Machine Learning Research School

Introduction to Kernel Methods

Kenji Fukumizu
The Institute of Statistical Mathematics /
Graduate School of Advanced Studies

August 10-11, 2019, Bangkok

Kernel method: nonlinear data analysis

- Kernel methods
  - Methodology developed since 1990's.
  - Started from the Support Vector Machine (Boser, Guyon, Vapnik 1992).
  - Suitable for extracting nonlinear features.
  - Applicable to non-vectorial data, e.g, strings, graphs, probabilities.

Vladimir Vapnik (1936 -)

The goal of this lecture

- To learn basic ideas of kernel methods through
  - mathematical aspects
  - typical examples of kernel methods: kernel ridge regression, kernel PCA
- To learn (very) basics of Support Vector Machine.
- To learn recent developments
  - Approximation methods for large data
  - Structured data and mean embedding

Outline

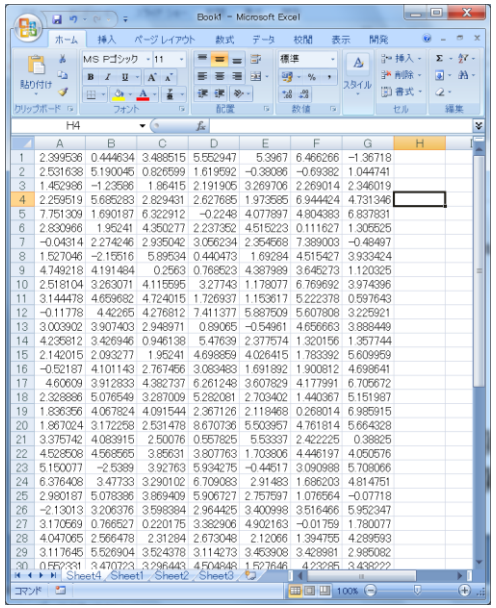
1. Introduction
2. Basic ideas and examples
3. Brief introduction to Support Vector Machine
4. Approximation for scalability
5. Non-vectorial data
6. Summary

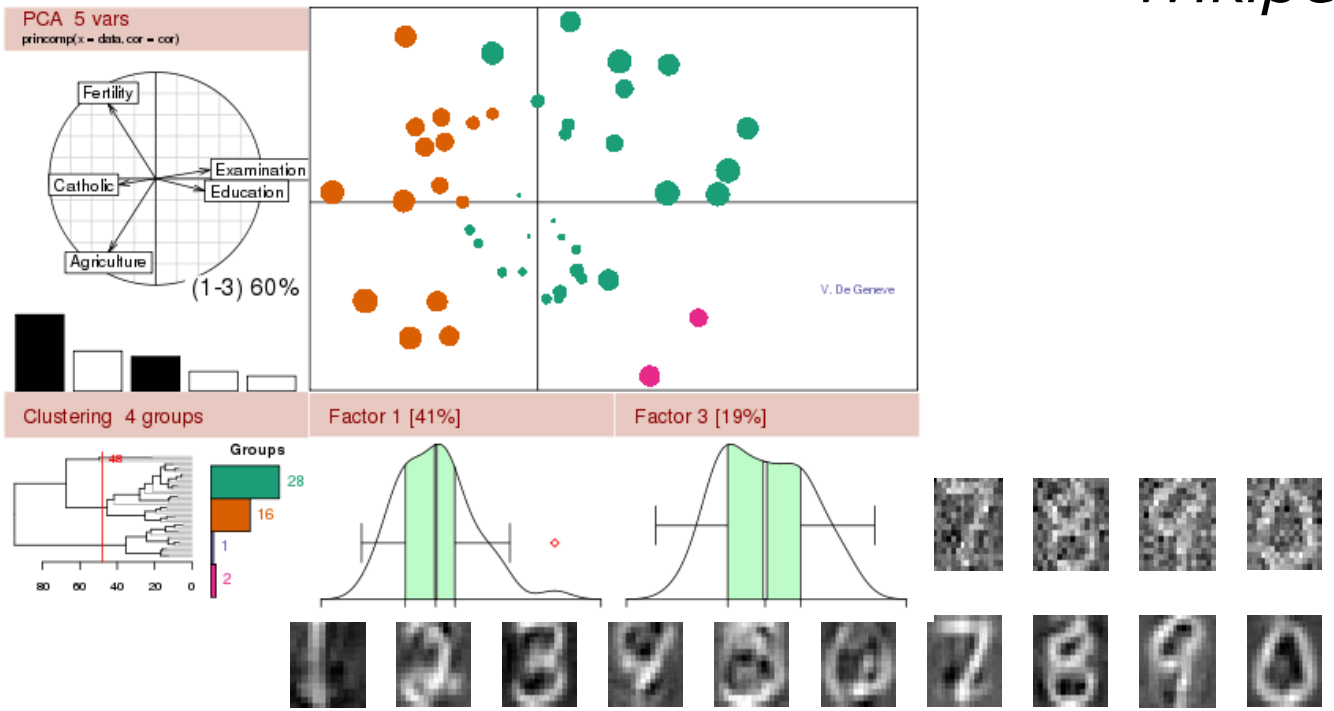
1. Introduction
Linear and nonlinear methods

