-1)$

- $I_H = -\sum_{j=1}^{c} p_j \log_2 (p_j)$
- p_i : proportion of samples that belong to class c for a particular node
- If all samples at a node belong to same class, entropy = 0

- SSE for continous



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

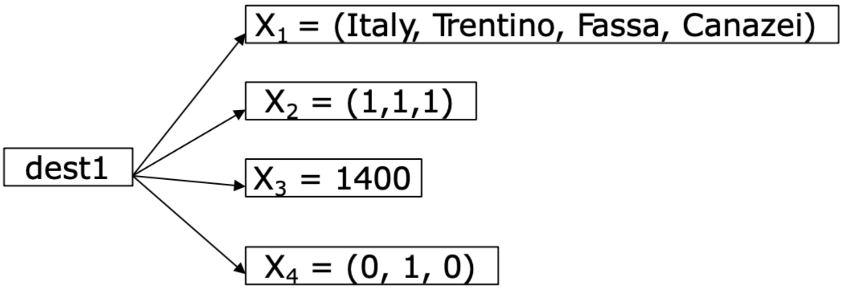
Query Augmentation

- Eg: searching for restaurant, "Thai" can be augmented to "Thai food"
- Eg: if "Thai food" fetches nothing, can augment to "Asian food"
- Eg: if "Asian food" fetches too many and user previously searched for "Chinese food", augment to "Chinese food"

Tree-based case representation

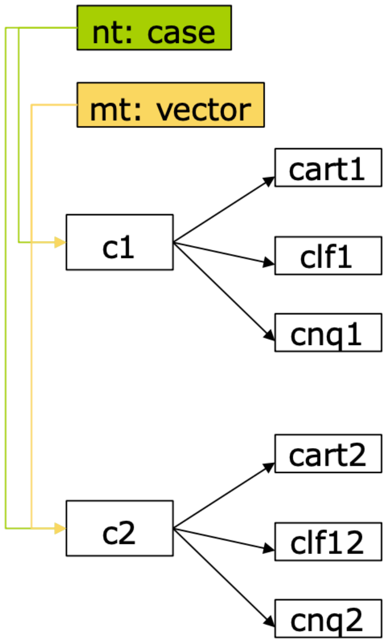
- Case: rooted tree
- Nodes: node type and metric type

	Node Type	Metric Type	Example: Canazei
X_1	LOCATION	Set of hierarchical related symbols	Country=ITALY, Region=TRENTINO, TouristArea=FASSA, Village=CANAZEI
X_2	INTERESTS	Array of Booleans	Hiking=1, Trekking=1, Biking=1
X_3	ALTITUDE	Numeric	1400
X_4	LOCTYPE	Array of Booleans	Urban=0, Mountain=1, Rivereside=0



- For querying: represent X as a vector $(x_1, x_2, \dots, x_n)$
  - $(Italy, Trentino, Fassa, Canazei, 1, 1, 1, 1400, 0, 1, 0)$
- Query: conjunction of constraints over features
  - $q = c_1 \wedge c_2 \wedge \dots \wedge c_m$ where $m \leq n$ and
  - $c_k = \begin{cases} x_{ik} = \text{true} & \text{if } x_{ik} \text{ is boolean} \\ x_{ik} = \nu & \text{if } x_{ik} \text{ is nominal} \\ l \leq x_{ik} \leq u & \text{if } x_{ik} \text{ is numerical} \end{cases}$
- Case distance

$$d(c_1,c_2) = \frac{1}{\sqrt{\sum_{i=1}^{3} W_i}} \sqrt{W_1 d(cart_1, cart_2)^2 + W_2 d(clf_1, clf_2)^2 + W_3 d(cnq_1, cnq_2)^2}$$



CBR Containers

1. Cases
2. Case representation language
3. Retrieval knowledge
4. Adaptation knowledge

3. Other Methods

Ensemble Methods - Bagging and Boosting

- Reduce bias and variance
- See MI unit 3
- More accurate, diverse than individual methods

(a) Bagging

- Boostrap aggregation
- Resampling
- Eg: random forest
- Goal: minimum variance

- Combine: majority vote
- Advantages
    - Reduce overfitting
    - Works with high dimensions
    - Maintains accuracy with missing data
- Disadvantages
    - Not precise predictions (mean prediction from subset trees)
    - Good for unstable algorithms but can hurt stable algorithm

(b) Boosting

- Reweight data
- All samples used (no resampling)
- Eg: adaboost
- Goal: maximum accuracy
- Combine: weighted average
- Advantages
    - Different loss functions
    - Works with interactions
- Disadvantages
    - Prone to overfitting
    - Careful hyperparameter tuning

