Unsupervised Learning

Yuyan Zhou

May 20, 2020

Types of learning

- Supervised learning
  - Classification
  - Regression
- Semi-supervised learning
- Active learning
- Unsupervised learning
  - Clustering
  - Dimension reduction
- Reinforcement learning
- …

labelled data. $x \xrightarrow{f} y$

Cluster analysis

- Top-down
- Bottom-up
- Key questions:
  - How to measure proximity
  - How to choose the number of clusters
  - Initialization

k too large n
too small 1

Proximity matrices

- N objects, N by N matrix D to describe their pairwise proximity
- Nonnegative? $D_{ij} \geq 0$?
- Symmetric? $D_{ij} = D_{ji}$
- Can be suitably convert to a dissimilarity matrix?
- Triangle inequality? $D_{ij} + D_{jk} > D_{ik}$?

Common dissimilarities

- N measurements x_{ij} for i=1,2,⋯,N and j=1,2,⋯,p (variables/attributes)
- Define $D(x_i, x_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$
- Most common $d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$
- Equivalent to based on correlation (similarity)

minimize $\sum (x_{ij} - x_{i'j})^2 \propto 2(1 - \rho(x_i, x_{i'}))$ ↑ maximize.

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}}$$

$\bar{x}_i = \frac{1}{p}\sum_j x_{ij}$.

Weighted dissimilarity

- Convex combination

$$D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot d_j(x_{ij}, x_{i'j}); \quad \sum_{j=1}^{p} w_j = 1.$$

$$\bar{D} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot \bar{d}_j,$$

weighted influence of attribute j.

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j})$$

Weighted squared Euclidean distance

$$D_I(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot (x_{ij} - x_{i'j})^2$$

d

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} (x_{ij} - x_{i'j})^2 = 2 \cdot \mathrm{var}_j,$$

influence of j over the data set.

1° $w_j = 1$

2° $w_j = \frac{1}{d_j}$ ✓

Clustering algorithms

$C(N) = i \quad i \in \{1, \dots, K\}$.

$N \longrightarrow K \qquad S_2(N, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^N$.

$S_2(19, 4) \approx 10^{10}$.

- Combinatorial algorithms
  - Work directly on the observed data
  - No direct inference to an underlying probability model
  - Encoder, many-to-one mapping
  - Local optimal, small-sized
- Mixture modeling
  - MLE, Bayesian approach
- Mode seeking
  - Bump hunting
  - PRIM

ESL. 9.3

Common Heuristic in Practice: The Lloyd's method

k-means.

[Least squares quantization in PCM, Lloyd, IEEE Transactions on Information Theory, 1982]

Input: A set of n datapoints $\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^n}$ in $\mathrm{R^d}$

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Initialize centers $\boldsymbol{c_1}, \mathbf{c_2}, \ldots, \boldsymbol{c_k} \in \mathrm{R^d}$ and clusters $\mathrm{C_1}, \mathrm{C_2}, \ldots, \mathrm{C_k}$ in any way.

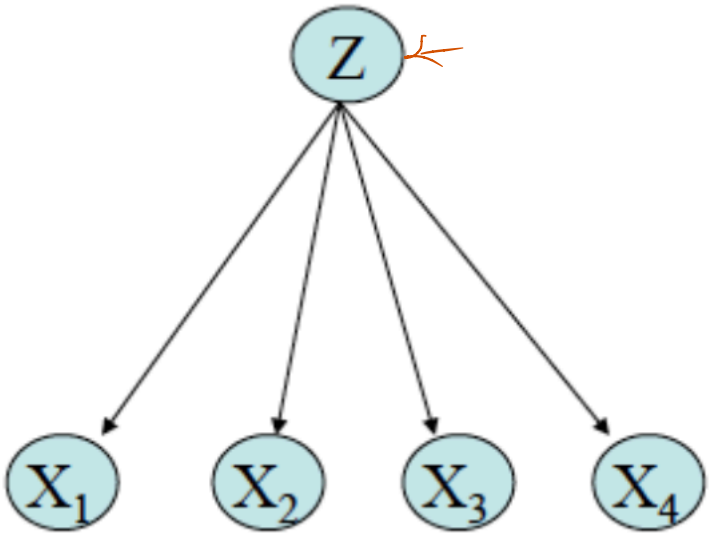
Repeat until there is no further change in the cost.

