

Help-sheet for Fundamentals of Big Data Analytics

Principle Component Analysis (PCA).

Samples: $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$,
 Samp. mean: $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$,
 Samp. covariance: $\mathbf{S}_n = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$,
 Spec. decomposition: $\mathbf{S}_n = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$
 $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$
 with $\lambda_1 \geq \dots \geq \lambda_p$
 $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_p) \in \mathcal{O}(p)$.
 $\mathbf{S}_n \mathbf{v}_i = \lambda_i \mathbf{v}_i$
 Projection: $\hat{\mathbf{x}}_i = \begin{pmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_k^T \end{pmatrix} \mathbf{x}_i, \quad i = 1, \dots, n$
 $\mathbf{Q} = \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T$.

Marchenko-Pastur(MP) and Spike Model.

MP density: $p, n \rightarrow \infty, \frac{p}{n} \rightarrow \gamma \in (0, 1]$
 $f_\gamma(u) = \frac{1}{2\pi\sigma^2 u \gamma} \sqrt{(b-u)(u-a)}, \quad a \leq u \leq b$
 $a(\gamma) = \sigma^2(1 - \sqrt{\gamma})^2$ and $b(\gamma) = \sigma^2(1 + \sqrt{\gamma})^2$
 Spike Model: $\text{Cov}(\mathbf{X}) = \mathbf{I}_p + \beta \mathbf{v}\mathbf{v}^T$
 If $\beta \leq \sqrt{\gamma}$:
 $|\langle \mathbf{v}_{\max}, \mathbf{v} \rangle| \rightarrow 0, \lambda_{\max} \rightarrow (1 + \sqrt{\gamma})^2$
 If $\beta > \sqrt{\gamma}$:
 $\lambda_{\max} \rightarrow (1 + \beta)(1 + \frac{\gamma}{\beta}), |\langle \mathbf{v}_{\max}, \mathbf{v} \rangle| \rightarrow \frac{1-\gamma/\beta^2}{1-\gamma/\beta}$.

Multidimensional Scaling (MDS).

Dissimilarity matrix: $\mathbf{\Delta} = (\delta_{ij})_{1 \leq i, j \leq n}$
 $-\frac{1}{2} \mathbf{E}_n \mathbf{\Delta}^{(2)} \mathbf{E}_n = \mathbf{V} \text{diag}(\lambda_1, \dots, \lambda_n) \mathbf{V}^T$ Non-separable case:
 MDS embedding: $\mathbf{X}^* = \left[\sqrt{\lambda_1^+} \mathbf{v}_1, \dots, \sqrt{\lambda_k^+} \mathbf{v}_k \right] \in \mathbb{R}^{n \times k}$.

Diffusion Maps.

Transition Matrix: $w_{ij} = K_\epsilon(\|\mathbf{x}_i - \mathbf{x}_j\|)$
 $\mathbf{W} = (w_{ij})_{1 \leq i, j \leq n}, \text{deg}(i) = \sum_j w_{ij}$
 $\mathbf{D} = \text{diag}(\text{deg}(1), \dots, \text{deg}(n))$,
 $\mathbf{M} = \mathbf{D}^{-1} \mathbf{W}$
 $\mathbf{S} = \mathbf{D}^{1/2} \mathbf{M} \mathbf{D}^{-1/2} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$
 $\mathbf{M} = \mathbf{D}^{-1/2} \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \mathbf{D}^{1/2} = \mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Psi}$
 Diffusion map: $\phi_t(i) = \begin{pmatrix} \lambda_1^t \phi_{1,i} \\ \vdots \\ \lambda_n^t \phi_{n,i} \end{pmatrix}, i = 1, \dots, n$

Linear Discriminant Analysis (LDA).

System parameters: $\mathbf{B} = \sum_{l=1}^g n_l (\bar{\mathbf{x}}_l - \bar{\mathbf{x}})(\bar{\mathbf{x}}_l - \bar{\mathbf{x}})^T$.
 $\mathbf{W} = \sum_{l=1}^g \mathbf{X}_l^T \mathbf{E}_l \mathbf{X}_l$,
 Disc. vector \mathbf{a} : dominant eigenvector of $\mathbf{W}^{-1} \mathbf{B}$
 Disc. rule: $|\mathbf{a}^T \mathbf{x} - \mathbf{a}^T \bar{\mathbf{x}}_l| < |\mathbf{a}^T \mathbf{x} - \mathbf{a}^T \bar{\mathbf{x}}_j|$
 for all $j \neq l$.

