Kernel pca (see ipynb)

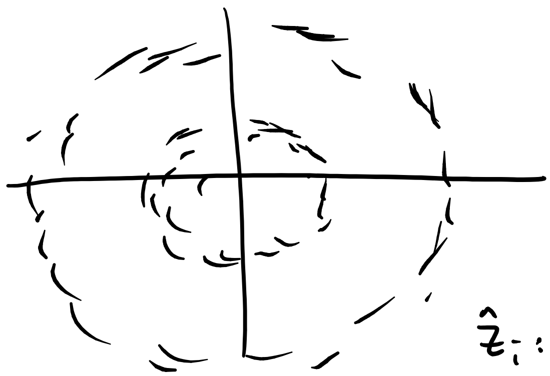
Thursday, May 11, 2017 10:58 AM

$K_{ij} = z_i^T z_j = \Phi(x_i)^T \Phi(x_j)$

$= k(x_i, x_j)$

SVD of X: $X = U \Sigma V^T$

Apply PCA to $Z = \Phi(X) = U \Sigma V^T$



$K = ZZ^T = U \Sigma V^T V \Sigma^T U^T$

$= U \Sigma \Sigma^T U^T$

$\Sigma \Sigma^T = \Lambda$ ← eigenvalues of K

↑ singular values of Z

The idea: use

$\hat{z}_i := (\sigma_1 u_{i1}, \sigma_2 u_{i2}, \ldots, \sigma_k u_{ik})$ $\quad \sigma_1 \geq \sigma_2 \ldots$ $\quad \Sigma_{ii} = \sqrt{\Lambda_{ii}}$

as a low-dimensional representation of $z_i = \Phi(x_i)$.

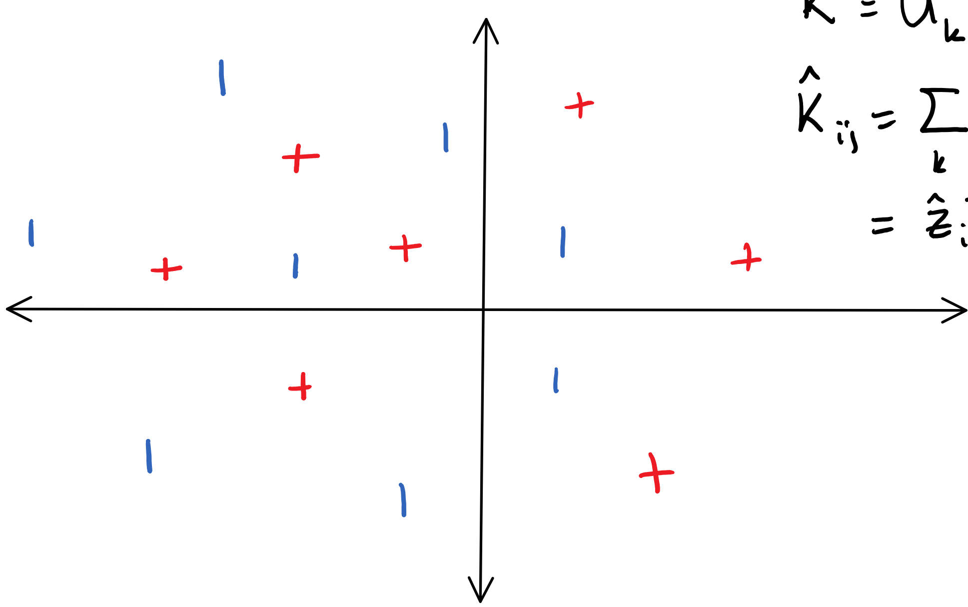
Distortion / Compression:

