Lecture 6
Kernel Machines – Part 2

(February 13, 2015)

Mu Zhu
University of Waterloo

The Kernel "Trick"

- separating hyperplane "merely" a linear classifier ... big deal
- ignited wide recognition of the "kernel trick"
  - statisticians like Grace Wahba ... long before the SVM
  - reproducing kernel Hilbert space, $\mathcal{H}_K$
- both training and prediction stages depend on $\boldsymbol{x}$'s only through their inner products, e.g., $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$ and $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}$
- can replace with $K(\boldsymbol{x}_i;\boldsymbol{x}_j)$ and $K(\boldsymbol{x};\boldsymbol{x}_i)$, which amounts to using a "different kind" of inner product

The Kernel "Trick"

SVM Training

$$\widehat{\boldsymbol{\beta}} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i \boldsymbol{x}_i, \quad \text{where } \widehat{\alpha}_i\text{'s obtained from}$$

$$\max_{\alpha_i} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \quad [\text{swap } K(\boldsymbol{x}_i; \boldsymbol{x}_j)]$$

$$\text{s.t} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq \gamma$$

SVM Prediction

$$f(\boldsymbol{x}) = \widehat{\beta}_0 + \widehat{\boldsymbol{\beta}}^{\mathrm{T}} \boldsymbol{x} = \widehat{\beta}_0 + \sum_{i=1}^{n} \widehat{\alpha}_i y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x} \quad [\text{swap } K(\boldsymbol{x}; \boldsymbol{x}_i)]$$

The Kernel "Trick"

- as long as $K(\cdot;\cdot)$ is positive semi-definite, $\exists\ \varphi(\cdot)$ such that

$$K(\boldsymbol{u};\boldsymbol{v}) = [\varphi(\boldsymbol{u})]^{\mathrm{T}}[\varphi(\boldsymbol{v})],$$

  a result due to Mercer dating back to 1909

- so get decision boundary linear in the space of $\varphi(\boldsymbol{x})$, but nonlinear in original space

- need not know the space of $\varphi(\boldsymbol{x})$ explicitly ... sometimes called the implicit feature space

- statisticians are used to explicitly adding higher order terms

Example in $\mathbb{R}^2$

$$\varphi(\boldsymbol{u}) = \left[\begin{array}{cccccc} 1 & \sqrt{2}u_1 & \sqrt{2}u_2 & u_1^2 & u_2^2 & \sqrt{2}u_1u_2 \end{array} \right]$$

$$\begin{aligned} K(\boldsymbol{u};\boldsymbol{v}) &= [\varphi(\boldsymbol{u})]^{\mathrm{T}}[\varphi(\boldsymbol{v})] \\ &= 1 + 2u_1v_1 + 2u_2v_2 + u_1^2v_1^2 + u_2^2v_2^2 + 2(u_1u_2)(v_1v_2) \\ &= (1 + u_1v_1 + u_2v_2)^2 \\ &= (1 + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v})^2 \end{aligned}$$

The Radial Basis Kernel in $\mathbb{R}$

$$\phi(u) = \left[\begin{array}{ccccc} 1 & u & \frac{u^2}{\sqrt{2!}} & \frac{u^3}{\sqrt{3!}} & \cdots \end{array} \right], \quad \varphi(u) = \frac{\phi(u)}{\|\phi(u)\|}$$

$$K(u;v) = [\varphi(u)]^{\mathrm{T}}[\varphi(v)]$$

$$= \frac{1 + uv + \frac{u^2v^2}{2!} + \frac{u^3v^3}{3!} + ...}{\sqrt{\left(1 + u^2 + \frac{u^4}{2!} + \frac{u^6}{3!} + ...\right)}\sqrt{\left(1 + v^2 + \frac{v^4}{2!} + \frac{v^6}{3!} + ...\right)}}$$

$$= \frac{e^{uv}}{e^{u^2/2}e^{v^2/2}} = \exp\left[-\frac{u^2 - 2uv + v^2}{2}\right]$$

$$= \exp\left[-\frac{(u-v)^2}{2}\right]$$

Reproducing Kernel Hilbert Space

- notion of inner product $\sim$ "soul" of Hilbert spaces

- $\mathcal{H}_K \sim$ a "nice" functional (Hilbert) space associated with a kernel $K(\boldsymbol{u};\boldsymbol{v})$ having the so-called "reproducing property":

$$\langle f(\boldsymbol{x}),\ \underbrace{K(\boldsymbol{x};\boldsymbol{z})}_{\text{think } g(\boldsymbol{x})}\ \rangle|_{\mathcal{H}_K} = f(\boldsymbol{z})$$

