Kernel Method: Data Analysis with Positive Definite Kernels

3. Various Kernel Methods

Kenji Fukumizu

The Institute of Statistical Mathematics.
Graduate University of Advanced Studies /
Tokyo Institute of Technology

Nov. 17-26, 2010
Intensive Course at Tokyo Institute of Technology

Outline

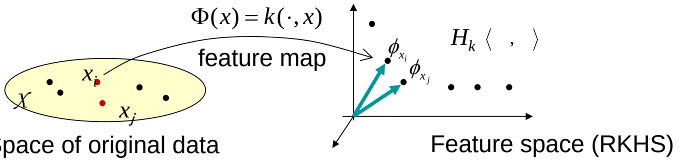
Kernel Methodology: Feature Space by RKHS

Kernel methodology = Data analysis by transforming data into a high-dimensional feature space given by RKHS.

k: positive definite kernel.

$$\Phi : \mathcal{X} \to \mathcal{H}_k, \qquad x \mapsto \Phi(x) := k(\cdot, x)$$

$$\mathcal{X} \ni X_1, \ldots, X_N \quad \mapsto \quad \Phi(X_1), \ldots, \Phi(X_N) \in \mathcal{H}_k$$



Apply linear methods on RKHS – kernelization

The computation of the inner product is feasible.

Higher-order Statistics by Positive Definite Kernel

- A nonlinear kernel includes higher-order statistics.

  Example: Polynomial kernel on $\mathbb{R}$: $k(y, x) = (yx + 1)^d$.

  - Data are transformed as $k(\cdot, X_1), \ldots, k(\cdot, X_N) \in \mathcal{H}_k$.
  - Regarding $\Phi(X) = k(y, X)$ as a function of y,

$$k(y, X) = X^d y^d + a_{d-1} X^{d-1} y^{d-1} + \cdots + a_1 X y + a_0 \qquad (a_i \neq 0).$$

  - W.r.t. the basis $\{1, y, y^2, \ldots, y^d\}$ of $\mathcal{H}_k$, the component of the feature vector $\Phi(X)$ is given by

$$(X^d, a_{d-1} X^{d-1}, \ldots, a_1 X, a_0)^T.$$

    This includes the statistics $(X, X^2, \ldots, X^d)$.

- Similar nonlinear statistics appear in other kernels such as Gaussian, Lapacian, etc.

Properties of Kernel Method

- The inner product of $\mathcal{H}$ is efficiently computable, while the dimensionality may be infinite: for $f = \sum_{i=1}^{n} a_i \Phi(X_i)$ and $g = \sum_{i=1}^{n} b_i \Phi(X_i)$,

$$\langle f, g \rangle = \sum_{i,j=1}^{n} a_i b_j k(X_i, X_j) \qquad \text{(Gram matrix)}$$

- The computational cost essentially depends on the sample size n.
  c.f. L^2 inner product / power series expansion

$$(X, Y, Z, W) \mapsto (X, Y, Z, W, X^2, Y^2, Z^2, W^2, XY, XZ, XW, YZ, \ldots)$$

- Advantageous for high-dimensional data. For a large sample, some techniques are needed (discussed in this chapter).
- Data may not be vectorial. The methods are applicable to structured data, such as strings, graphs, etc. (Discussed later).

Kernel PCA I

Kernel PCA ([SSM98])

- $X_1, \ldots, X_N$: data on $\mathcal{X}$.
- $k : \mathcal{X} \times \mathcal{X}$ positive definite kernel, $\mathcal{H}_k$: RKHS.
- Transform the data into $\mathcal{H}_k$ by $\Phi(x) = k(\cdot, x)$:

$$X_1, \ldots, X_N \quad \mapsto \Phi(X_1), \ldots, \Phi(X_N).$$

- Apply PCA to $\{\Phi(X_i)\}$ on $\mathcal{H}_k$.

$$\text{1st principal direction } = \arg\max_{\|f\|=1} \mathrm{Var}[\langle f, \Phi(X) \rangle]$$

- It suffices to use $f = \sum_{i=1}^{N} a_i \tilde{\Phi}(X_i)$, where $\tilde{\Phi}(X_i) = \Phi(X_i) - \frac{1}{N}\sum_{j=1}^{N} \Phi(X_j)$.

Kernel PCA II

- The PCA solution:
  p-th principal direction: $f^{(p)} = \sum_{i=1}^N a_i^{(p)} \tilde{\Phi}(X_i)$.

$$\max \alpha^{(p)T} \tilde{K}^2 \alpha^{(p)} \quad \text{subj. to} \quad \begin{cases} \alpha^{(p)} \tilde{K} a^{(p)} = 1 \\ \alpha^{(p)} \tilde{K} a^{(r)} = 0 \quad (r = 1, \ldots, p-1). \end{cases}$$

  where $\tilde{K}$ is $N \times N$ matrix with $\tilde{K}_{ij} = \langle \tilde{\Phi}(X_i), \tilde{\Phi}(X_j) \rangle$.

$$\tilde{K}_{ij} = k(X_i, X_j) - \tfrac{1}{N}\textstyle\sum_{b=1}^N k(X_i, X_b) - \tfrac{1}{N}\sum_{a=1}^N k(X_a, X_j) + \tfrac{1}{N^2}\sum_{a,b=1}^N k(X_a, X_b)$$

(centered Gram matrix).

Principal components of kernel PCA

$\tilde{K} = \sum_{p=1}^N \lambda_p u^{(p)} u^{(p)T}$: eigen decomposition ($\lambda_1 \geq \cdots \geq \lambda_N \geq 0$).

