Advanced Data Analysis: More on Kernels

Masashi Sugiyama (Computer Science)

W8E-406, sugi@cs.titech.ac.jp

http://sugiyama-www.cs.titech.ac.jp/~sugi

Kernel Trick

- For feature transformation $\boldsymbol{\phi}(\boldsymbol{x})$ $(= \boldsymbol{f})$, there exists a bivariate function $K(\boldsymbol{x}, \boldsymbol{x}')$ such that

$$K_{i,j} = \langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

  if $\boldsymbol{K}$ is symmetric and positive semi-definite:

$$\boldsymbol{K}^\top = \boldsymbol{K} \qquad \forall \boldsymbol{y}, \ \ \langle \boldsymbol{K}\boldsymbol{y}, \boldsymbol{y} \rangle \geq 0$$

- Such $K(\boldsymbol{x}, \boldsymbol{x}')$ is called the reproducing kernel.
- Rather than directly specifying $\boldsymbol{\phi}(\boldsymbol{x})$, we implicitly specify $\boldsymbol{\phi}(\boldsymbol{x})$ by a reproducing kernel.

Combination of Reproducing Kernels

For any reproducing kernels (RKs)

$$K^{(1)}(\boldsymbol{x},\boldsymbol{x}') = \langle \phi^{(1)}(\boldsymbol{x}), \phi^{(1)}(\boldsymbol{x}') \rangle$$

$$K^{(2)}(\boldsymbol{x},\boldsymbol{x}') = \langle \phi^{(2)}(\boldsymbol{x}), \phi^{(2)}(\boldsymbol{x}') \rangle$$

- Positive scaling of RK is still RK

$$K(\boldsymbol{x},\boldsymbol{x}') = \alpha K^{(1)}(\boldsymbol{x},\boldsymbol{x}') \qquad \alpha > 0$$

- Sum of RKs is still RK:

$$K(\boldsymbol{x},\boldsymbol{x}') = K^{(1)}(\boldsymbol{x},\boldsymbol{x}') + K^{(2)}(\boldsymbol{x},\boldsymbol{x}')$$

- Product of RKs is still RK:

$$K(\boldsymbol{x},\boldsymbol{x}') = K^{(1)}(\boldsymbol{x},\boldsymbol{x}')K^{(2)}(\boldsymbol{x},\boldsymbol{x}')$$

Proof

We prove that there exists a feature map $\phi(\boldsymbol{x})$ such that $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle = K(\boldsymbol{x}, \boldsymbol{x}')$.

- For $\boldsymbol{\phi}(\boldsymbol{x}) = \sqrt{\alpha}\boldsymbol{\phi}^{(1)}(\boldsymbol{x})$,

$$\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle = \alpha \langle \boldsymbol{\phi}^{(1)}(\boldsymbol{x}), \boldsymbol{\phi}^{(1)}(\boldsymbol{x}') \rangle = \alpha K^{(1)}(\boldsymbol{x}, \boldsymbol{x}')$$

- For $\boldsymbol{\phi}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{\phi}^{(1)}(\boldsymbol{x}) \\ \boldsymbol{\phi}^{(2)}(\boldsymbol{x}) \end{pmatrix}$,

$$\begin{aligned} \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle &= \langle \boldsymbol{\phi}^{(1)}(\boldsymbol{x}), \boldsymbol{\phi}^{(1)}(\boldsymbol{x}') \rangle + \langle \boldsymbol{\phi}^{(2)}(\boldsymbol{x}), \boldsymbol{\phi}^{(2)}(\boldsymbol{x}') \rangle \\ &= K^{(1)}(\boldsymbol{x}, \boldsymbol{x}') + K^{(2)}(\boldsymbol{x}, \boldsymbol{x}') \end{aligned}$$

- For $[\boldsymbol{\phi}(\boldsymbol{x})]_{i,j} = [\boldsymbol{\phi}^{(1)}(\boldsymbol{x})]_i [\boldsymbol{\phi}^{(2)}(\boldsymbol{x})]_j$,

$$\begin{aligned} \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle &= \sum_{i,j} [\boldsymbol{\phi}^{(1)}(\boldsymbol{x})]_i [\boldsymbol{\phi}^{(2)}(\boldsymbol{x})]_j [\boldsymbol{\phi}^{(1)}(\boldsymbol{x}')]_i [\boldsymbol{\phi}^{(2)}(\boldsymbol{x}')]_j \\ &= \langle \boldsymbol{\phi}^{(1)}(\boldsymbol{x}), \boldsymbol{\phi}^{(1)}(\boldsymbol{x}') \rangle \langle \boldsymbol{\phi}^{(2)}(\boldsymbol{x}), \boldsymbol{\phi}^{(2)}(\boldsymbol{x}') \rangle \\ &= K^{(1)}(\boldsymbol{x}, \boldsymbol{x}') \, K^{(2)}(\boldsymbol{x}, \boldsymbol{x}') \end{aligned}$$