Data analysis

Analysis of data is a process of inspecting, cleaning, transforming, and modeling data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making.

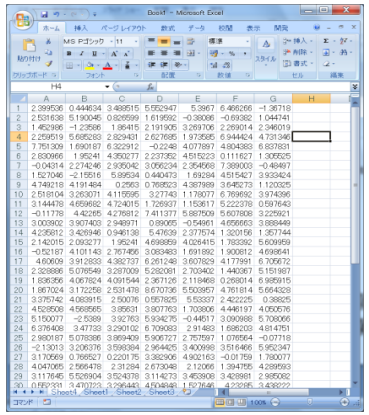
– *Wikipedia*





“Linear” data analysis

- Table of numbers



Matrix expression

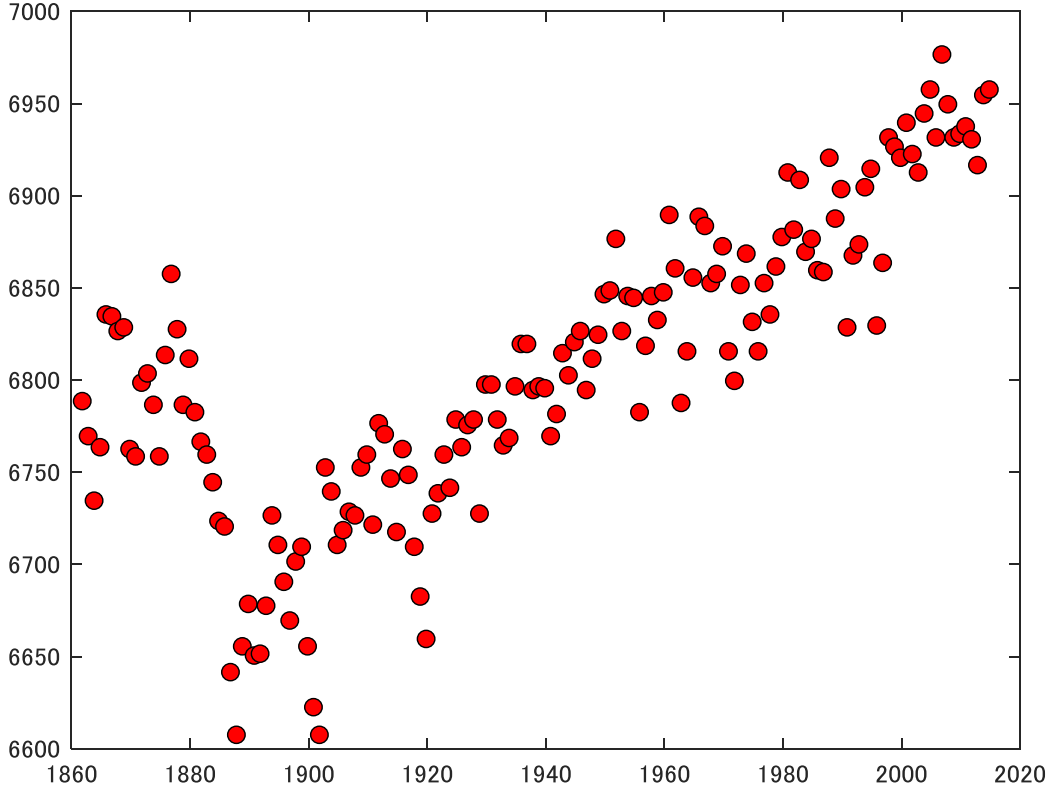
$$\mathbf{X} = \begin{pmatrix} X_1^{(1)} & \cdots & X_m^{(1)} \\ X_1^{(2)} & \cdots & X_m^{(2)} \\ \vdots & \ddots & \vdots \\ X_1^{(N)} & \cdots & X_m^{(N)} \end{pmatrix}$$

m dimension
N data

- Linear algebra works. E.g
  - Principal component analysis (PCA)
  - Correlation, Canonical correlation analysis (CCA)
  - Linear regression, Fisher descriminaiton, logistic regression, etc.