- For each j: $\mathrm{C_j} \leftarrow \{x \in S$ whose closest center is $\mathbf{c_j}\}$
- For each j: $\mathbf{c_j} \leftarrow$ mean of $\mathrm{C_j}$

Holding $\boldsymbol{c_1}, \mathbf{c_2}, \ldots, \boldsymbol{c_k}$ fixed, pick optimal $\mathrm{C_1}, \mathrm{C_2}, \ldots, \mathrm{C_k}$

Holding $\mathrm{C_1}, \mathrm{C_2}, \ldots, \mathrm{C_k}$ fixed, pick optimal $\boldsymbol{c_1}, \mathbf{c_2}, \ldots, \boldsymbol{c_k}$

Gaussian mixtures as soft K-means



K clusters $P(k=k|\vec{\pi})$

$P(x|k=k) = \prod_i N(x_i|\mu_k, \sigma^2)$

$P(x) = \sum_{k=1}^{K} P(k=k|\vec{\pi}) \cdot \prod_i N(x_i|\mu_k, \sigma^2)$.

E-step: $P(z|x,\theta)$.

M-step: $\max_{\theta'} E_{z|x,\theta} \log P(x,z|\theta')$.

$\sigma^2 \to 0$ soft K-means $\longrightarrow$ K-means.

Hierarchical clustering

- Bottom-up
- Top-down
- Different linkage
  - Single linkage
  - Complete linkage
  - Average linkage
  - Ward’ s linkage

Spectral clustering

- Non-convex clusters
- Undirected similarity graph G=<V, E>
- Graph-partition problem
  - Edges between different groups have low weight
  - Within a group have high weight
- Random walk on the graph with transition probability matrix P
  - Construct groups that the random walk seldom transitions from one group to another

PCA

$X \in \mathbb{R}^{D \times n}$ $x_1, x_2, \dots, x_n \in \mathbb{R}^D$

$V \in \mathbb{R}^{D \times d}$ $d \ll D$.

- Principal components
  - Projections of data ✓
  - Mutually uncorrelated (orthogonal) $v_i \cdot v_j = 0$ $i \neq j$. $v_i \cdot v_j = 1$ $i = j$ $\Rightarrow \|v_i\| = 1$
  - Ordered in variance

X 中心化. $x_i - \bar{x}$

$\frac{1}{n} \sum_{i=1}^{n} (V^T x_i)^2 = V^T X X^T V$.

maximize $V^T X X^T V$

s.t. $V^T V = I$.

$\partial (V^T X X^T V - \lambda V^T V) / \partial V = 0 \Rightarrow (XX^T)V = V\Lambda$

$\Lambda = \begin{bmatrix} d & & & \\ & 0 & & \\ & & 0 & \\ & & & \ddots \\ & & & & 0 \end{bmatrix}$

Kernel principal components

$\phi: \mathbb{R}^D \to \mathbb{R}^P$ $F: \mathbb{R}^P$ $D<P$ | $k(x,y)=\phi(x)^T\phi(y)$. inner product.

$X=[x_1,\dots,x_n]$ ✓

$\phi(X)=[\phi(x_1),\phi(x_2),\dots,\phi(x_n)] \in \mathbb{R}^{P\times n}$ ✓

中心化.

cov matrix: $C_F=\frac{1}{N}\phi(X)\phi(X)^T=\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T$

$C_F \cdot p=\lambda p$. ①

$\lambda\neq 0 \Rightarrow p=\frac{1}{\lambda}\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T p=\sum_{i=1}^{N} a_i \phi(x_i)=\phi(X)\cdot\alpha$ $p=\phi(X)\alpha$.

a_i $\alpha=\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}$

① $\phi(X)\phi(X)^T\phi(X)\alpha=\lambda\phi(X)\alpha$. ✓

同时乘 $\phi(X)^T$.

② $\phi(X)^T\phi(X)\phi(X)^T\phi(X)\alpha=\lambda\phi(X)^T\phi(X)\alpha$.

$K=\phi(X)^T\phi(X) \in \mathbb{R}^{N\times N}$ $K_{ij}=\phi(x_i)^T\phi(x_j)$

② $K\cdot K\alpha=\lambda K\alpha \Rightarrow K\alpha=\lambda\alpha$.

ICA

- PCA finds directions of maximum variation
- ICA would find directions most "aligned" with data
  - Statistically independent linear transformation
  - Optimize based on mutual inforamtion