Gaussian ML Discriminant Rule.

ML rule: minimizes the Mahalanobis distance:
 $(\mathbf{x} - \boldsymbol{\mu}_l)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_l)$.
 ML for two classes: $\boldsymbol{\alpha}^T (\mathbf{x} - \boldsymbol{\mu}) > 0$,
 $\boldsymbol{\alpha} = \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ and $\boldsymbol{\mu} = \frac{1}{2} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$.

k-means clustering.

Problem: $\min_{\mathbf{S}_1, \dots, \mathbf{S}_k, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k} \sum_{l=1}^k \sum_{i \in \mathbf{S}_l} \|\mathbf{x}_i - \boldsymbol{\mu}_l\|$,
 Lloyd's algorithm: Iterate:
 Step 1: Given $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$ find clusters:
 $l = \arg \min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j\|$,
 Step 2: Given clusters find $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$

Spectral clustering.

Algorithm: Step 1: compute the diffusion map:
 $\phi_t(i) = \begin{pmatrix} \lambda_1^t \phi_{1,i} \\ \vdots \\ \lambda_n^t \phi_{n,i} \end{pmatrix}, i = 1, \dots, n$
 Step 2: run k-means clustering on $\phi_t(v_i)$

Support Vector Machines (SVM).

The optimal margin classifier:

$$\min_{\mathbf{a} \in \mathbb{R}^p, b \in \mathbb{R}} \frac{1}{2} \|\mathbf{a}\|^2$$

s.t. $y_i (\mathbf{a}^T \mathbf{x}_i + b) \geq 1, i = 1, \dots, n$.

$$\min_{\mathbf{a}, b, \xi} \frac{1}{2} \|\mathbf{a}\|^2 + c \sum_{i=1}^n \xi_i$$

s.t. $y_i (\mathbf{a}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1, \dots, n$
 $\xi_i \geq 0 \quad i = 1, \dots, n$.

The dual problem:

$$W(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i,j} y_i y_j \lambda_i \lambda_j \mathbf{x}_i^T \mathbf{x}_j$$

s.t. $0 \leq \lambda_i \leq c$
 $\sum_{i=1}^n \lambda_i y_i = 0$.

Kernels:

$$\phi : \mathbb{R}^p \rightarrow \mathcal{V}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

Linear regression.

$$\mathbf{X} = \begin{pmatrix} 1 & \mathbf{x}_1^T \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^T \end{pmatrix}$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\vartheta} + \boldsymbol{\epsilon}$$

$$\boldsymbol{\vartheta}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

1-dim. regression: $y_i = \vartheta_0 + \vartheta_1 x_i + \epsilon_i, i = 1, \dots, n$

$$\vartheta_1^* = \frac{\sigma_{xy}}{\sigma_x^2},$$

$$\vartheta_0^* = \bar{y} - \vartheta_1^* \bar{x},$$

$$\sigma_x^2 = \frac{1}{n} \sum_i x_i^2 - \bar{x}^2$$

$$\sigma_{xy} = \frac{1}{n} \sum_i x_i y_i - \bar{x} \cdot \bar{y}$$

Logistic regression.

$$h_{\boldsymbol{\vartheta}}(\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\vartheta}^T \mathbf{x}}}$$

Likelihood: $L(\boldsymbol{\vartheta}) = \prod_{i=1}^n h_{\boldsymbol{\vartheta}}(\mathbf{x}_i)^{y_i} (1 - h_{\boldsymbol{\vartheta}}(\mathbf{x}_i))^{1-y_i},$

Log-likelihood: $l(\boldsymbol{\vartheta}) = \log L(\boldsymbol{\vartheta}),$

Objective: $\max_{\boldsymbol{\vartheta} \in \mathbb{R}^{p+1}} l(\boldsymbol{\vartheta}),$

Update rule: $v_j^{(k+1)} = v_j^{(k)} + \alpha \sum_{i=1}^n (y_i - h_{\boldsymbol{\vartheta}^{(k)}}(\mathbf{x}_i)) x_{ij}.$

Perceptron algorithm.

$$h_{\boldsymbol{\vartheta}}(\mathbf{x}) = g(\boldsymbol{\vartheta}^T \mathbf{x})$$

$$g(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0. \end{cases}$$

Update rule: $v_j^{(k+1)} = v_j^{(k)} + \alpha \sum_{i=1}^n (y_i - h_{\boldsymbol{\vartheta}^{(k)}}(\mathbf{x}_i)) x_{ij}.$