Are linear methods enough?

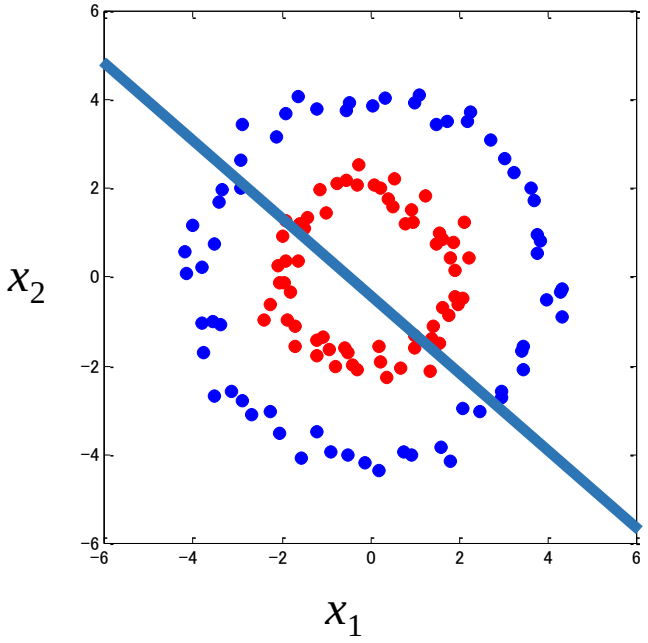
Change of average sea level (Vlissingen, Netherland)



Permanent Service for Mean Sea Level (PSMSL), 2016, "Tide Gauge Data", Retrieved 05 Dec 2016. Holgate et al. (2013) *ournal of Coastal Research*.

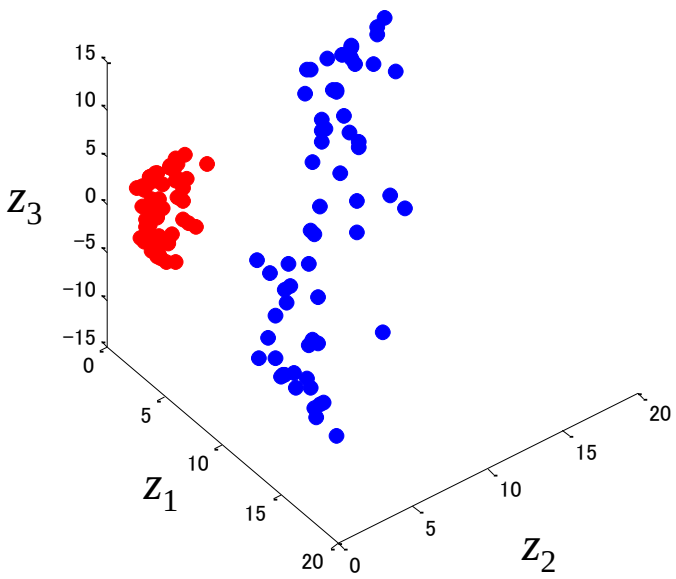
Binary classification

Linearly inseparable



transform

Linearly separable



$$(z_1, z_2, z_3) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

Watch the movie! https://www.youtube.com/watch?v=3liCbRZPrZA

Nonlinear transform helps!

Analysis of data is a process of inspecting, cleaning, transforming, and modeling data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making.

- *Wikipedia.*

Kernel method = a systematic way of transforming data into a high-dimensional feature space to extract nonlinearity or higher-order moments of data.