- quite abstract ... a bit "easier" to appreciate if one imagines eigen-functions of $K(\boldsymbol{u};\boldsymbol{v})$ forming a basis of $\mathcal{H}_K$, so

$$K(\boldsymbol{x};\boldsymbol{z}) = \sum_{\ell} \lambda_\ell \phi_\ell(\boldsymbol{x})\phi_\ell(\boldsymbol{z}) \quad \text{and} \quad f(\boldsymbol{x}) = \sum_{\ell} c_\ell \phi_\ell(\boldsymbol{x})$$

Representer Theorem

Theorem Consider the regularized functional estimation problem in $\mathcal{H}_K$,

$$\min_{f \in \mathcal{H}_K} \quad \sum_{i=1}^{n} L[y_i, f(\boldsymbol{x}_i)] + \lambda \|f(\boldsymbol{x})\|^2_{\mathcal{H}_K}.$$

The minimizing solution, $f(\boldsymbol{x})$, must have the form,

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}; \boldsymbol{x}_i).$$

Inner Products

$$f(\boldsymbol{x}) = \sum_i a_i K(\boldsymbol{x}; \boldsymbol{x}_i), \quad g(\boldsymbol{x}) = \sum_j b_j K(\boldsymbol{x}; \boldsymbol{x}_j)$$

$$\langle f(\boldsymbol{x}), g(\boldsymbol{x}) \rangle|_{\mathcal{H}_K} = \left\langle \underbrace{\sum_i a_i K(\boldsymbol{x}; \boldsymbol{x}_i)}_{f(\boldsymbol{x})}, g(\boldsymbol{x}) \right\rangle \Bigg|_{\mathcal{H}_K}$$

$$= \sum_i a_i \underbrace{\langle K(\boldsymbol{x}; \boldsymbol{x}_i), g(\boldsymbol{x}) \rangle|_{\mathcal{H}_K}}_{g(\boldsymbol{x}_i)} = \sum_i a_i \underbrace{\left[\sum_j b_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \right]}_{g(\boldsymbol{x}_i)}$$

$$= \sum_i \sum_j a_i b_j K(\boldsymbol{x}_i; \boldsymbol{x}_j)$$

Regularized Functional Estimation

$$\min_{f \in \mathcal{H}_K} \quad \sum_{i=1}^{n} L[y_i, f(\boldsymbol{x}_i)] + \lambda \|f(\boldsymbol{x})\|^2_{\mathcal{H}_K},$$

$$\min_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} L\left[y_i, \sum_{j=1}^{n} \alpha_j K(\boldsymbol{x}_i; \boldsymbol{x}_j)\right] + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(\boldsymbol{x}_i; \boldsymbol{x}_j)$$

$$\min_{\boldsymbol{\alpha}} \quad L\left(\boldsymbol{y}, \boldsymbol{K}\boldsymbol{\alpha}\right) + \lambda \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\alpha}$$

SVM does exactly this, with "hinge" loss

The Gram Matrix

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 & \ldots & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_n \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_2 & \ldots & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_n \end{bmatrix}$$

Kernelization

$$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 & \dots & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_n \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_2 & \dots & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_2 & \dots & \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{x}_n \end{bmatrix}$$

$\Downarrow$ replace $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$ with $K(\boldsymbol{x}_i;\boldsymbol{x}_j)$ $\Downarrow$

$$\boldsymbol{K} = \begin{bmatrix} K(\boldsymbol{x}_1;\boldsymbol{x}_1) & K(\boldsymbol{x}_1;\boldsymbol{x}_2) & \dots & K(\boldsymbol{x}_1;\boldsymbol{x}_n) \\ K(\boldsymbol{x}_2;\boldsymbol{x}_1) & K(\boldsymbol{x}_2;\boldsymbol{x}_2) & \dots & K(\boldsymbol{x}_2;\boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n;\boldsymbol{x}_1) & K(\boldsymbol{x}_n;\boldsymbol{x}_2) & \dots & K(\boldsymbol{x}_n;\boldsymbol{x}_n) \end{bmatrix}$$