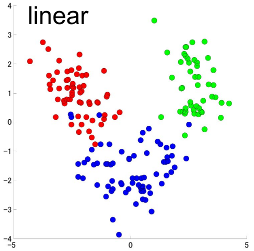
p-th principal component of the data X_i

$$= \langle \tilde{\Phi}(X_i), \textstyle\sum_{j=1}^N \alpha_j^{(p)} \tilde{\Phi}(X_j) \rangle = \sum_{j=1}^N \sqrt{\lambda_p} u_i^{(p)}.$$

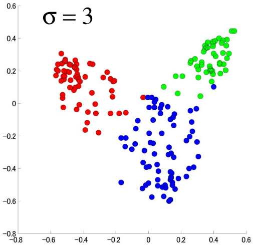
Example of Kernel PCA:

- Wine data (from UCI repository [MA94]).
  - 178 data of 13 dimension, which represent chemical measurements of different wine.
  - There are three clusters corresponding to types of wine.
  - The classes are shown in different colors, but not used for the PCA analysis.

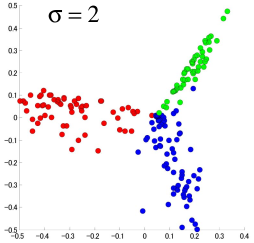
Linear PCA

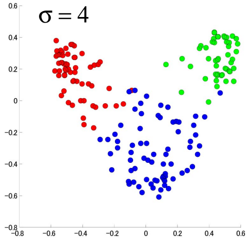


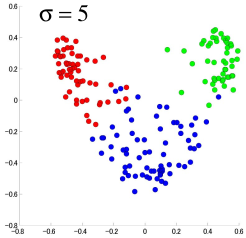
KPCA Gaussian kernel



- KPCA with Gaussian kernels. $k(x, y) = \exp\{-\frac{1}{\sigma^2}\|x - y\|^2\}$.







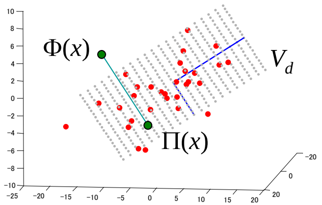
- The results depends much on the kernel parameter σ.

Application of Kernel PCA to Noise Reduction

- PCA can be used for noise reduction (principal directions represent signal).
- Apply kernel PCA to noise reduction:
  - Compute d-dim. subspace V_d spanned by $f^{(1)}, \ldots, f^{(d)}$.
  - $\Pi(x)$ ($\in \mathcal{H}_k$): orthogonal projection of $\Phi(x)$ onto V_d.
  - Find a point y in the original space such that

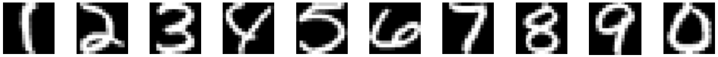
$$y = \arg\min_{y \in \mathcal{X}} \|\Phi(y) - \Pi(x)\|_{\mathcal{H}_k}.$$

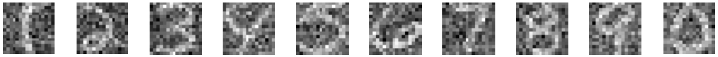
Note: $\Pi(x)$ is not necessarily in the image of Φ.

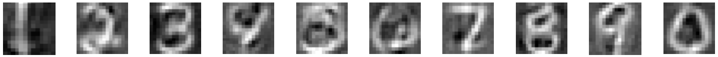


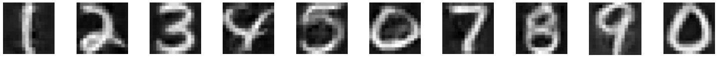
USPS hand-written digits data:

7191 images of hand-written digits of 16 $\times$ 16 pixels.


Sample of original images (not used for experiments)


Sample of noisy images


Sample of denoised images (linear PCA)


Sample of denoised images (kernel PCA, Gaussian kernel)

Generated by Matlab Stprtool (by V. Franc).

Properties of kernel PCA

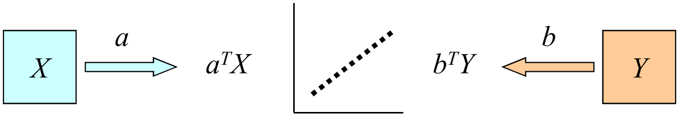
- Nonlinear PCA: Nonlinear features can be extracted.
- The results depend on the choice of kernel and kernel parameters. Interpreting the results may not be straightforward.
- Can be used for a preprocessing of other analysis such as classification and regression. (dimension reduction / feature extraction).
- How to choose a kernel and kernel parameter?
  - Cross-validation is not possible (unsupervised learning).
  - If it is a preprocessing, the performance of the final analysis should be maximized.

Canonical Correlation Analysis I

Canonical correlation analysis (CCA)

- Linear dependence of two multi-dimensional variables.
  - Data $(X_1, Y_1), \ldots, (X_N, Y_N)$, $X_i \in \mathbb{R}^m, Y_i \in \mathbb{R}^\ell$.
- Find the directions a and b so that the correlation between the projections $a^T X$ and $b^T Y$ is maximized:

$$\rho = \max_{a \in \mathbb{R}^m, b \in \mathbb{R}^\ell} \mathrm{Corr}[a^T X, b^T Y]$$



Canonical Correlation Analysis II

- CCA:

$$\rho = \max_{a \in \mathbb{R}^m, b \in \mathbb{R}^\ell} \frac{a^T \widehat{V}_{XY} b}{\sqrt{a^T \widehat{V}_{XX} a}\sqrt{b^T \widehat{V}_{YY} b}},$$

  $\widehat{V}_{XX}, \widehat{V}_{YY}, \widehat{V}_{XY}$: sample (co)variance matrices.