$\hat{K} = U_k \Lambda_k U_k^T$

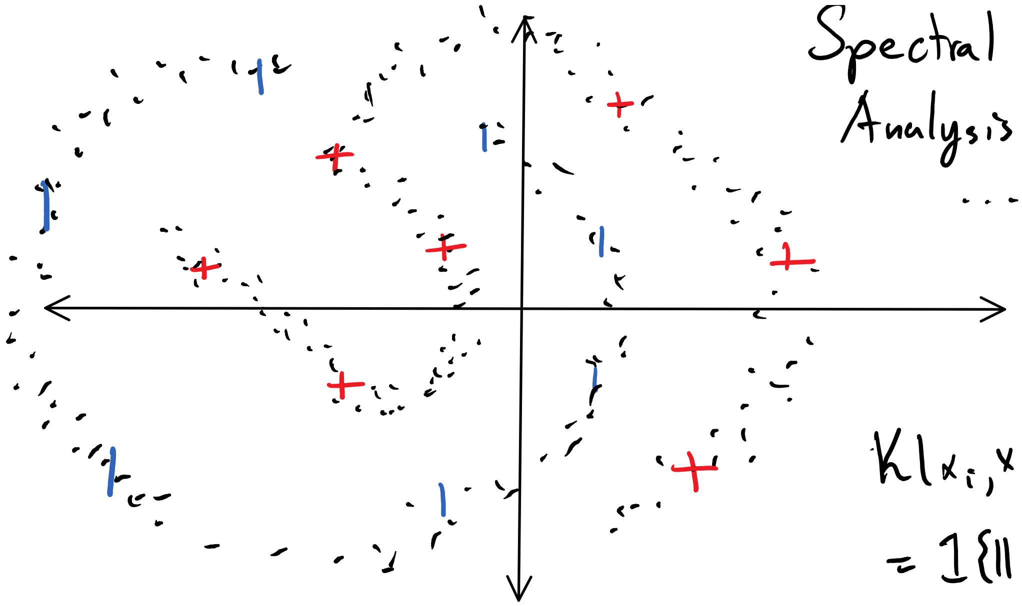
$\hat{K}_{ij} = \sum_k \lambda_k u_{ki} u_{kj}$

$= \hat{z}_i^T \hat{z}_j$

Semisupervised learning



Spectral Connectivity



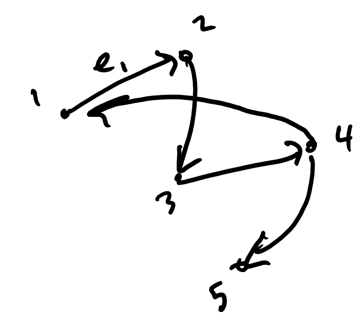
Spectral Connectivity Analysis Ann Lee, ...

$$K(x_i, x_j) = \mathbb{1}\{\|x_i - x_j\|_2 \le \tau\}$$

Unlabelled data can help!

Laplacian Eigenmaps

Graph is vertices $V = \{1, \dots, N\}$, $E = \{(e_1^+, e_1^-), (e_2^+, e_2^-), \dots, (e_m^+, e_m^-)\}$.



$e_1^- = 1 \quad e_1^+ = 2$

Adjacency matrix:

$$W_{ij} = 1 \quad \text{if } (i,j) \in E \text{ or } (j,i) \in E$$
$$= 0 \quad \text{otherwise}$$

(in general, can assign weights)

Degree matrix: diagonal $D_{ii} = \sum_{j=1}^{N} W_{ij}$

$d_i := D_{ii} = w_{e_1} + w_{e_2} + w_{e_3}$

Combinatorial Laplacian: $D - W = L$

chain graph $N=8$ $m=7$

degree: 1   deg: 2

$W_{12} = 1$ $W_{21} = 1,\ W_{23} = 1$

$$\begin{bmatrix} 1 & -1 & 0 & \cdots & & & 0 \\ -1 & 2 & -1 & 0 & \cdots & & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ & & & \ddots & & & \\ 0 & \cdots & & & 0 & -1 & 1 \end{bmatrix}$$