SVM

- MI unit 2
- Identify the correct hyperplane
- Kernel trick: do not need to explicitly add a new dimension for non-linear data

ANN

- MI unit 2
- Can learn any non-linear function
- Also called Universal Function Approximators
- Activation functions introduce non-linearity
- Further eading: transfer learning

Clustering

- Group objects (unsupervised learning)

(a) K-means clustering

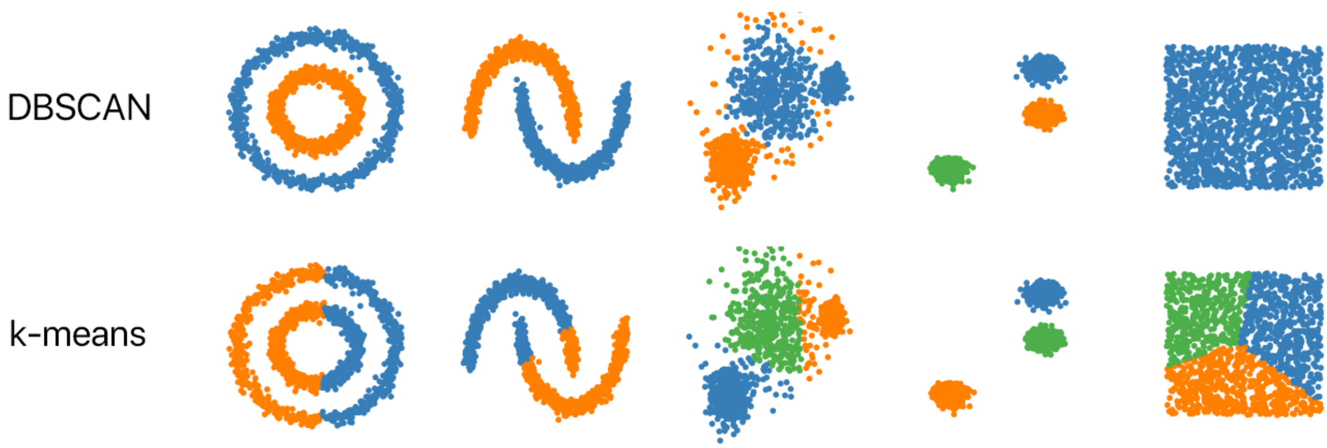
- EM algorithm - MI unit 4
- Time complexity $O(n \times k \times I \times d)$
- Within cluster SSE
- Does not work well for inherently nonglobular clusters
- Recommended number of clusters
  - $\mathrm{CH}(k) = \dfrac{B(k)/k-1}{W(k)/(n-k)}$

(b) Agglomerative clustering

- MI unit 4

(c) DBSCAN

- Clusters based on density
- Eg: concentric circles



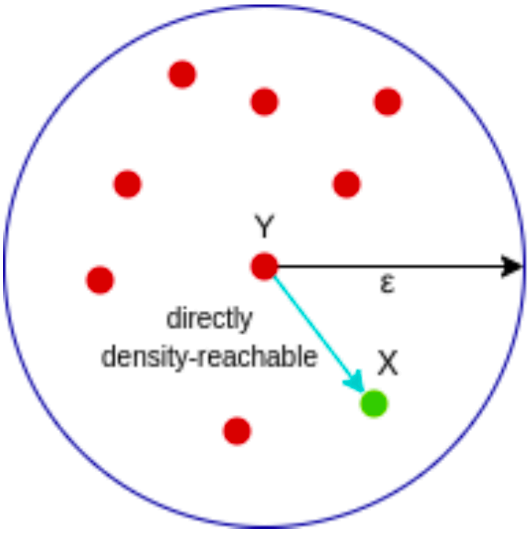
- Noise considered a different cluster
- Density-Based Spatial Clustering of Applications with Noise
- **Density:** number of points within a specified radius
- **Core point:** point that has more than MinPts number of points within radius of Eps
- **Border point:** point that has fewer than MinPts number of points within Eps, but is in the neighbourhood of a core point
- **Noise point:** point that is neither a core nor a border point

$current\_cluster\_label \leftarrow 1$
for all core points **do**
  if the core point has no cluster label **then**
    $current\_cluster\_label \leftarrow current\_cluster\_label + 1$
    Label the current core point with cluster label $current\_cluster\_label$
  end if
  for all points in the Eps-neighborhood, except i^{th} the point itself **do**
    if the point does not have a cluster label **then**
      Label the point with cluster label $current\_cluster\_label$
    end if
  end for
end for