- Equivalent form:

$$\max a^T \widehat{V}_{XY} b \quad \text{subject to } a^T \widehat{V}_{XX} a = b^T \widehat{V}_{YY} b = 1.$$

- Solution = the largest ρ for the generalized eigenproblem:

$$\begin{pmatrix} O & \widehat{V}_{XY} \\ \widehat{V}_{YX} & O \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \rho \begin{pmatrix} \widehat{V}_{XX} & O \\ O & \widehat{V}_{YY} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

Canonical Correlation Analysis III

Derivation:

Lagrange multiplier method.

$$L(a,b;\mu,\nu) = a^T\widehat{V}_{XY}b + \frac{\mu}{2}(a^T\widehat{V}_{XX}a - 1) + \frac{\nu}{2}(b^T\widehat{V}_{YY}b - 1).$$

From $\partial L/\partial a = 0, \partial L/\partial b = 0,$

$$\widehat{V}_{XY}b + \mu\widehat{V}_{XX}a = 0, \qquad \widehat{V}_{YX}a + \nu\widehat{V}_{YY}b = 0.$$

From $\partial L/\partial \mu = 0, \partial L/\partial \nu = 0,$

$$a^T\widehat{V}_{XX}a = b^T\widehat{V}_{YY}b = 1. \qquad \text{(constraints)}$$

1st equation $\Rightarrow$ $\quad a^T\widehat{V}_{XY}b = -\mu a^T\widehat{V}_{XX}a = -\mu$.
2nd equation $\Rightarrow$ $\quad b^T\widehat{V}_{YX}a = -\nu b^T\widehat{V}_{YY}b = -\nu$.

Thus, $\mu = \nu$. Set $\rho = -\mu = -\nu$. Then,

$$\widehat{V}_{XY}b = \rho\widehat{V}_{XX}a, \qquad \widehat{V}_{YX}a = \rho\widehat{V}_{YY}b.$$

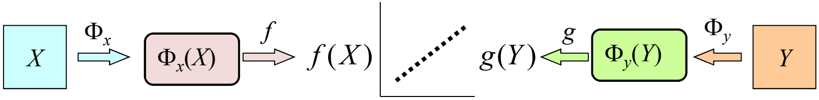
Kernel CCA I

Kernel CCA: kernelization of CCA ([Aka01, MRB01, BJ02]).

- Data: $(X_1, Y_1), \ldots, (X_N, Y_N)$.
  - X_i, Y_i: arbitrary variables taking values in $\mathcal{X}$ and $\mathcal{Y}$ (resp.).
- Transforming: prepare kernels $k_\mathcal{X}$ on $\mathcal{X}$ and $k_\mathcal{Y}$ on $\mathcal{Y}$.

$$X_1, \ldots, X_N \quad \mapsto \quad \Phi_\mathcal{X}(X_1), \ldots, \Phi_\mathcal{X}(X_N) \in \mathcal{H}_\mathcal{X}.$$
$$Y_1, \ldots, Y_N \quad \mapsto \quad \Phi_\mathcal{Y}(Y_1), \ldots, \Phi_\mathcal{Y}(Y_N) \in \mathcal{H}_\mathcal{Y}.$$

- Apply CCA on $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$.



Kernel CCA II

$$\rho = \max_{f\in\mathcal{H}_\mathcal{X}, g\in\mathcal{H}_\mathcal{Y}} \frac{\sum_{i=1}^N \langle f, \tilde{\Phi}_\mathcal{X}(X_i)\rangle_{\mathcal{H}_\mathcal{X}} \langle g, \tilde{\Phi}_\mathcal{Y}(Y_i)\rangle_{\mathcal{H}_\mathcal{Y}}}{\sqrt{\sum_{i=1}^N \langle f, \tilde{\Phi}_\mathcal{X}(X_i)\rangle^2_{\mathcal{H}_\mathcal{X}}}\sqrt{\sum_{i=1}^N \langle g, \tilde{\Phi}_\mathcal{Y}(Y_i)\rangle^2_{\mathcal{H}_\mathcal{Y}}}}$$
$$= \max_{f\in\mathcal{H}_\mathcal{X}, g\in\mathcal{H}_\mathcal{Y}} \frac{\mathrm{Cov}[f(X_i), g(Y_i)]}{\mathrm{Var}[f(X_i)]^{1/2}\mathrm{Var}[g(Y_i)]^{1/2}},$$

where $\tilde{\Phi}_\mathcal{X}(X_i) = \Phi_\mathcal{X}(X_i) - \frac{1}{N}\sum_{j=1}^N \Phi_\mathcal{X}(X_j)$, and $\tilde{\Phi}_\mathcal{Y}(Y_i)$ similar.

- We can assume $f = \sum_{i=1}^N \alpha_i \tilde{\Phi}_\mathcal{X}(X_i)$ and $g = \sum_{i=1}^N \beta_i \tilde{\Phi}_\mathcal{Y}(Y_i)$.

$$\rho = \max_{\alpha\in\mathbb{R}^N, \beta\in\mathbb{R}^N} \frac{\alpha^T \tilde{K}_X \tilde{K}_Y \beta}{\sqrt{\alpha^T \tilde{K}_X^2 \alpha}\sqrt{\beta^T \tilde{K}_Y^2 \beta}},$$

$\tilde{K}_X$ and $\tilde{K}_Y$ are the centered Gram matrices.

Kernel CCA III

- This problem is ill-posed with correlation 1, (if $\mathcal{R}(\tilde{K}_X)) \cap \mathcal{R}(\tilde{K}_Y)) \neq 0$).
- Kernel CCA with regularization:

$$\max_{f \in \mathcal{H}_\mathcal{X}, g \in \mathcal{H}_\mathcal{Y}} \frac{\sum_i \langle f, \tilde{\Phi}_\mathcal{X}(X_i) \rangle_{\mathcal{H}_\mathcal{X}} \langle g, \tilde{\Phi}_\mathcal{Y}(Y_i) \rangle_{\mathcal{H}_\mathcal{Y}}}{\sqrt{\sum_i \langle f, \tilde{\Phi}_\mathcal{X}(X_i) \rangle^2_{\mathcal{H}_\mathcal{X}} + \varepsilon_N \|f\|^2} \sqrt{\sum_i \langle g, \tilde{\Phi}_\mathcal{Y}(Y_i) \rangle^2_{\mathcal{H}_\mathcal{Y}} + \varepsilon_N \|g\|^2}}$$

- Kernel CCA

$$\begin{pmatrix} O & \tilde{K}_X \tilde{K}_Y \\ \tilde{K}_Y \tilde{K}_X & O \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \rho \begin{pmatrix} \tilde{K}_X^2 + \varepsilon_N K_X & O \\ O & \tilde{K}_Y^2 + \varepsilon_N K_y \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