$x \in \mathbb{R}^N$

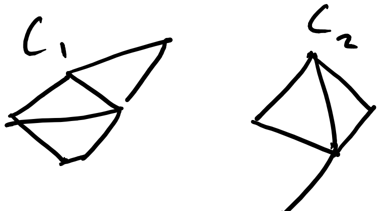
$$(Lx)_i = d_i x_i - (Wx)_i = d_i x_i - \sum_{j=1}^{N} W_{ij} x_j$$
$$= \left(\sum_{j=1}^{N} W_{ij}\right) x_i - \sum_{j=1}^{N} W_{ij} x_j$$
$$= \sum_{j=1}^{N} W_{ij}(x_i - x_j)$$

$$(L\mathbb{1})_i = \sum_{j=1}^{N} W_{ij}(1-1) = 0$$

$$x^T L x = \sum_i (Lx)_i x_i = \sum_{i=1}^{N}\left(\sum_{j=1}^{N} W_{ij}(x_i - x_j)\right) x_i$$
$$\boxed{= \frac{1}{2}\sum_{ij} W_{ij}(x_i - x_j)^2} = \frac{1}{2}\sum_{ij} W_{ij}(x_i^2 - 2x_i x_j + x_j^2)$$
$$= \frac{1}{2} 2\left(\sum_{ij} W_{ij}(x_i^2 - x_i x_j)\right)$$

$$x^T L x \geq 0$$

C_1 C_2



Graph with 2 connected components

$C_j \subseteq V$

$$\mathbb{1}_{C_j} := (\underbrace{1, 1, \ldots, 1}_{C_j}, 0, \ldots, 0)$$

$$(L\mathbb{1}_{C_j})_i = \sum_k W_{ik}(\mathbb{1}_{C_j i} - \mathbb{1}_{C_j k})$$

$$(L\mathbb{1}_{c_j})_i = \sum_k W_{ik}\,(\mathbb{1}_{c_j i} - \mathbb{1}_{c_j k})$$

$$= 0$$

<u>prop</u> L is p.s.d. and $\{\mathbb{1}_{c_j}\}_{j=1}^{K}$ spans the null space of $L \equiv \mathbb{1}_{c_j}$ are eigenvectors with eigenvalue 0,

<u>Idea</u>: If we want to partition a graph then look at eigenvectors with low eigenvalues.

<u>Laplacian eigenmaps</u>

Construct a graph $W_{ij} = K(x_i, x_j)$

Construct a Laplacian matrix

(comb) $L = D - W$

(norm) $L = I - D^{-1/2} W D^{-1/2}$

Spectrum of L : $((\lambda_1, u_1), \ldots, (\lambda_k, u_k))$

s.t. $\lambda_1 = 0 \le \ldots \le \lambda_k$

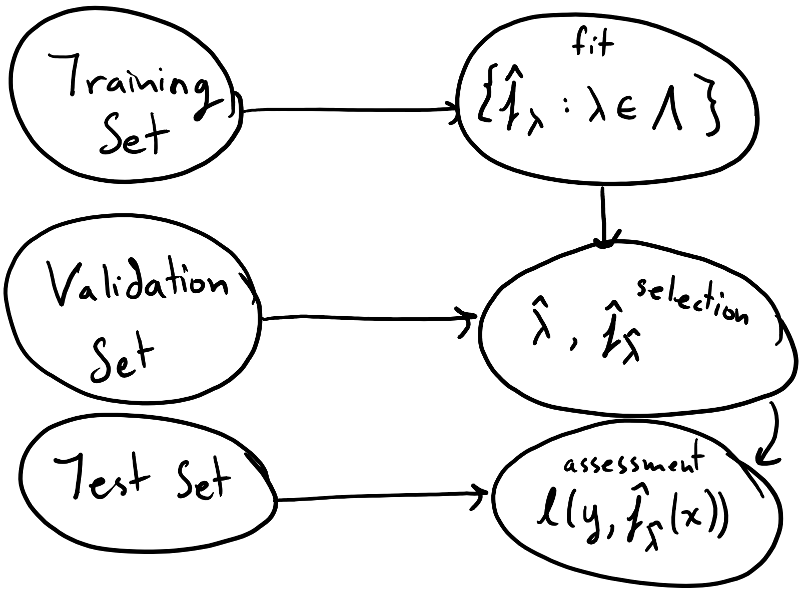
Embed: $x_i \longrightarrow \Phi(x_i) = (\sqrt{\lambda_1}\,u_{1i}, \sqrt{\lambda_2}\,u_{2i}, \ldots, \sqrt{\lambda_k}\,u_{ki})$