Ex I: Ridge Regression

Training

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \quad \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda \underbrace{\|\boldsymbol{\beta}\|^2}_{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}}$$

$\Downarrow$ reparameterize $\boldsymbol{\beta}$ as $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}$ $\Downarrow$

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad \|\boldsymbol{y}-\boldsymbol{X}\underbrace{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}}_{\boldsymbol{\beta}}\|^2 + \lambda \underbrace{\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{X}}_{\boldsymbol{\beta}^{\mathrm{T}}}\underbrace{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}}_{\boldsymbol{\beta}}$$

$\Downarrow$ replace Gram matrix $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ with $\boldsymbol{K}$ $\Downarrow$

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{\boldsymbol{n}}} \quad \|\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\alpha}\|^2 + \lambda\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\alpha}$$

Ex I: Ridge Regression

Prediction

$$\widehat{y}_{new} = \boldsymbol{x}_{new}^{\mathrm{T}}\widehat{\boldsymbol{\beta}} = \boldsymbol{x}_{new}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\widehat{\boldsymbol{\alpha}}$$

$$= \boldsymbol{x}_{new}^{\mathrm{T}}\left[\begin{array}{cccc}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n\end{array}\right]\widehat{\boldsymbol{\alpha}} = \sum_{i=1}^{n}\left(\boldsymbol{x}_{new}^{\mathrm{T}}\boldsymbol{x}_i\right)\widehat{\alpha}_i$$

$$= \sum_{i=1}^{n}\widehat{\alpha}_i K\left(\boldsymbol{x}_{new};\boldsymbol{x}_i\right)$$

Remark Though it appears to be optional, the ridge penalty term $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\alpha}$ is necessary because, once kernelized, the regression will invariably overfit without it.

Ex II: Canonical Correlation Analysis

- for random variables $x, y \in \mathbb{R}$, their correlation coefficient is

$$\rho = \mathbb{C}\text{orr}(x, y) = \frac{\mathbb{C}\text{ov}(x, y)}{\sqrt{\mathbb{V}\text{ar}(x)\mathbb{V}\text{ar}(y)}}$$

- for $\boldsymbol{x} \in \mathbb{R}^p$ and $\boldsymbol{y} \in \mathbb{R}^q$,

$$\rho(\boldsymbol{x}, \boldsymbol{y}) \equiv \max_{\boldsymbol{u} \in \mathbb{R}^p, \boldsymbol{v} \in \mathbb{R}^q} \mathbb{C}\text{orr}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}, \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y})$$

  is called the canonical correlation coefficient (Hotelling, 1936; *Biometrika*)

- classic ... many (almost all) techniques in a standard textbook on "multivariate analysis" can be reduced/converted to canonical correlation analysis (CCA)

Ex II: Canonical Correlation Analysis

- since

$$\begin{aligned}\mathbb{C}\text{orr}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}, \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}) &= \frac{\mathbb{C}\text{ov}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}, \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y})}{\sqrt{\mathbb{V}\text{ar}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x})\mathbb{V}\text{ar}(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y})}} \\ &= \frac{\boldsymbol{u}^{\mathrm{T}}\mathbb{C}\text{ov}(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{v}}{\sqrt{[\boldsymbol{u}^{\mathrm{T}}\mathbb{V}\text{ar}(\boldsymbol{x})\boldsymbol{u}][\boldsymbol{v}^{\mathrm{T}}\mathbb{V}\text{ar}(\boldsymbol{y})\boldsymbol{v}]}},\end{aligned}$$

- maximization problem for computing $\rho(\boldsymbol{x}, \boldsymbol{y})$ equivalent to

$$\begin{aligned}\max_{\boldsymbol{u}\in\mathbb{R}^p, \boldsymbol{v}\in\mathbb{R}^q} \quad & \boldsymbol{u}^{\mathrm{T}}[\mathbb{C}\text{ov}(\boldsymbol{x}, \boldsymbol{y})]\boldsymbol{v}, \\ \text{subject to} \quad & \boldsymbol{u}^{\mathrm{T}}[\mathbb{V}\text{ar}(\boldsymbol{x})]\boldsymbol{u} = 1 \quad \text{and} \\ & \boldsymbol{v}^{\mathrm{T}}[\mathbb{V}\text{ar}(\boldsymbol{y})]\boldsymbol{v} = 1\end{aligned}$$