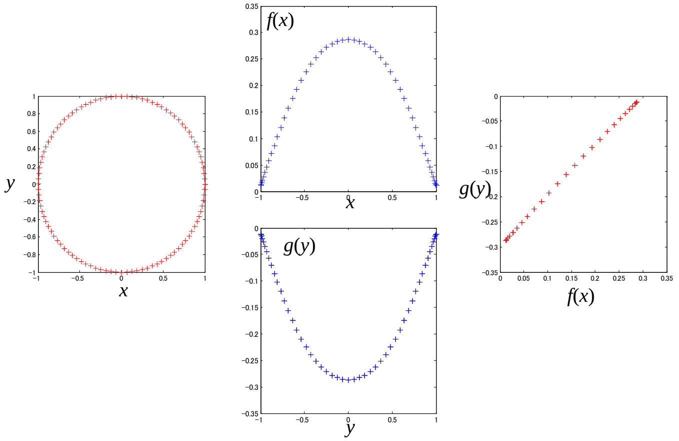
The Solution is obtained as a generalized eigenproblem.

Some Properties of Kernel CCA

- The multiple feature vectors (second, third, eigenvectors) can be also obtained.
- The canonical correlation value may not represent the dependence value well (by regularization).
- The results depends on the choice of kernels and ε_N. Choice of parameters:
  - Cross-validation may be possible.
  - Some methods have been proposed ([HSST04] See later.).
- The consistency is known if ε_N decreases sufficiently slowly as $N \to \infty$ ([FBG07]).

Toy Example of Kernel CCA

X, Y: one-dimensional. Gaussian RBF kernels are used.



Application of Kernel CCA to Image Retrieval ([HSST04])

Idea: use d eigenvectors $f_1, \ldots, f_d$ and $g_1, \ldots, g_d$ as the feature spaces which contain the dependence between X and Y.

- X_i: image, Y_i: text (extracted from the same webpage).

Y_i: 'Phoenix', 'sky', 'harbor', ...

- For text, "bag-of-words" kernel (histogram of frequency of words) is used.
- Compute the d-eigenvectors $f_1, \ldots, f_d$ and $g_1, \ldots, g_d$ by kernel CCA.
- The regularization parameter ε is chosen so that

$$\varepsilon = \arg\max \|\boldsymbol{\rho}(\varepsilon) - \boldsymbol{\rho}_R(\varepsilon)\|$$

($\boldsymbol{\rho}(\varepsilon)$: eigenspectrum of KCCA. $\boldsymbol{\rho}_R$: eigenspectrum with randomized data.)

- Compute the feature vectors by projections $\xi_i = (\langle \Phi_{\mathcal{X}}(X_i), f_a \rangle_{\mathcal{H}_{\mathcal{X}}})_{a=1}^{d} \in \mathbb{R}^d$ for all images.
- For a text query Y_{new}, compute the feature $\zeta = (\langle \Phi_{\mathcal{Y}}(Y_{new}), g_a \rangle_{\mathcal{H}_{\mathcal{Y}}})_{a=1}^{d} \in \mathbb{R}^d$, and output the image such that

$$\arg\max_i = \xi_i^T \zeta.$$

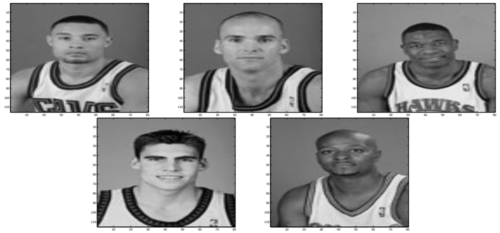


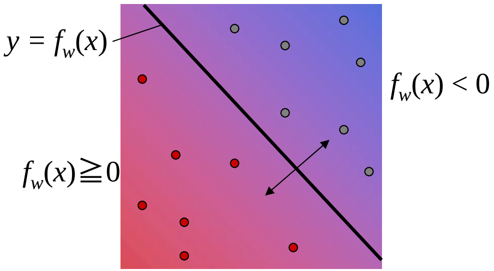
Figure 3 Images retrieved for the text query: "height: 6-11 weight: 235 lbs position: forward born: september 18, 1968, split, croatia college: none"

From Hardoon et al. *Neural Computation* (2004).

Linear Classifier

- $(X_1, Y_1), \ldots, (X_N, Y_N)$: data
  - X_i: explanatory variable (m-dimensional)
  - $Y_i \in \{+1, -1\}$ binary,
- Linear classifier

$$f(x) = \operatorname{sgn}\left(w^T x + b\right)$$



Large Margin Classifier I

Linear support vector machine (in $\mathbb{R}^m$)

- Assumption: the data is linearly separable.
- Large margin criterion:
  Among infinite number of separating hyperplanes, choose the one to give the largest margin.
  - Margin = distance of two classes measured along the direction of w.
  - The classifying hyperplane is the middle of the margin.

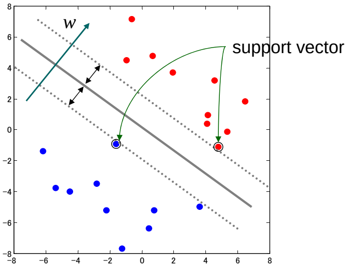
Large Margin Classifier II

To fix a scale, assume

$$\begin{cases} \min(w^T X_i + b) = 1 & i : Y_i = +1 \\ \max(w^T X_i + b) = -1 & i : Y_i = -1 \end{cases}$$

Then,

$$\text{Margin} = \frac{2}{\|w\|}$$



Large Margin Classifier III

- Large margin linear classifier

$$\max \frac{1}{\|w\|} \qquad \text{subj. to } \begin{cases} w^T X_i + b \geq 1 & \text{if } Y_i = +1, \\ w^T X_i + b \leq -1 & \text{if } Y_i = -1. \end{cases}$$

Equivalently,

Linear support vector machine (hard margin)

$$\min_{w,b} \|w\|^2 \qquad \text{subject to} \qquad Y_i(w^T X_i + b) \geq 1 \quad (\forall i).$$

- This problem is quadratic programming (QP, quadratic objective function with linear constraints. Discussed later).
  - free from local minima!
  - Many standard solvers available.