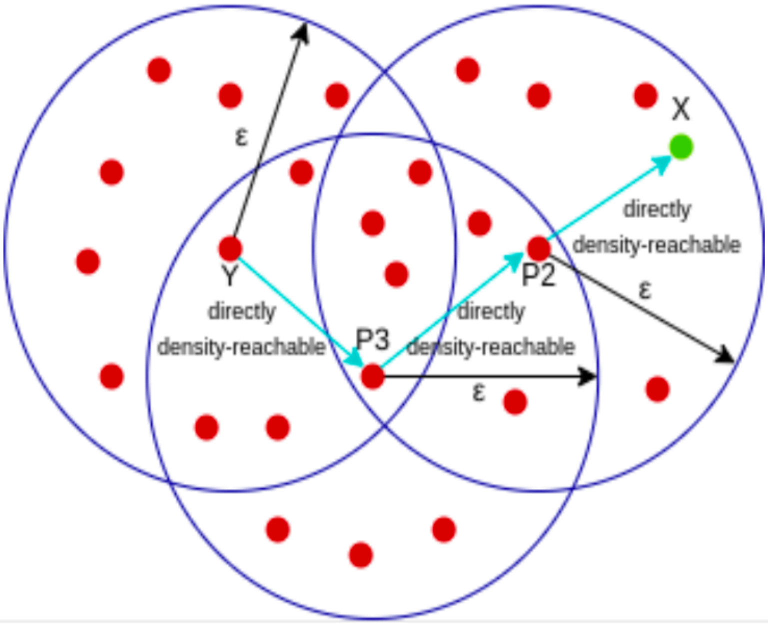
- Robust to outliers
- Does not require the number of clusters to be set beforehand
- Only epsilon (radius) and minpoints to be specified

Reachability and Connectivity

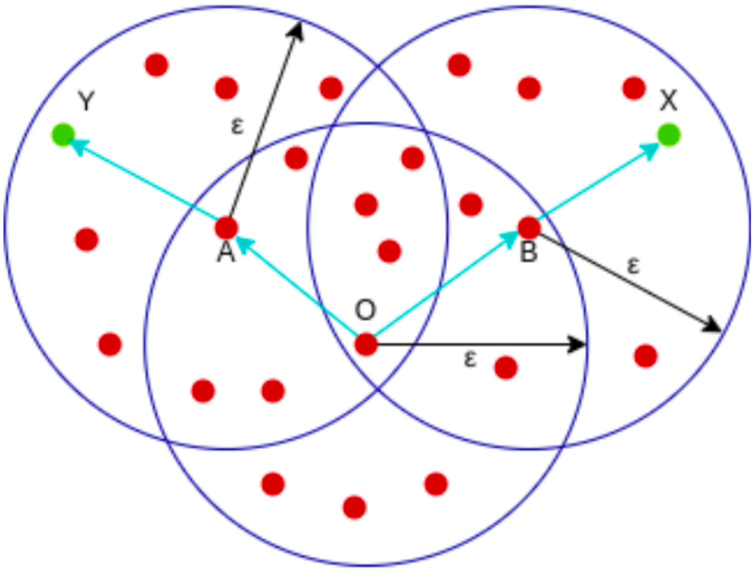
- A point X is directly density reachable from a point Y if X is a border point or core point in core point Y's neighbourhood



- A point is X indirectly density reachable from a point Y if X is directly rechable from a core point Z that is indirectly or directly reachable from Y



- Vice versa not true (X is not a core point)
- Two points X and Y are density connected if they both are density reachable from a common core point O



- DBSCAN is sensitive to parameters epsilon and minpoints
  - $\text{minPoints} \geq \text{Dimensions} + 1$
  - $\text{minPoints} \geq 3$
  - Generally, $\text{minPoints} = 2 \times \text{Dimensions}$

Cluster Cohesion

- How compact a cluster is - WCSS

Cluster Separation

- How distinct clusters are
- Between cluster sum of squares - BCSS
- $\text{BCSS} = \sum_i |C_i|(m - m_i)^2$ where C_i is the size of cluster i

4. Content-Based

- Content/desciption is exploited for recommendation
- Keywords, TF-IDF, tree of concepts
- Useful when few ratings available (cold start)
- Not much to do with other users; mainly target user's own ratings
- Dependent on 2 sources of data
  1. Description of various items (by manufacturer)
  2. User profile (generated from implicit/explicit feedback)
- Steps
  - Preprocessing and feature extraction
  - Content-based learning of user profiles
  - Filtering and recommendation