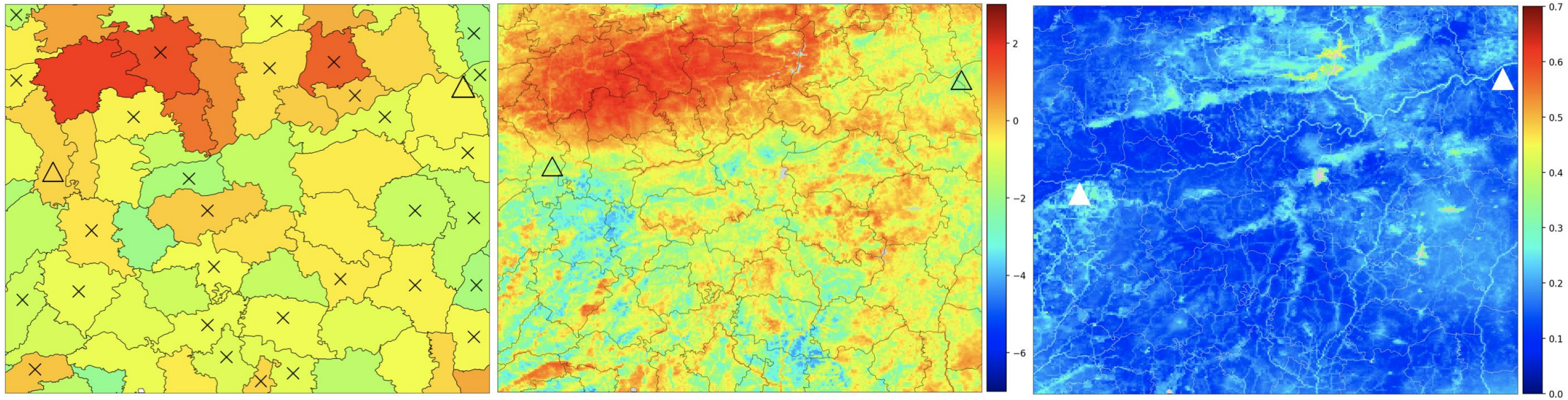
Easy to apply for mapping non-vector data to vectors.

Why kernel methods?

- An example

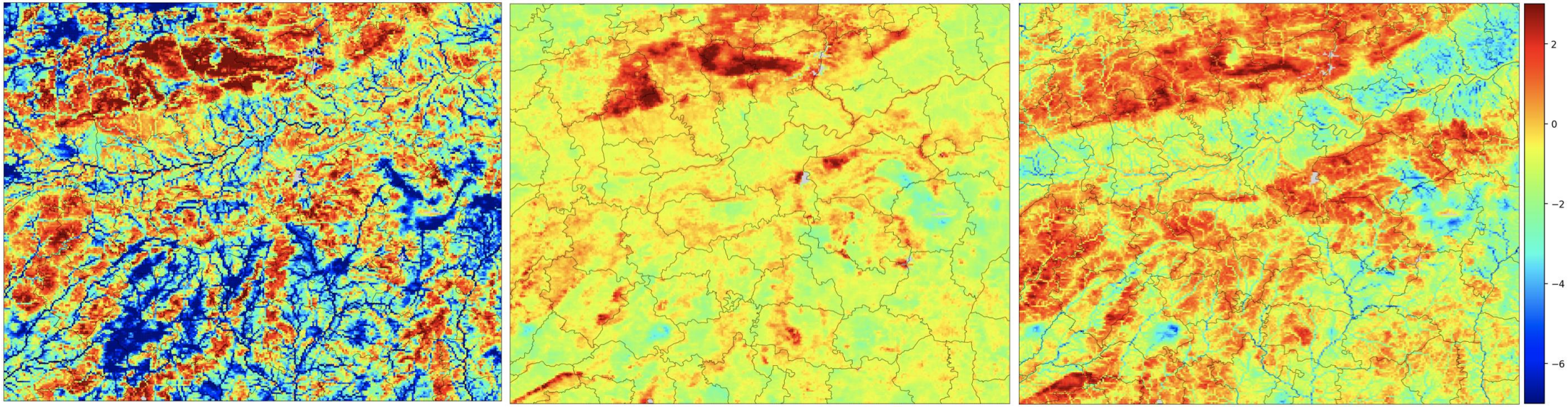
  Estimation of frequency of malaria patients per population

  Kernel-based method



Average intensity

Neural networks with three different learning results (random choice of training / validation data)



This problem is not a complete supervised problem.
The NN is too flexible and the results are random.

2. Basic ideas and examples of kernel methods

Nonlinear mapping of data

Does the following power expansion always work?