SVM with Soft Margin

Relax the separability assumption. The linear separability is too restrictive in practice.

- Hard constraint: $Y_i(w^T X_i + b) \geq 1$
- Soft constraint: $Y_i(w^T X_i + b) \geq 1 - \xi_i \quad (\xi_i \geq 0)$

Linear support vector machine (soft margin)

$$\min_{w,b,\xi_i} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \qquad \text{subj. to} \qquad \begin{cases} Y_i(w^T X_i + b) \geq 1 - \xi_i, \\ \xi_i \geq 0. \end{cases}$$

- The optimization is still QP.
- C is a hyper-parameter, which we have to decide.

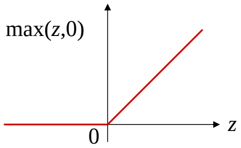
Soft Margin as Regularization

- Soft margin linear SVM is equivalent to the following regularization problem ($\lambda = 1/C$):

$$\min_{w,b} \sum_{i=1}^{N} \big(1 - Y_i(w^T X_i + b)\big)_+ + \lambda \|w\|^2$$

where

$$(z)_+ = \max(z, 0)$$



- $\ell(f(x), y) = (1 - yf(x))_+$: hinge loss.

Tikhonov Regularization

General theory of regularization

- When the solution of the optimization

$$\min_{\alpha \in A} \Omega(\alpha)$$

  $(A \subset \mathcal{H})$ is not unique or stable, a regularization technique is often used.
- Tikhonov regularization: add a regularization term (or penalty term), e.g.,

$$\min_{\alpha \in A} \quad \Omega(\alpha) + \lambda \|\alpha\|^2.$$

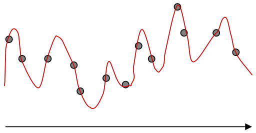
  $\lambda > 0$: regularization coefficient.
- The solution is often unique and stable.
- Other regularization terms, such as $\|\alpha\|$ and $\sum_i |\alpha_i|$, are also possible, but differentiability may be lost.

Tikhonov Regularization II

- Example
  - Ill-posed problem:

    $$\min_f (Y_i - f(X_i))^2.$$

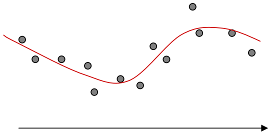
    Many f give zero error, if f is taken from a large space.



  - Regularized objective function

    $$\min_f (Y_i - f(X_i))^2 + \lambda \|f\|^2$$

    finds a unique solution, which is often smoother
    $\Rightarrow$ Kernel ridge regression.



SVM with Kernels I

Kernelization of linear SVM

- $(X_1, Y_1), \ldots, (X_N, Y_N)$: data
  - X_i: arbitrary covariate taking values in $\mathcal{X}$,
  - $Y_i \in \{+1, -1\}$ binary,
- k: positive definite kernel on $\mathcal{X}$. $\mathcal{H}$: associated RKHS.
- $\Phi(X_i) = k(\cdot, X_i)$: transformed data in $\mathcal{H}$.

- Linear classifier on RKHS

$$f(x) = \mathrm{sgn}\big(\langle h, \Phi(x)\rangle_{\mathcal{H}} + b\big) = \mathrm{sgn}(h(x) + b).$$

SVM with kernels II

- Large margin objective function (soft margin):

$$\min_{h,b,\xi_i} \|h\|_{\mathcal{H}}^2 + C\sum_{i=1}^{N}\xi_i \quad \text{subj. to} \quad \begin{cases} Y_i(\langle h, \Phi(X_i)\rangle + b) \geq 1 - \xi_i, \\ \xi_i \geq 0, \end{cases}$$

  or equivalently

$$\min_{h,b} \sum_{i=1}^{N}\big(1 - Y_i(\langle h, \Phi(X_i)\rangle + b)\big)_+ + \lambda\|h\|^2$$

- It suffices to assume

$$h = \textstyle\sum_{i=1}^{N} c_i \Phi(X_i)$$

  The orthogonal direction only increases the regularization term without changing the first term.

- Note

$$\|h\|^2 = \textstyle\sum_{i,j=1}^{N} c_i c_j k(X_i, X_j), \quad \langle h, \Phi(X_i)\rangle = \sum_{j=1}^{N} c_j k(X_i, X_j).$$

SVM with kernels III

In summary,

SVM with kernel

$$\min_{c_i, b, \xi_i} \sum_{i,j=1}^{N} c_i c_j k(X_i, X_j) + C \sum_{i=1}^{N} \xi_i,$$

$$\text{subj. to} \quad \begin{cases} Y_i(\sum_{j=1}^{N} k(X_i, X_j) c_j + b) \geq 1 - \xi_i, \\ \xi_i \geq 0. \end{cases}$$

- The optimization is numerically solved with QP.
- The dual form is simpler to solve (discussed later.)
- The parameter C and the kernel are often chosen by cross-validation.

Demonstration of SVM

Webpages for SVM Java applet

- `http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml`

Results on character recognition

MNIST: Handwritten digit recognition
28×28 binary pixels.

60000 training data
10000 test data

	k-NN Euclid	10PCA + quad.	RBF + lin.	LeNet-4	LeNet-5	SVM poly4	RS-SVM poly5
Test error (%)	5.0	3.3	3.6	1.1	0.95	1.1	1.0

Taken from [LBBH01]

Mini-summary on SVM

- Kernel trick (a common property of kernel methods):
  - linear classifier on RKHS.
  - High-dimensional feature space, but the computation of inner product is easy.
- Large margin criterion
  - May not be the Bayes optimal, but causes other good properties.
- Quadratic programming:
  - The objective function is solved by the standard QP.
- Sparse representation:
  - The classifier is represented by a small number of support vectors (discussed in the next lecture).
- Regularization:
  - The soft margin objective function is equivalent to the margin loss with regularization.