Cross Validation

Monday, May 15, 2017 10:55 PM

Model selection: choose tuning parameter, $\hat{\lambda}$, to minimize $R(\lambda) = \mathbb{E}\, \ell(y, \hat{f}_\lambda(x))$

Model assessment: honest assessment of loss $\ell(y, \hat{f}_{\hat{\lambda}}(x))$ - ROC, PR



K-fold CV

$D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\quad D = \left.\begin{array}{c} x_1 \;\; y_1 \\ \vdots \\ x_{\frac{N}{K}} , y_{\frac{N}{K}} \end{array}\right\}$ 1st fold $\quad \vdots$

Create K-folds: $F_i = \{(x_{\frac{N}{K}(i-1)+1}, y_{\ldots}), \ldots, (x_{\frac{N}{K}(i-1)+\frac{N}{K}}, y_{\ldots})\}$

Predict on F_i and fit on $D - F_i$ for all $i = 1, \ldots, K$

$\hookrightarrow$ Risk on F_i: R^i_{cv}

Output $\frac{1}{K}\sum_{i=1}^{K} R^i_{cv} \longrightarrow$ use for $\hat{\lambda}$ selection

True risk v. Empirical risk

$$R_n(\hat{f}) = \frac{1}{n}\sum_i (y_i - \hat{f}(x_i))^2$$

$$R_n'(\hat{f}) = \frac{1}{n}\sum_i (y_i' - \hat{f}(x_i))^2$$

iid copy

$\mathbb{E} R_n'(\hat{f}) = R(\hat{f})$ $\quad (\underbrace{y_i' - y_i} + y_i - \hat{f}(x_i))^2$

Derive: $R(\hat{f}) = \mathbb{E} R_n(\hat{f}) + \frac{2}{n} \sum_i \mathrm{Cov}(\hat{y}_i, y_i)$

Linear fit, additive error $Y = f(X) + \varepsilon$

$\varepsilon \sim N(0, \sigma_\varepsilon^2)$

$$\sum_{i=1}^{n} \mathrm{Cov}(\hat{y}_i, y_i) = p\sigma_\varepsilon^2$$

$$\frac{1}{2\sigma_\varepsilon^2}(y_i - \hat{f}(x_i))^2$$

$$R(\hat{f}) = \mathbb{E}R_n(\hat{f}) + 2 \cdot \frac{p}{n}\sigma_\varepsilon^2 \qquad \frac{1}{n}\sum_i \frac{1}{\sigma_\varepsilon^2}(y_i - \hat{f}(x_i))^2$$

AIC: $-\frac{2}{n}\sum_i \log L + 2\frac{p}{n}$

<u>Effective # of Parameters</u>

$\hat{y} = Sy$ then $df(S) = \mathrm{tr}(S)$

<u>derive</u>

$$\hat{y}_i \cdot y_i = (S_i^T y) y_i \qquad \mathbb{E}\, \hat{y}_i y_i = S_{ii}\, \mathbb{E}\, y_i^2$$

y_i were mean 0 (center y_i)

$$\sum_i \operatorname{Cov}(\hat{y}_i, y_i) = \sum_i S_{ii}\, \mathbb{V}(y_i \mid x_i) = \sigma_\varepsilon^2 \cdot \operatorname{tr}(S)$$

Generalized CV

$$\tilde{f}^{i}(x_i) = \begin{bmatrix} S_{:,i-1} & S_{i+1:} \end{bmatrix} y = S_i^T y - S_{ii} y_i = \hat{f}(x_i) - S_{ii} y_i$$