Ex II: Canonical Correlation Analysis

If both data matrices

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix}_{n \times p} \quad \text{and} \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{y}_n^{\mathrm{T}} \end{bmatrix}_{n \times q}$$

are column centered, then, the "usual" empirical estimates are

$$\widehat{\mathbb{C}\mathrm{ov}(\boldsymbol{x}, \boldsymbol{y})} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{y}_i^{\mathrm{T}} = \frac{1}{n} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{Y},$$

$$\widehat{\mathbb{V}\mathrm{ar}(\boldsymbol{x})} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} = \frac{1}{n} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}, \quad \widehat{\mathbb{V}\mathrm{ar}(\boldsymbol{y})} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^{\mathrm{T}} = \frac{1}{n} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{Y}.$$

Ex II: Canonical Correlation Analysis

- thus, empirical version of the problem is

$$\widehat{\rho}(\boldsymbol{x}, \boldsymbol{y}) \quad = \quad \max_{\substack{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{u}=1 \\ \boldsymbol{v}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v}=1}} \boldsymbol{u}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v},$$

  a generalized eigenvalue problem

- compare with principal component analysis (PCA)

$$\underbrace{\max_{\|\boldsymbol{u}\|=1} \mathbb{V}\mathrm{ar}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x})}_{\text{population version}} \quad \text{or} \quad \underbrace{\max_{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}=1} \boldsymbol{u}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{u}}_{\text{sample version}},$$

  a "regular" eigenvalue problem

Ex II: Canonical Correlation Analysis

- eigenvalue problems:

$$\max_{\|\boldsymbol{a}\|=1} \boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{a} \qquad \text{for symmetric } \boldsymbol{S}$$

$$\max_{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{a}=1} \boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{a} \qquad \text{for symmetric } \boldsymbol{S}, \boldsymbol{M}$$

$$\max_{\substack{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{a}=1 \\ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{N}\boldsymbol{b}=1}} \boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{b} \qquad \text{for symmetric } \boldsymbol{M}, \boldsymbol{N}$$

- e.g., PCA (top), Fisher discriminants (middle); CCA (bottom)

Ex II: Canonical Correlation Analysis

$$\widehat{\rho}(\boldsymbol{x},\boldsymbol{y}) = \max_{\substack{\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{u}=1 \\ \boldsymbol{v}^{\mathrm{T}}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})\boldsymbol{v}=1}} \boldsymbol{u}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v}$$

$\Downarrow$ reparameterize $\boldsymbol{u} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}$, $\boldsymbol{v} = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\theta}$ $[\boldsymbol{u} \in \mathbb{R}^p,\ \boldsymbol{v} \in \mathbb{R}^q,\ \boldsymbol{\alpha}, \boldsymbol{\theta} \in \mathbb{R}^n]$ $\Downarrow$

$$\widehat{\rho}(\boldsymbol{x},\boldsymbol{y}) = \max_{\substack{\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}=1 \\ \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\theta}=1}} \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\theta}$$

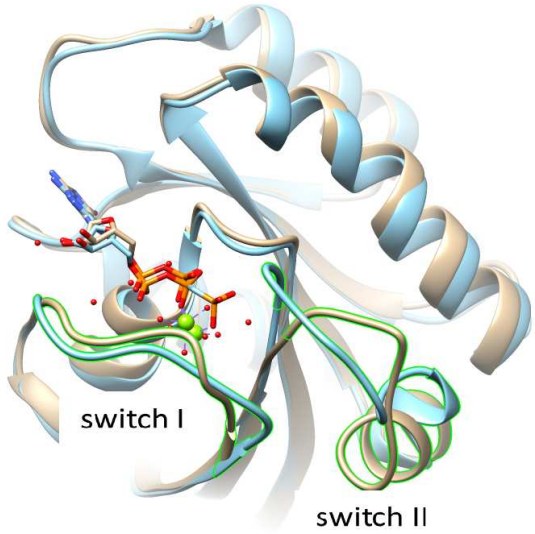
$\Downarrow$ replace $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ with $\boldsymbol{K}_X$ and $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$ with $\boldsymbol{K}_Y$ + <u>regularize</u> $\Downarrow$

