

ClusteringInput: $S_n = \{\bar{x}^{(i)}\}_{i=1}^n, \bar{x} \in \mathbb{R}^d$ Output: a set of cluster assignments c_1, \dots, c_n , where $c_i \in \{1, \dots, k\}$ **K-means**Datapoints $\bar{x}^{(1)}, \dots, \bar{x}^{(n)}$ and fixed k . Initial means $\bar{\mu}^{(1)}, \dots, \bar{\mu}^{(k)}$
Iteratively,

1. reassign $\bar{x}^{(i)}$ to $c_i = \arg \min_j \|\bar{x}^{(i)} - \bar{\mu}^{(j)}\|^2$
2. recompute $\bar{\mu}^{(j)} = \frac{\sum_{i \in [c_i=j]} \bar{x}^{(i)}}{\sum_{i \in [c_i=j]} 1}$

That is, define $J(\bar{c}, \bar{\mu}) = \sum_{i=1}^n \|\bar{x}^{(i)} - \bar{\mu}^{(c_i)}\|^2$ First, fix $\bar{\mu}$, choose \bar{c} to minimize J . Second, fix \bar{c} , choose $\bar{\mu}$ to minimize J .

k-means performs coordinate descent on the objective function. k-means is guaranteed to converge (because objective function is monotonically decreasing), but not necessarily the global minimum. Solution: vary initialization of k-means, and pick clustering with lowest (eventual) objective function value (for a fixed k).

Spectral ClusteringWeight matrix: W , where w_{ij} represents the similarity between v_i and v_j .Degree of a vertex: $d_i = \sum_{j=1}^n w_{ij}$ Degree matrix D : $D_{ii} = \sum_{j=1}^n w_{ij}$, $D_{ij} = 0$ for $i \neq j$ Cost of a cut between A and \bar{A} : $\text{cut}(A, \bar{A}) = \sum_{i \in A} \sum_{j \in \bar{A}} w_{ij}$ Graph Laplacian: $L = D - W$.Build matrix with the first k eigenvectors (corresponding to the k smallest eigenvalues) as columns interpret rows as new data points
Apply k -means to new data representation**Hierarchical Clustering**

1. Assign each pt. its own cluster, for each point i , $C_i = \{\bar{x}^{(i)}\}$
2. Find the closest clusters & merge, repeat until convergence

$$\arg \min_{i,j} d(C_i, C_j) \text{ where } i \neq j$$

Recommender Systems**Approach #1: Nearest Neighbor Prediction**User: a , b . Y_{ai} is missing, i.e., a has not rated movie i .Define $R(a, b)$: set of movies rated by both users a and b

$$\tilde{Y}_{a:b} = \frac{1}{|R(a,b)|} \sum_{j \in R(a,b)} Y_{aj}$$

$$\text{sim}(a, b) = \frac{\sum_{j \in R(a,b)} (Y_{aj} - \tilde{Y}_{a:b}) (Y_{bj} - \tilde{Y}_{b:a})}{\sqrt{\sum_{j \in R(a,b)} (Y_{aj} - \tilde{Y}_{a:b})^2 \sum_{j \in R(a,b)} (Y_{bj} - \tilde{Y}_{b:a})^2}}$$

Define $KNN(a, i)$: k nearest neighbors, i.e., the k most similar users to a , who have rated movie.

$$\hat{Y}_{ai} = \bar{Y}_a + \frac{1}{\sum_{b \in KNN(a,i)} |\text{sim}(a, b)|} \sum_{b \in KNN(a,i)} \text{sim}(a, b) (Y_{bi} - \bar{Y}_b)$$

Approach #2: Matrix FactorizationLet $A \in \mathbb{R}^{n \times m}$, and $\text{rank}(A) = r$. Then there exists $X \in \mathbb{R}^{n \times r}$, $Y \in \mathbb{R}^{r \times m}$, such that $A = XY$ Given Y with empty cells, construct low rank- d \hat{Y} with no empty cells. We may think of Y as being approximated by $\hat{Y} = UV^T$, where $U \in \mathbb{R}^{n \times d}$ contains the relevant features of the user, and $V \in \mathbb{R}^{m \times d}$ contains the relevant features of the movie.

$$J(U, V) = \frac{1}{2} \sum_{(a,i) \in D} (Y_{ai} - [UV^T]_{ai})^2 + \frac{\lambda}{2} \sum_{a=1}^n \sum_{k=1}^d U_{ak}^2 + \frac{\lambda}{2} \sum_{i=1}^m \sum_{k=1}^d V_{ik}^2$$

$$U = [\bar{u}^{(1)}, \dots, \bar{u}^{(n)}]^T, \text{ and } V^T = [\bar{v}^{(1)}, \dots, \bar{v}^{(m)}]$$