Review on Derivation of Kernel Methods

- For the objective function of kernel methods, the solution $f \in \mathcal{H}_k$ has the form

$$f = \sum_{i=1}^{N} \alpha_i k(\cdot, X_i).$$

- By plugging the above form in the objective function, the problem can be typically written by Gram matrices.

- Optimize the objective function by a suitable method, *e.g.* matrix inversion, eigendecomposition, quadratic program, etc.

Representer Theorem I

Minimization problems on RKHS

$$\min_{f\in\mathcal{H}_k}\sum_{i=1}^{N}(Y_i - f(X_i))^2 + \lambda\|f\|^2 \qquad \text{(kernel ridge regression)},$$

$$\min_{f\in\mathcal{H}_k, b}\sum_{i=1}^{N}\big(1-(Y_i f(X_i)+b)\big)_+ + \lambda\|f\|^2 \qquad \text{(SVM)}.$$

$$\min_{f\in\mathcal{H}_k}\Big[-\textstyle\sum_{i=1}^{N}\Big(f(X_i)-\frac{1}{N}\sum_{j=1}^{N}f(X_j)\Big)^2\Big]+I(\|f\|) \quad \text{(Kernel PCA)},$$

where $I(t)=0$ for $t\le 1$ and $=\infty$ for $t>1$.

We have seen that the solution can be taken from

$$f=\sum_{i=1}^{N}\alpha_i k(\cdot, X_i).$$

Representer Theorem II

- General problem:
  - $\mathcal{H}$: RKHS with associated with a positive definite kernel k.
  - $X_1, \ldots, X_N, Y_1, \ldots, Y_N$: data.
  - $h_1(x), \ldots, h_m(x)$: fixed functions.
  - $\Psi : [0\ \infty) \to \mathbb{R} \cup \{+\infty\}$: non-decreasing function (regularization).

Minimization

$$\min_{f \in \mathcal{H}, c \in \mathbb{R}^m} L\Big(\{X_i\}_{i=1}^N, \{Y_i\}_{i=1}^N, \{f(X_i) + \textstyle\sum_{a=1}^m c_a h_a(X_i)\}_{i=1}^N\Big) + \Psi(\|f\|).$$

Representer theorem

The solution of the above minimization is given by the form

$$f = \textstyle\sum_{i=1}^N \alpha_i k(\cdot, X_i).$$

- The optimization in an high (or infinite) dim. space is reduced to the problem of N dimension (sample size).

Proof of Representer Theorem

- Decomposition:

$$\mathcal{H}_k = H_0 \oplus H_0^{\perp},$$

  $H_0 = \text{span}\{k(\cdot, X_1), \ldots, k(\cdot, X_N)\}$, $H_0^{\perp}$: orthogonal complement.
  Decompose

$$f = f_0 + f^{\perp}$$

  accordingly.
- Because

$$\langle f^{\perp}, k(\cdot, X_i) \rangle = 0,$$

  the loss function L does not change by replacing f with f_0.
- The second term:

$$\|f_0\| \leq \|f\| \qquad \Longrightarrow \qquad \Psi(\|f_0\|) \leq \Psi(\|f\|).$$

- Thus, the optimum f can be in the space H_0.

Kernel Fisher Discriminant Analysis

- Fisher's linear discriminant analysis (LDA):
  - X: m-dimensional explanatory variable.
  - Y represents binary classes. $Y \in \{\pm 1\}$.
  - Find the linear classifier

    $$h(x) = w^T X + b$$

    so that it maximizes

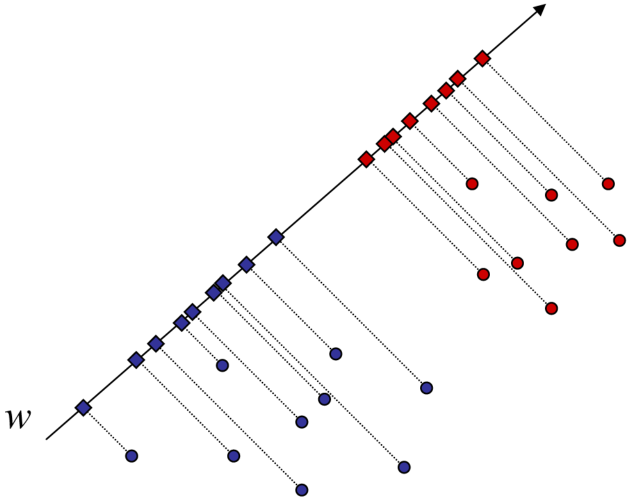
    $$J(w) = \frac{\text{Between-class variance along } w}{\text{Sum of within-class variances along } w}.$$

- Kernel Fisher Discriminant Analysis (Kernel FDA):
  - Find the linear classifier in RKHS,

    $$h(x) = f(x) + b = \langle f, \Phi(x) \rangle + b$$

    so that it maximizes

    $$J_{\mathcal{H}}(f) = \frac{\text{Between-class variance along } f}{\text{Sum of within-class variances along } f}.$$


w

Kernel Logistic Regression

- Logistic regression:
  - X: m-dimensional explanatory variable
  - Y represents L classes.
    $Y \in \{(1,0,\dots,0),(0,1,0,\dots,0),(0,\dots,0,1)\}$.
  - In Binary case ($Y \in \{\pm 1\}$),

$$P(Y=+1|X) = \frac{e^{a^T X+b}}{1+e^{a^T X+b}} = \frac{1}{1+e^{-(a^T X+b)}},$$
$$P(Y=-1|X) = \frac{1}{1+e^{aX+b}},$$

or equivalently

$$P(Y|X) = \frac{1}{1+e^{-Y(a^T X+b)}} \qquad (Y \in \{\pm 1\}).$$

  - With sample $(X_1,Y_1),\dots,(X_N,Y_N)$,

$$\max_{a,b} \sum_{i=1}^{N} -\log(1+e^{-Y_i(a^T X_i+b)}).$$

- Kernel Logistic Regression: ([Rot01, ZH05])
    - Objective function

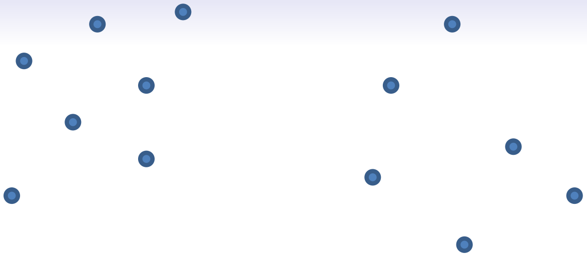
$$\min_{f,b} \sum_{i=1}^{N} \log(1 + e^{-Y_i(f(X_i)+b)}) + \lambda \|f\|^2$$