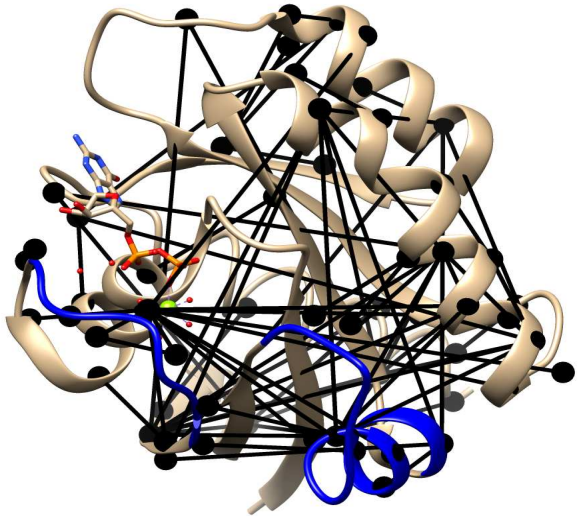
$$\widehat{\rho}_K(\boldsymbol{x},\boldsymbol{y}) = \max_{\substack{\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{K}_X^2+\lambda\boldsymbol{I})\boldsymbol{\alpha}=1 \\ \boldsymbol{\theta}^{\mathrm{T}}(\boldsymbol{K}_Y^2+\lambda\boldsymbol{I})\boldsymbol{\theta}=1}} \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{K}_X\boldsymbol{K}_Y\boldsymbol{\theta}$$

Discussion

- Easier
  - to specify a good feature space, $\varphi(\cdot)$, directly, or
  - to specify a good similarity measure, $K(\cdot;\cdot)$ instead?
- Many "fancy" kernels formed directly by $[\varphi(\cdot)]^{\mathrm{T}}[\varphi(\cdot)]$, e.g.,
  - for text,
  - for protein sequences,
  - and so on.

Application: Proteins





Want to quantify the dependence between a pair of residues, say a and b, in a protein. (Ongoing work by L. Soltan-Ghoraie, F. Burkowski & M. Zhu)

Application: Proteins

- we do so by computing $\widehat{\rho}_K(\boldsymbol{x}_{[-\boldsymbol{z}]}, \boldsymbol{y}_{[-\boldsymbol{z}]})$, where

$$\begin{aligned} \boldsymbol{x} &= \text{vector of } p \text{ dihedral angles for } a \quad (0 \le p \le 4), \\ \boldsymbol{y} &= \text{vector of } q \text{ dihedral angles for } b \quad (0 \le q \le 4), \\ \boldsymbol{z} &= \text{vector of } d \text{ dihedral angles for the rest} \quad (d \text{ large}) \end{aligned}$$

- need a sample $\mathbb{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i)\}_{i=1}^n$ — i.e., different conformations of the same protein

- need a kernel function $K(\cdot, \cdot)$, for measuring the similarity of two conformations (expressed in terms of dihedral angles)

Remark The "$\boldsymbol{x}_{[-\boldsymbol{z}]}, \boldsymbol{y}_{[-\boldsymbol{z}]}$" part is important for our application, but not so important for this course — can ignore it.

Application: Proteins

- $\mathbb{D}$: randomly fixed positions for 20% of the residues, and asked existing prediction algorithms to locate the rest

- $K(\cdot,\cdot)$: for $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^p$, used the von-Mises kernel,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad = \quad w_i w_j \prod_{t=1}^{p} \exp\left[\kappa_t \cos(x_{it} - x_{jt})\right],$$

  and likewise for $\boldsymbol{y}_i, \boldsymbol{y}_j \in \mathbb{R}^q$

- used weights w_i, w_j inversely proportional to the energy of samples i, j

Application: Proteins

$$\cos(x) \approx 1 - \frac{x^2}{2}$$

$$\Downarrow$$

von-Mises kernel

$$\exp\left[\kappa_t \cos(x_{it} - x_{jt})\right]$$

$$\approx (e^{\kappa_t}) \exp\left[-\frac{(x_{it} - x_{jt})^2}{2/\kappa_t}\right]$$

like a Gaussian kernel with "standard deviation unit"

$$\sigma_t = \sqrt{\frac{1}{\kappa_t}} \times \frac{360^\circ}{2\pi}$$

κ_t for different dihedral angles

t	κ_t	σ_t
1st	8	20°
2nd	8	20°
3rd	4	30°
4th	2	40°

σ_t rounded to nearest 10-th

[current setting; still fine-tuning]