TF-IDF

- TF: term frequency - frequency of word in a document
- IDF: inverse document frequency - among the whole corpus of documents
- TF-IDF: product of TF and IDF

- Term frequency of a term t in document d
  - $\text{tf}_{t,d}$
- Weighted term frequency
  - $ \text{TF} = w_{t,d} = \left\{ \begin{array}{ c l }
    1 + \log_{10}\textrm{tf}_{t,d} & \quad \textrm{if tf}_{t,d} \gt 0 \\
    0   & \quad \textrm{otherwise }
  \end{array}\right.$
- Inverse document frequency for term i
  - $\text{IDF} = \log_{10}\left(\frac{n}{n_i}\right)$
  - n : total number of docs
  - n_i : number of documents in which the term i appears
  - Sometimes a 1 is added for smoothening

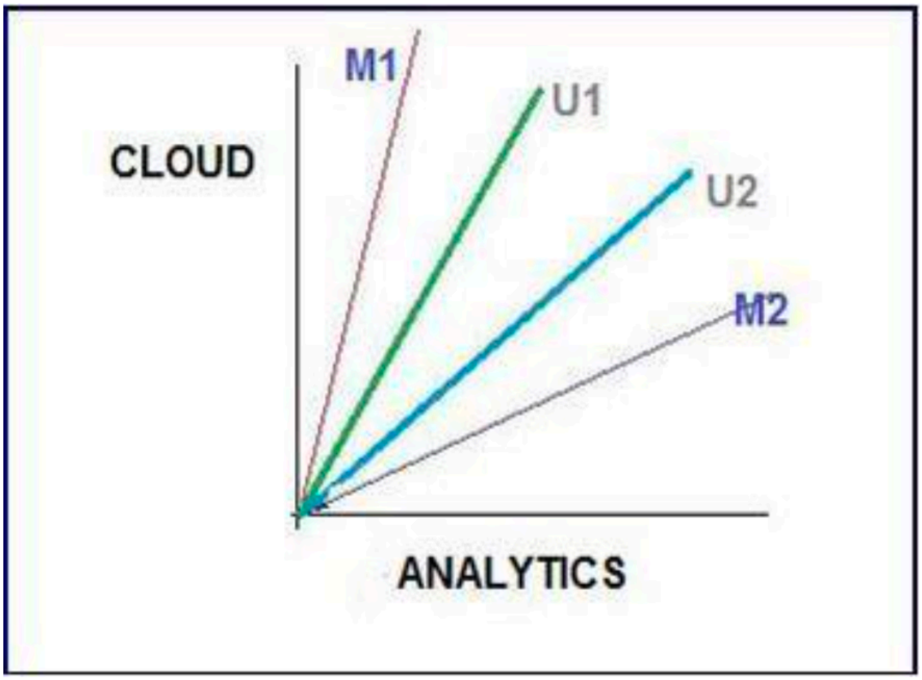
Text to Numbers

- Stop word removal
- Stemming (hoping -> hope)
    - Problem: hope -> hop
    - Lemmatisation
- Phrase extraction (n grams)

Vector Space Model

- Each item: vector of its attributes
- Similarity: angle between vectors
- User profile vectors also created

- Eg: Users U1, U2 and documents M1 and M2



Example problem

- Google search for "IoT and analytics"
- Top 5 links out of 1 million (corpus)

Articles	Analytics	Data	Cloud	Smart	Insight
Article 1	21	24	***0***	2	2
Article 2	24	59	2	1	***0***
Article 3	40	115	8	10	19
Article 4	4	28	5	***0***	1
Article 5	8	48	4	3	4
Article 6	17	49	8	***0***	5
DF	5,000	50,000	10,000	5,00,000	7000

- Calculate TF of article 1
  - $\mathrm{TF} = 1 + \log_{10} 21 = 1 + 1.3222 = 2.3222$
- Attribute vectors of each article

Articles	Analytics	Data	Cloud	Smart	Insight	Length of Vector
Article 1	2.322219295	2.380211242	0	1.301029996	1.301029996	**3.800456039**
Article 2	2.380211242	2.770852012	1.301029996	1	0	**4.004460697**
Article 3	2.602059991	3.06069784	1.903089987	2	2.278753601	**5.380804488**
Article 4	1.602059991	2.447158031	1.698970004	0	1	**3.527276247**
Article 5	1.903089987	2.681241237	1.602059991	1.477121255	1.602059991	**4.257450611**
Article 6	2.230448921	2.69019608	1.903089987	0	1.698970004	**4.326697114**