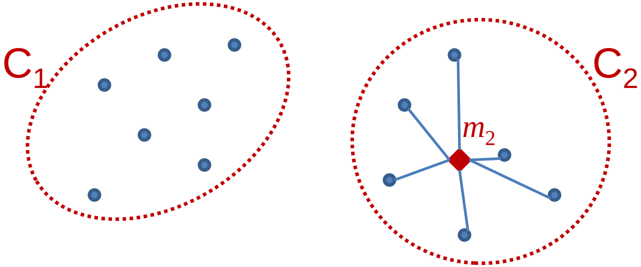
    - The objective function is convex, but not so simple as QP.

Kernel K-means Clustering

- K-means clustering:
  - Partition $X_1, \dots, X_N$ into K clusters $C_1, \dots, C_K$.
  - Objective
$$\min \sum_{k=1}^{K} \sum_{X_i \in C_k} \|X_i - m_k\|^2$$
    where $m_k = \frac{1}{|C_k|} \sum_{X_j \in C_k} X_j$ (mean vector in C_k).
  - Iterative algorithm is used.

- Kernel K-means clustering: ([DGK04])
  Since the mean and norm can be computed for feature vectors, we can kernelize K-means clustering.




C_1
C_2
m_2

Other Kernel Methods

- Kernel PLS (partial least square)
- Support vector regression (SVR)
- ν-SVM
- One-class SVM etc...

Choice of Kernel

How to choose / design a kernel?

- Reflect knowledge on the problem as much as possible. (structured data)
- For supervised learning such as SVM, use cross-validation.
- For unsupervised learning such as kernel PCA and kernel CCA, there are no theoretically guaranteed methods so far.

  Suggestions: make a relevant supervised method and use cross-validation.
- Kernel learning:
  - Multiple kernel learning (MKL): optimize a kernel among $\sum_{a=1}^{L} w_\ell k_\ell(x, y)$.

Supervised and Unsupervised Learning

Supervised learning:

- Data for input X and output Y are prepared.
- Y is regarded as supervisor or teacher of the learning.

$$X \quad \mapsto \quad f(X) \;\approx\; Y.$$

- *e.g.* classification, regression, prediction.

Unsupervised learning:

- There is no teaching data Y.
- *e.g.* PCA, CCA, clustering.

Semisupervised learning is also considered.

Empirical Loss and Expected Loss I

Supervised learning:

- $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$: training data. i.i.d. sample.
- $X_i \in \mathcal{X}$: input, $\quad Y_i \in \mathcal{Y}$: output.
- $\mathcal{F} \subset \{f : \mathcal{X} \to \mathcal{Y}\}$: function class.

Expected loss and empirical loss

- Loss function $\ell(y, f)$: measure discrepancy of Y_i and $f(X_i)$.
  e.g. $\ell(y, f) = \|y - f\|^2$ (square error)

Empirical Loss and Expected Loss II

- Empirical loss (training error):

$$L_n(f) = \widehat{E}_n[\ell(Y, f(X))] = \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, f(X_i)) \qquad (f \in \mathcal{F}).$$

- Expected loss (test error, prediction error): final goal of learning is to minimize the expected loss:

$$L(f) = E[\ell(Y, f(X))] \qquad (f \in \mathcal{F}).$$

- Learning must be done with data:

$$\widehat{f} = \arg\min_{f \in \mathcal{F}} L_n(f).$$

Examples of Loss Function

- Mean square error.
  - $\ell(y, f) = (y - f)^2$.
  - Empirical loss: $\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (Y_i - f(X_i))^2$ (least mean square).
  - Expected loss = $E[(Y - f(X))^2]$ (mean square error)
- 0-1 loss. $y, f(x) \in \{\pm 1\}$.
  - $\ell(y, f) = \frac{1 - yf(x)}{2}$.
  - Empirical loss = ratio of errors: $\frac{1}{n} |\{i \mid Y_i \neq f(X_i)\}|$.
  - Expected error = mean error rate: $\Pr(Y \neq f(X))$.
- Log likelihood
  - $\ell(y, f) = -\log p(y|f)$.
  - Empirical loss = - Empirical log likelihood.
  - Expected loss = - Expected log likelihood.

Estimation of Expected Loss

- We wish to know the expected loss $L(\widehat{f})$.

$$L(\widehat{f}) - \underbrace{\widehat{L}_n(\widehat{f})}_{\text{known}} = \underbrace{E[\ell(Y,\widehat{f}(X))|\mathcal{D}] - \widehat{E}_n[\ell(Y,\widehat{f}(X))]}_{?}.$$

- Approaches to analysis.
    - Asymptotic expansion of the expectation:

      e.g. $$E_{\mathcal{D}}\Big[E[\ell(Y,\widehat{f}(X))] - \widehat{E}_n[\ell(Y,\widehat{f}(X))]\Big] = \frac{A}{n} + \ldots$$

      $\Longrightarrow$ AIC, GIC.
    - Upper bound: (PAC)