Exercise: Playing with Kernel Trick

- Norm:

$$\|\boldsymbol{f}\| = \sqrt{K(\boldsymbol{x}, \boldsymbol{x})}$$

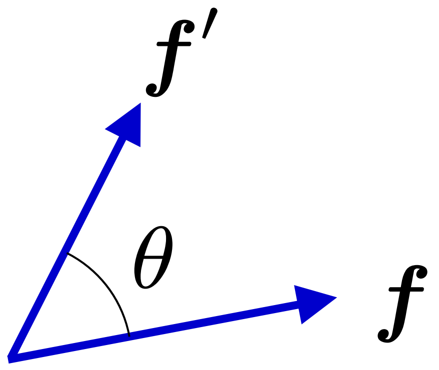
- Distance:

$$\|\boldsymbol{f} - \boldsymbol{f}'\|^2 = K(\boldsymbol{x}, \boldsymbol{x}) - 2K(\boldsymbol{x}, \boldsymbol{x}') + K(\boldsymbol{x}', \boldsymbol{x}')$$

- Angle:

$$\cos\theta = \frac{K(\boldsymbol{x}, \boldsymbol{x}')}{\sqrt{K(\boldsymbol{x}, \boldsymbol{x})K(\boldsymbol{x}', \boldsymbol{x}')}}$$

$$\langle \boldsymbol{f}, \boldsymbol{f}' \rangle = \|\boldsymbol{f}\|\|\boldsymbol{f}'\| \cos\theta$$

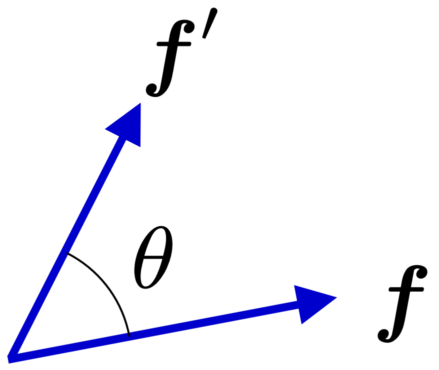


Playing with Kernel Trick (cont.)

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c^2\right)$$

$$c > 0$$

- For Gaussian kernels,
  - $\|\boldsymbol{f}\|^2 = 1$
  - $\|\boldsymbol{f} - \boldsymbol{f}'\|^2 = 2 - 2K(\boldsymbol{x}, \boldsymbol{x}')$
  - $\cos\theta = K(\boldsymbol{x}, \boldsymbol{x}')$



Kernel Trick Revisited

$$\langle \boldsymbol{f}, \boldsymbol{f}' \rangle = K(\boldsymbol{x}, \boldsymbol{x}')$$

- An inner product in the feature space can be efficiently computed by the kernel function.
- If a linear algorithm is expressed only in terms of the inner product, it can be non-linearlized by the kernel trick:
  - PCA, LPP, FDA, LFDA
  - K-means clustering
  - Perceptron (support vector machine)

Kernel LPP

- Kernel LPP embedding of sample $\boldsymbol{f}$ $(= \phi(\boldsymbol{x}))$:

$$\boldsymbol{g} = \boldsymbol{A}^\top \boldsymbol{k}$$

$$\boldsymbol{k} = (K(\boldsymbol{x}, \boldsymbol{x}_1), K(\boldsymbol{x}, \boldsymbol{x}_2), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_n))^\top$$

$$\boldsymbol{A} = (\boldsymbol{\alpha}_{n-m+1} | \boldsymbol{\alpha}_{n-m+2} | \cdots | \boldsymbol{\alpha}_n)$$

  - $\{\lambda_i, \boldsymbol{\alpha}_i\}_{i=1}^m$:Sorted generalized eigenvalues and normalized eigenvectors of $\boldsymbol{K}\boldsymbol{L}\boldsymbol{K}\boldsymbol{\alpha} = \lambda \boldsymbol{K}\boldsymbol{D}\boldsymbol{K}\boldsymbol{\alpha}$

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \qquad \langle \boldsymbol{K}\boldsymbol{D}\boldsymbol{K}\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j \rangle = \delta_{i,j}$$

$$K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \qquad \boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \qquad \boldsymbol{D} = \mathrm{diag}\left(\sum_{j=1}^n \boldsymbol{W}_{i,j}\right)$$

- Note: When $\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}$ is not full-rank, it may be replaced with $\boldsymbol{K}\boldsymbol{D}\boldsymbol{K} + \varepsilon \boldsymbol{I}_n$.