- Calculate cosine as dot product of unit vectors

Text Classification

- Bag of words: document is a dict of words and frequencies (independent of sequence)
- Document is sequence of words: n-grams, unigram, bigram

Feature Selection

- Stop word removal
- Stemming
- POS tagging
- Etc

Domains of Text Classification

- News filtering and Organization
- Document Organization and Retrieval
- Opinion Mining
- Email Classification and Spam Filtering

Naive Bayes Classifier

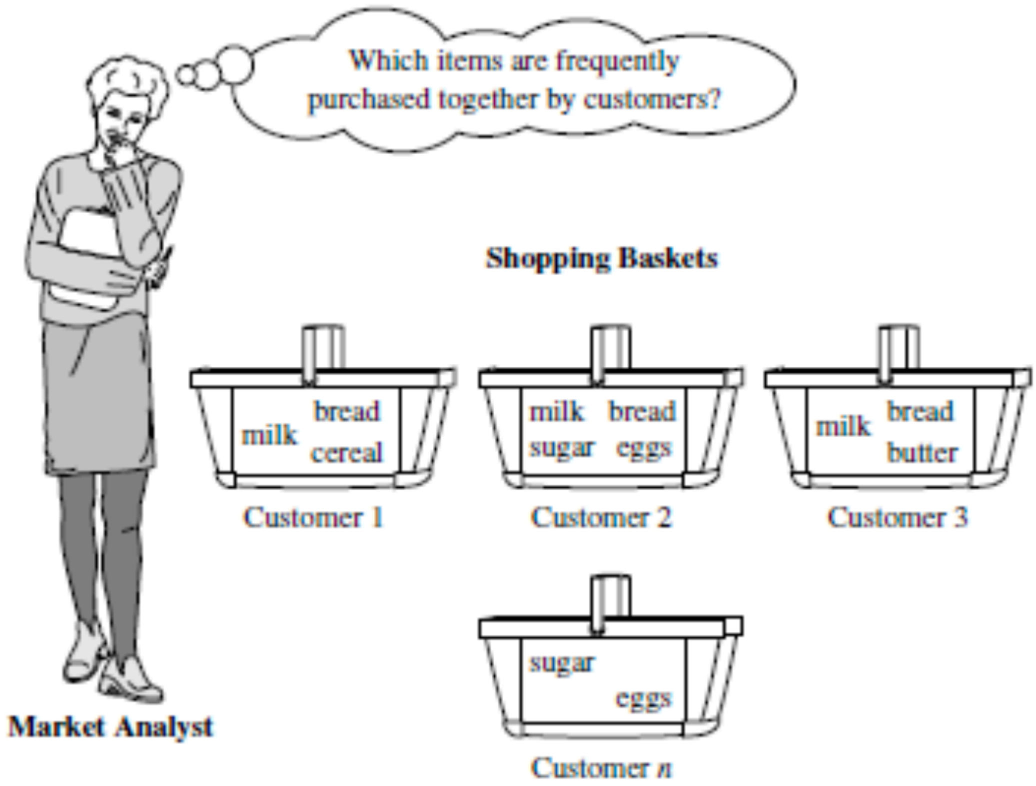
- MI unit 3
- Multivariate model: no frequencies
- Multinomial model: frequencies
- Bayes theorem
  - $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$

Mixture Models

- Clustering
- Unlabelled data is much more copiously available than labelled data
- When labelled data is sparse, it should be used in order to assist the classification process
- Documents in the same class are often mixtures of multiple topics
- Probability (not hard clustering)

Market Based Analysis (Frequent Itemset Mining)

- Describe many-many relationship between two kinds of objects
- Items and baskets
- **Basket:** contains a set of items - called items
  - Number of items in a basket small
  - Much smaller than total number of items
  - Eg: shopping cart
  - Number of baskets very large (cannot fit in memory)
- Data: file containing sequence of basket objects
- Associations between different items that customers place in their shopping baskets



- Helps decide placement of frequently bought together items

Frequent Itemset

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Itemset:** collection of one or more items
  - Eg: $\{\text{Milk, Bread, Diaper}\}$
  - k-itemset: itemset containing k items
- **Support count** σ
  - Frequency of occurence of an itemset
  - Eg: $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$
- **Support** s
  - Fraction of transactions that contain an itemset
  - Eg: $s(\{\text{Milk, Bread, Diaper}\}) = \frac{2}{5}$
- **Frequent itemset**
  - Itemset whose support $s \geq \text{minsup}$ threshold
- **Association rule**

