Help-sheet for Fundamentals of Big Data Analytics

PCA.

Samples: $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$,
Samp. mean: $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$,
Samp. covariance: $\mathbf{S}_n = \frac{1}{n-1}\sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T$,
Spec. decomposition: $\mathbf{S}_n = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$
$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_p)$
with $\lambda_1 \geq \cdots \geq \lambda_p$
$\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_p) \in \mathcal{O}(p)$.
$\mathbf{S}_n \mathbf{v}_i = \lambda_i \mathbf{v}_i$
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.

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

MDS.

Dissimilarity matrix: $\mathbf{\Delta} = (\delta_{ij})_{1 \leq i,j \leq n}$
$-\frac{1}{2}\mathbf{E}_n \mathbf{\Delta}^{(2)} \mathbf{E}_n = \mathbf{V}\mathrm{diag}(\lambda_1, \dots, \lambda_n)\mathbf{V}^T$
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: $w_{ij} = K_\epsilon(\|\mathbf{x}_i - \mathbf{x}_j\|)$
$\mathbf{W} = (w_{ij})_{1 \leq i,j \leq n}$, $\deg(i) = \sum_j w_{ij}$
$\mathbf{D} = \mathrm{diag}(\deg(1), \dots, \deg(n))$,
$\mathbf{M} = \mathbf{D}^{-1}\mathbf{W}$
$\mathbf{S} = \mathbf{D}^{1/2}\mathbf{M}\mathbf{D}^{-1/2} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$
$\mathbf{M} = \mathbf{D}^{-1/2}\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{D}^{1/2} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Psi}$
Diffusion map: $\boldsymbol{\phi}_t(i) = \begin{pmatrix} \lambda_1^t \phi_{1,i} \\ \vdots \\ \lambda_n^t \phi_{n,i} \end{pmatrix}, i = 1, \dots, n$

LDA.

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

Gaussian ML Discriminant Rule.

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

$k-$means clustering.

Problem: $\min_{\substack{S_1,\dots,S_k \\ \boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_k}} \sum_{l=1}^k \sum_{i \in S_l} \|\mathbf{x}_i - \boldsymbol{\mu}_l\|$,
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:
$\boldsymbol{\phi}_t(i) = \begin{pmatrix} \lambda_1^t \phi_{1,i} \\ \vdots \\ \lambda_n^t \phi_{n,i} \end{pmatrix}, i = 1, \dots, n$
Step 2: run $k-$means clustering on $\boldsymbol{\phi}_t(v_i)$

Support vector machines.

The optimal margin classifier:

$$\min_{\mathbf{a} \in \mathbb{R}^p,\, b \in \mathbb{R}} \frac{1}{2}\|\mathbf{a}\|^2$$
$$\text{s.t. } y_i\big(\mathbf{a}^T\mathbf{x}_i + b\big) \geq 1, i = 1, \dots, n.$$

Non-separable case:

$$\min_{\mathbf{a}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{a}\|^2 + c\sum_{i=1}^n \xi_n$$
$$\text{s.t.} \quad y_i(\mathbf{a}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1, \dots, n$$
$$\xi_i \geq 0 \quad i = 1, \dots, n.$$

The dual problem:

$$W(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}} \sum_{i=1}^n \lambda_i - \frac{1}{2}\sum_{i,j} y_i y_j \lambda_i \lambda_j \mathbf{x}_i^T \mathbf{x}_j$$
$$\text{s.t.} \quad 0 \leq \lambda_i \leq c$$
$$\sum_{i=1}^n \lambda_i y_i = 0.$$

Kernels:

$$\boldsymbol{\phi} : \mathbb{R}^p \to \mathcal{V}$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j)$$

Linear regression.

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

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

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

1-dim. regression:

$$y_i = \vartheta_0 + \vartheta_1 x_i + \epsilon_i,\ i = 1, \ldots, n$$
$$\vartheta_1^\star = \tfrac{\sigma_{xy}}{\sigma_x^2},$$
$$\vartheta_0^\star = \overline{y} - \vartheta_1^\star \overline{x},$$
$$\sigma_x^2 = \frac{1}{n}\sum_i x_i^2 - \overline{x}^2$$
$$\sigma_{xy} = \frac{1}{n}\sum_i x_i y_i - \overline{xy}.$$

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

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