Algorithm Overview (coordinate descent)

1. Initialize V to small (random) values.

2. Iterate until convergence

Fix $\bar{v}^{(1)}, \dots, \bar{v}^{(m)}$. Solve for $\bar{u}^{(1)}, \dots, \bar{u}^{(n)}$

$$\min_{\bar{u}^{(a)}} \frac{1}{2} \sum_{i \in D_a} (Y_{ai} - \bar{u}^{(a)} \cdot \bar{v}^{(i)})^2 + \frac{\lambda}{2} \|\bar{u}^{(a)}\|^2$$

Fix $\bar{u}^{(1)}, \dots, \bar{u}^{(n)}$. Solve for $\bar{v}^{(1)}, \dots, \bar{v}^{(m)}$

$$\min_{\bar{v}^{(i)}} \frac{1}{2} \sum_{a \in D_i} (Y_{ai} - \bar{u}^{(a)} \cdot \bar{v}^{(i)})^2 + \frac{\lambda}{2} \|\bar{v}^{(i)}\|^2$$

i.i.d assumptionIdentically drawn from \mathcal{D} , each trial independent from each other.**Gaussian Distribution**

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

Spherical Gaussian

$$\mathcal{N}(\bar{x}|\bar{\mu}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|\bar{x} - \bar{\mu}\|^2\right)$$

MLE results

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^n \bar{x}^{(i)}, \sigma^2 = \frac{1}{nd} \sum_{i=1}^n \|\bar{x}^{(i)} - \bar{\mu}\|^2$$

GMM, EM

Iterate until convergence:

E step: use current estimate of mixture model to assign examples to clusters

M step: re-estimate each cluster model separately based on the points assigned to it (similar to the “known cluster” case)

Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{\sum_A P(B|A)P(A)}$$

E-step: fix parameters $\bar{\theta} = [\gamma_1, \dots, \gamma_k, \bar{\mu}^{(1)}, \dots, \bar{\mu}^{(k)}, \sigma_1^2, \dots, \sigma_k^2]$, compute posterior distribution

$$P(j|i) = \frac{\gamma_i \cdot \mathcal{N}(\bar{x}^{(i)}|\bar{\mu}^{(j)}, \sigma_j^2)}{\sum_{t=1}^k \gamma_t \cdot \mathcal{N}(\bar{x}^{(i)}|\bar{\mu}^{(t)}, \sigma_t^2)}$$

M-step: fix posterior distribution $P(j|i)$, compute MLE parameters $\bar{\theta}$.

$$n_j = \sum_{i=1}^n P(j|i), \quad \gamma_j = \frac{n_j}{n}$$

$$\bar{\mu}^{(j)} = \frac{1}{n_j} \sum_{i=1}^n P(j|i) \bar{x}^{(i)}, \quad \sigma_j^2 = \frac{1}{n_j d} \sum_{i=1}^n P(j|i) \|\bar{x}^{(i)} - \bar{\mu}^{(j)}\|^2$$

EM in general**Bayesian Networks**

For a given graph, the joint distribution can be written as a product of the conditional probability of each variable given its parents

$$P(X_1, \dots, X_d) = \prod_{i=1}^d P(X_i | X_{pa(i)})$$

Three Rules

1. $P(X, Y) = P(X|Y)P(Y)$
2. $P(X) = \sum_Y P(X, Y)$
3. $\sum_X P(X|Y) = 1$

d-separation

Step 1: keep only ancestral graph of the variables of interest

Step 2: (“moralize”) add undirected edge between any two variables in ancestral graph that have a common child & change to undirected.

Step 3: (i) if there is **no** path between variables of interest, they are marginally independent.

(ii) if **all** paths go through a particular node, then the variables are independent given that node.

MLE learning on Bayesian Networks

log-likelihood: $l(\theta; S_n, G) = \sum_{t=1}^n \sum_{i=1}^d \log \theta_i(x_i^{(t)} | x_{pai}^{(t)})$

Corresponding MLE:

$$\hat{\theta}_i(x_i | x_{pai}) = \frac{\#(X_i = x_i, X_{pai} = x_{pai})}{\sum_{x'_i} \#(X_i = x'_i, X_{pai} = x_{pai})}$$

Bayesian Information Criterion, n is number of training data

$$\text{BIC}(D; \bar{\theta}) = l(D; \bar{\theta}) - \frac{\# \text{param}}{2} \log(n)$$

Hidden Markov Model (HMM)