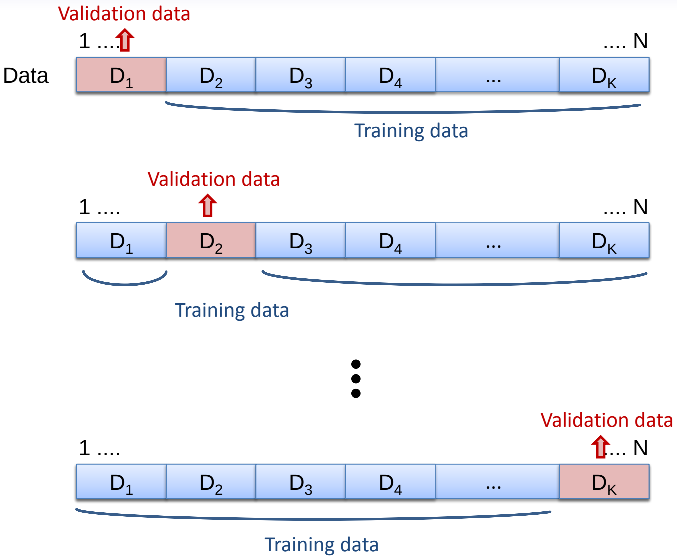
      e.g. $$\Pr\big(E[\ell(Y,\widehat{f}(X))|\mathcal{D}] \le \widehat{E}_n[\ell(Y,\widehat{f}(X))] + \varepsilon\big)$$
      $$\le \Pr\Big(\sup_{f\in\mathcal{F}}\big(E[\ell(Y,f(X))] - \widehat{E}_n[\ell(Y,f(X))]\big) \le \varepsilon\Big) \quad \le \alpha e^{-\beta\varepsilon^2 n}.$$

- For SVM the 2nd approach is often used, but not discussed in this course.

Cross-Validation I

- Cross-validation (CV): a method of estimating expected loss.
- K-fold CV
  - Partitioned data (randomly) into K subsamples.
  - $i = 1, \dots, K$
    - Use i-th subsample for testing (validation), and use the remaining data for training.
  - Average the K losses.
- Leave-one-out CV (LOOCV)
  - $K = N$. For $i = 1, \dots, N$, use i-th data for testing, and the remaining data for training.
  - Average N losses.

Cross-Validation II



K-fold cross-validation

Cross-Validation III

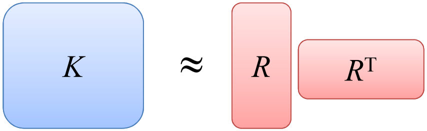
- If data is an i.i.d. sample of size N, LOOCV is an unbiased estimator for the expected error given by $N - 1$ training data.
- CV (especially LOOCV) is computationally expensive.

Low-Rank Approximation I

- If the sample size N is large, operations on Gram matrix K is not feasible.
  Inversion, eigendecomposition costs $O(N^3)$.
- Low-rank approximation:

$$K \approx RR^T$$

where R is $N \times r$ matrix ($r \ll N$).



Low-Rank Approximation II

- Computational cost is reduced drastically. For example, in kernel ridge regression,

$$\begin{aligned} Y^T(K+\lambda I_N)^{-1}\mathbf{k}(x) &\approx Y^T(RR^T+\lambda I_N)^{-1}\mathbf{k}(x) \\ &= \frac{1}{\lambda}\{Y^T\mathbf{k}(x) - Y^T R(R^T R+\lambda I_r)^{-1}R^T\mathbf{k}(x)\}, \end{aligned}$$

  which costs $O(r^2N + r^3)$.

- Two popular methods for low-rank approximation:
  - Incomplete Cholesky decomposition: sample complexity $O(r^2N)$, space complexity $O(rN)$.
  - Nyström approximation: random sampling + eigendecomposition.

Summary of Section 3

- Various classical linear methods of data analysis can be kernelized – efficient linear algorithms on RKHS.

  Kernel PCA, SVM, kernel CCA, kernel FDA, etc.

- The solution often has the form

  $$f = \sum_{i=1}^{N} \alpha_i k(\cdot, X_i)$$

  (representer theorem).

- The problem is reduced to operations on Gram matrices of the sample size N.

- The kernel methods can be applied to any type of data including non-vectorial (structured) data, such as graphs, strings, etc, if a positive definite kernel is provided.

References I

[Aka01] Shotaro Akaho.
A kernel method for canonical correlation analysis.
In *Proceedings of International Meeting on Psychometric Society (IMPS2001)*, 2001.

[BJ02] Francis R. Bach and Michael I. Jordan.
Kernel independent component analysis.
Journal of Machine Learning Research, 3:1–48, 2002.

[DGK04] I. S. Dhillon, Y. Guan, and B. Kulis.
Kernel k-means, spectral clustering and normalized cuts.
In *The Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 551–556, 2004.

[FBG07] Kenji Fukumizu, Francis R. Bach, and Arthur Gretton.
Statistical consistency of kernel canonical correlation analysis.
Journal of Machine Learning Research, 8:361–383, 2007.

References II

[HSST04] David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor.
Canonical correlation analysis: An overview with application to learning methods.
Neural Computation, 16:2639–2664, 2004.

[LBBH01] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner.
Gradient-based learning applied to document recognition.
In Simon Haykin and Bart Kosko, editors, *Intelligent Signal Processing*, pages 306–351. IEEE Press, 2001.

[MA94] Patrick M. Murphy and David W. Aha.
UCI repository of machine learning databases.
Technical report, University of California, Irvine, Department of Information and Computer Science.
http://www.ics.uci.edu/~mlearn/MLRepository.html, 1994.

References III

[MRB01] Thomas Melzer, Michael Reiter, and Horst Bischof.
Nonlinear feature extraction using generalized canonical correlation analysis.
In *Proceedings of International Conference on Artificial Neural Networks (ICANN)*, pages 353–360, 2001.

[Rot01] Volker Roth.
Probabilistic discriminative kernel classifiers for multi-class problems.
In *Pattern Recognition: Proceedings of 23rd DAGM Symposium*, pages 246–253. Springer, 2001.

[SSM98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller.
Nonlinear component analysis as a kernel eigenvalue problem.
Neural Computation, 10:1299–1319, 1998.

References IV

[ZH05] Ji Zhu and Trevor Hastie.
Kernel logistic regression and the import vector machine.
Journal of Computational and Graphical Statistics, 14(1):185–205, 2005.