  ε :small positive scalar

Kernel LPP Embedding of Given Features

- Kernel LPP embedding of $\{\boldsymbol{f}_i\}_{i=1}^n$:

$$\boldsymbol{G} = \boldsymbol{A}^\top \boldsymbol{K} \qquad \boldsymbol{G} = (\boldsymbol{g}_1 | \boldsymbol{g}_2 | \cdots | \boldsymbol{g}_n)$$

- $\boldsymbol{G}$ can be directly obtained as

$$\boldsymbol{G} = \boldsymbol{\Psi}^\top \quad \boldsymbol{\Psi} = (\boldsymbol{\psi}_{n-m+1} | \boldsymbol{\psi}_{n-m+2} | \cdots | \boldsymbol{\psi}_n)$$

  - $\{\gamma_i, \boldsymbol{\psi}_i\}_{i=1}^n$:Sorted eigenvalues and normalized eigenvectors of $\boldsymbol{L}\boldsymbol{\psi} = \gamma \boldsymbol{D}\boldsymbol{\psi}$

$$\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_n \qquad \langle \boldsymbol{D}\boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle = \delta_{i,j}$$

- Note: When similarity matrix $\boldsymbol{W}$ is sparse, $\boldsymbol{L}$ and $\boldsymbol{D}$ are also sparse. Sparse eigenproblems can be solved efficiently.

Laplacian Eigenmap

$$\boldsymbol{L}\boldsymbol{\psi} = \gamma \boldsymbol{D}\boldsymbol{\psi}$$

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$$

$$\boldsymbol{D} = \mathrm{diag}\left(\sum_{j=1}^{n} \boldsymbol{W}_{i,j}\right)$$

- Definition of $\boldsymbol{L}$ implies $\boldsymbol{L}\mathbf{1} = \mathbf{0}$.

$\Rightarrow$ $\gamma_n = 0, \quad \boldsymbol{\psi}_n \propto \mathbf{1}$

- In practice, we remove $\boldsymbol{\psi}_n$ and use

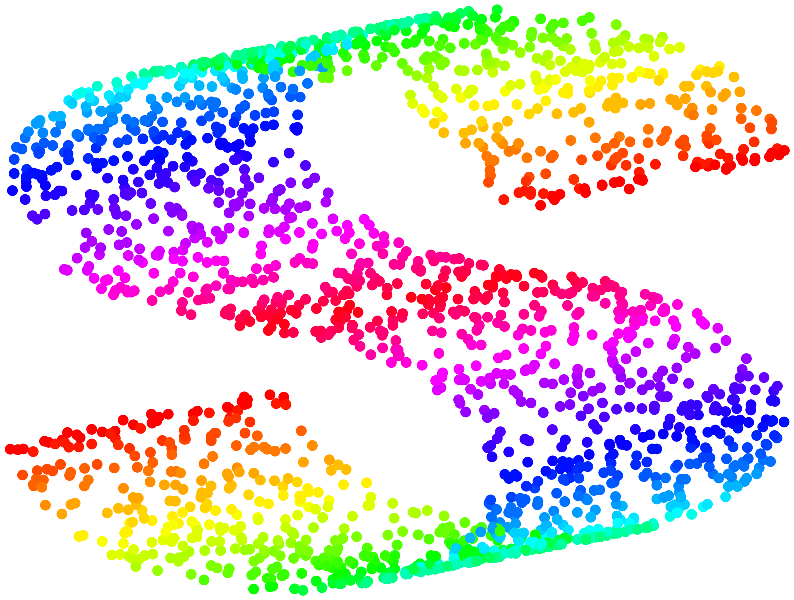
$$\boldsymbol{G} = (\boldsymbol{\psi}_{n-m} | \boldsymbol{\psi}_{n-m+1} | \cdots | \boldsymbol{\psi}_{n-1})^\top$$