Eigenvalue Problem

$$\max_{\|\boldsymbol{a}\|=1} \quad \boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{a}$$

$\boldsymbol{S}$ symmetric $\quad \Rightarrow \quad \boldsymbol{S} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathrm{T}}$ $\quad [\boldsymbol{U}$ orthonormal, $\boldsymbol{D}$ diagonal$]$

$$\boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{a} = \underbrace{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{U}}_{\boldsymbol{b}^{\mathrm{T}}}\boldsymbol{D}\underbrace{\boldsymbol{U}^{\mathrm{T}}\boldsymbol{a}}_{\boldsymbol{b}} = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{b} = \sum_i d_i b_i^2$$

$$\|\boldsymbol{b}\|^2 = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{b} = \boldsymbol{a}^{\mathrm{T}}\underbrace{\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}}_{\boldsymbol{I}}\boldsymbol{a} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{a} = \|\boldsymbol{a}\|^2$$

$$w_i = b_i^2 \quad \Rightarrow \quad \max_{w_1, w_2, \ldots} \quad \sum_i d_i w_i \quad \text{s.t.} \quad w_i \geq 0 \quad \text{and} \quad \sum_i w_i = 1$$

Eigenvalue Problem

$$d_1 \geq d_2 \geq ... \quad \Rightarrow \quad \widehat{w}_1 = 1, \widehat{w}_2 = \widehat{w}_3 = ... = 0$$

$$\widehat{\boldsymbol{b}} \quad = \quad (1, 0, ..., 0)^{\mathrm{T}}$$

$$\widehat{\boldsymbol{a}} = \boldsymbol{U}\widehat{\boldsymbol{b}} \quad = \quad \text{first column of } \boldsymbol{U}$$

$$= \quad \text{first eigenvector of } \boldsymbol{S}$$

$$\widehat{\boldsymbol{a}}^{\mathrm{T}}\boldsymbol{S}\widehat{\boldsymbol{a}} \quad = \quad d_1$$

$$= \quad \text{largest eigenvalue of } \boldsymbol{S}$$

Generalized Eigenvalue Problem

$$\underbrace{\max_{\substack{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{a}=1\\ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{N}\boldsymbol{b}=1}} \boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{b} \quad \Leftrightarrow \quad \max_{\|\boldsymbol{c}\|=\|\boldsymbol{d}\|=1} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{N}^{-1/2}\boldsymbol{d}}_{\boldsymbol{c}=\boldsymbol{M}^{1/2}\boldsymbol{a},\quad \boldsymbol{d}=\boldsymbol{N}^{1/2}\boldsymbol{b}}$$

solve RHS problem for $\widehat{\boldsymbol{c}}$ and $\widehat{\boldsymbol{d}}$, and then obtain

$$\widehat{\boldsymbol{a}} = \boldsymbol{M}^{-1/2}\widehat{\boldsymbol{c}} \quad \text{and} \quad \widehat{\boldsymbol{b}} = \boldsymbol{N}^{-1/2}\widehat{\boldsymbol{d}}$$

Exercise

(a) Show that the solution to the RHS problem above is:

$$\begin{aligned} \widehat{\boldsymbol{c}} &= \text{first left singular vector of } \boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{N}^{-1/2}; \\ \widehat{\boldsymbol{d}} &= \text{first right singular vector of } \boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{N}^{-1/2}. \end{aligned}$$

(b) Why do we have to regularize kernel CCA?

Summary

- key ideas:
  - margin; generalization error
  - constrained optimization; KKT conditions
  - regularized functional estimation in $\mathcal{H}_K$
  - kernelization (of algorithms driven by inner products)
- specific methods:
  - SVM
  - kernel ridge regression; kernel CCA
- applications:
  - long-range dependences in a protein molecule