- Implication expression of the form $X \to Y$ where X and Y are itemsets
- Eg: $\{\text{Milk, Diaper} \to \text{Beer}\}$

- **Confidence** c
    - How often items in Y appear in transactions that contain X from an association rule $X \to Y$
    - Eg:
    $$c(\{\text{Milk, Diaper} \to \{\text{Beer}\}) = P(\{\text{Beer}\}|\{\text{Milk, Diaper}\}) = \frac{\sigma(\{\text{Milk, Diaper, Beer}\})}{\sigma(\{\text{Milk, Diaper}\})} = \frac{2}{3}$$

Apriori Principle

- If an itemset is frequent, then all of its subsets must also be frequent
- $\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$
- Support of an itemset never exceeds the support of its subsets
- **Anti-monotone** property of support

Association Rule Mining

Two step approach

1. Frequent itemset generation
    - Generate all frequent itemsets (support $\geq$ minsup)
2. Rule generation
    - Generate high confidence rules from each frequent itemset
    - Each rule is a binary partitioning of a frequent itemset
    - Confidence does not necessarily have an **anti-monotone** property

Contingency Table for $X \to Y$

	Y	Y	
X	f_{11}	f_{10}	f_{1+}
X	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	|T|

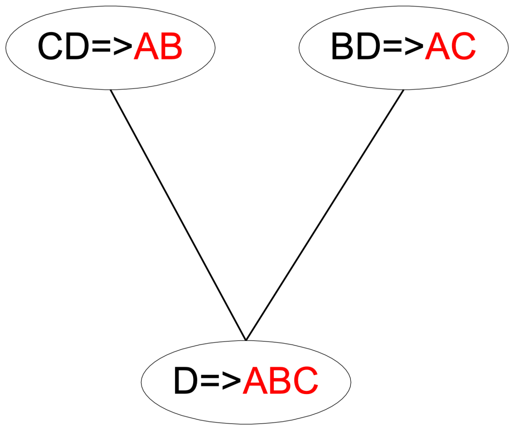
- f_{11} : support of X and Y
- f_{11} : support of X and $\bar{Y}$
- f_{11} : support of $\bar{X}$ and Y
- f_{11} : support of $\bar{X}$ and $\bar{Y}$

- **Lift** of $X \to Y$

- $$\frac{P(Y|X)}{P(Y)}$$
- **Interest** of $X \rightarrow Y$
  - $$\frac{P(X,Y)}{P(X)P(Y)}$$
- **PS**
  - $P(X,Y) - P(X)P(Y)$
- **ϕ-coefficient**
  - $$\frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)(1-P(X))P(Y)(1-P(Y))}}$$

Rule Generation

- Confidence of rules generated from the same itemset has an anti-monotone property
- **Candidate rule** : generated by merging two rules that share the same prefix in the rule consequent
  - $\text{Join}(CD \implies AB, BD \implies AC)$ produces $D \implies ABC$



- Prune rule $D \implies ABC$ if a subset $AD \implies BC$ does not have high confidence
- Given a frequent itemself L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement
  - Eg: $L = \{A, B, C, D\}$

ABC →D,	ABD →C,	ACD →B,	BCD →A,
A →BCD,	B →ACD,	C →ABD,	D →ABC
AB →CD,	AC → BD,	AD → BC,	BC →AD,
BD →AC,	CD →AB		

  - Here $|L| = 4$
  - If $|L| = k$ then there are $2^k - 2$ candidate association rules ($L \rightarrow \phi$ and $\phi \rightarrow L$ are omitted)

Example

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

1. Generating frequent itemsets for minsup = 3
   - 1-itemsets

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

   - 2-itemsets

Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

   - And so on
   - Prune itemsets that are not frequent
   - Set of k-itemsets that are frequent are denoted as L_k
2. Generate rules

Handling of Categorical Attributes

- More than 2 values
- **Potential solution:** Aggregate the low-support attribute values
- If highly skewed, can drop high frequency

Handling of Continuous Attributes

- Equal-width binning
- Equal-depth binning
- Clustering

Supervised: Attribute values, v

Class	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
Anomalous	0	0	20	10	20	0	0	0	0
Normal	150	100	0	0	0	100	100	150	100
	bin_1		bin_2			bin_3			

Evaluation of Recommender Systems

- Objective
- Subjective