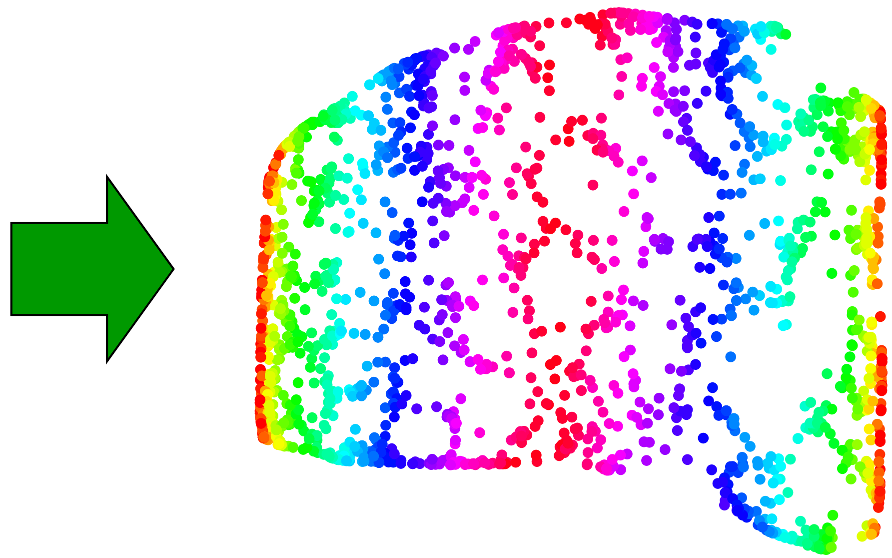
- This non-linear embedding method is called Laplacian eigenmap.

Example

Original data (3D)

Embedded Data (2D)





Note: Similarity matrix is defined by the nearest-neighbor-based method with 10 nearest neighbors.

- Laplacian eigenmap can successfully unfold the non-linear manifold.

Homework

1. Implement Laplacian eigenmap and unfold the 3-dimensional S-curve data.

http://sugiyama-www.cs.titech.ac.jp/~sugi/data/DataAnalysis

   Test Laplacian eigenmap with your own (artificial or real) data and analyze its characteristics.

Homework (cont.)

2. Prove that the dual eigenvalue problem of Fisher discriminant analysis is given by

$$\boldsymbol{K}\boldsymbol{L}^{(b)}\boldsymbol{K}\boldsymbol{\alpha} = \lambda \boldsymbol{K}\boldsymbol{L}^{(w)}\boldsymbol{K}\boldsymbol{\alpha}$$

$$\boldsymbol{L}^{(b)} = \boldsymbol{D}^{(b)} - \boldsymbol{W}^{(b)}$$

$$\boldsymbol{D}^{(b)} = \mathrm{diag}(\textstyle\sum_{j=1}^{n} \boldsymbol{W}_{i,j}^{(b)})$$

$$\boldsymbol{W}_{i,j}^{(b)} = \begin{cases} 1/n - 1/n_\ell & (y_i = y_j = \ell) \\ 1/n & (y_i \neq y_j) \end{cases}$$

$$\boldsymbol{L}^{(w)} = \boldsymbol{D}^{(w)} - \boldsymbol{W}^{(w)}$$

$$\boldsymbol{D}^{(w)} = \mathrm{diag}(\textstyle\sum_{j=1}^{n} \boldsymbol{W}_{i,j}^{(w)})$$

$$\boldsymbol{W}_{i,j}^{(w)} = \begin{cases} 1/n_\ell & (y_i = y_j = \ell) \\ 0 & (y_i \neq y_j) \end{cases}$$

Note: When solving the above eigenproblem, we may practically need to regularize it as

$$\boldsymbol{K}\boldsymbol{L}^{(b)}\boldsymbol{K}\boldsymbol{\alpha} = \lambda(\boldsymbol{K}\boldsymbol{L}^{(w)}\boldsymbol{K} + \epsilon \boldsymbol{I}_n)\boldsymbol{\alpha}$$

- LFDA can also be kernelized similarly!

Notification of Final Assignment

- Data Analysis: Apply dimensionality reduction or clustering techniques to your own data set and “mine” something interesting!

- Deadline: July 31st (Wed) 17:00
  - Bring your printed report to W8E-406.
  - E-mail submission is also possible (though not recommended).

Mini-Conference on Data Analysis

- On July 16th and 23rd, we have a mini-conference on data analysis.
- Some of the students may present their data analysis results.
- Those who give a talk at the conference will have very good grades!

Mini-Conference on Data Analysis

- Application procedure: On June 25th, just say to me “I want to give a talk!”.
- Presentation: approx. 10 min (?)
  - Description of your data
  - Methods to be used
  - Outcome
- Slides should be in English.
- Better to speak in English, but Japanese may also be allowed (perhaps your friends will provide simultaneous translation!).