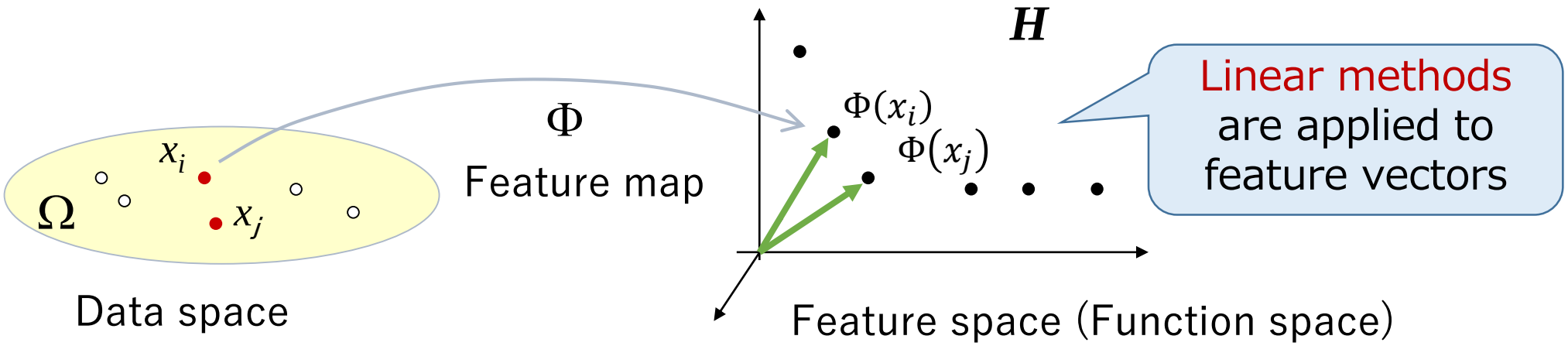
$$(X, Y, Z) \mapsto (X, Y, Z, X^2, Y^2, Z^2, XY, YZ, ZX, \dots)$$

- Computational issue:
  Imagine our data is high dimensional:
  the above approach is intractable!

  e.g. 10000 dimensional data, up to 2nd moments
  ${}_{10000}C_1 + {}_{10000}C_2 = 50{,}015{,}000$

- Need a cleverer way → Kernel method.

Kernel methods: big picture



- In kernel methods,
  - the feature map can incorporate various nonlinear information of the original data.
  - the inner product can be easily computable.
    Essential for many linear methods for data analysis.

Positive definite kernel – definition

Def. Ω: set. $k : \Omega \times \Omega \to \mathbf{R}$

k is positive definite if the following two conditions hold.

1. [Symmetry] $k(x, y) = k(y, x)$
2. [Positive definite] For any number of points $x_1, \ldots, x_n \in \Omega$, the matrix

$$\begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix}$$

Gram matrix

is positive semidefinite (non-negative definite), i.e.,

$$\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \qquad (\forall c_i \in \mathbf{R}).$$

- Examples: positive definite kernels on $\mathbf{R}^m$:
  - Euclidean inner product

$$k_{lin}(x, y) = x^T y$$

  - Gaussian RBF kernel

$$k_G(x, y) = \exp\left(-\frac{1}{2\sigma^2}||x - y||^2\right) \quad (\sigma > 0)$$

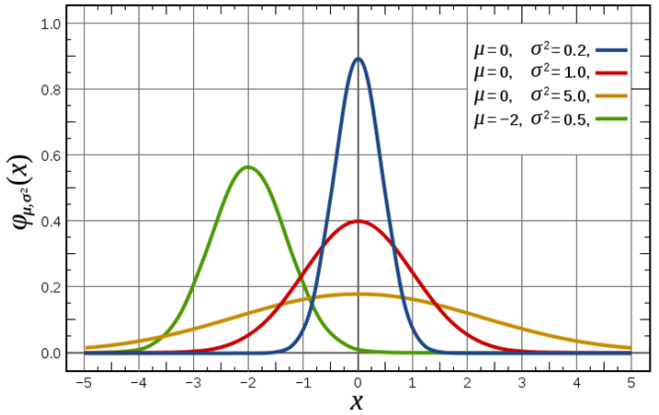
  - Laplace kernel

$$k_L(x, y) = \exp\left(-\alpha \sum_{a=1}^{d} |x_a - y_a|\right) \quad (\alpha > 0)$$

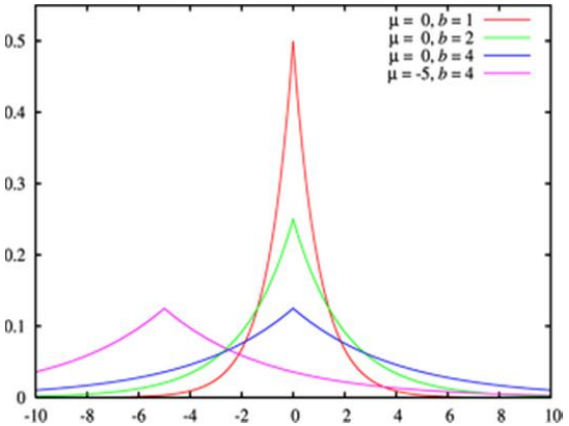
  - Polynomial kernel

$$k_P(x, y) = (c + x^T y)^d \qquad (c \geq 0, d \in \mathbf{N})$$

Gaussian



Laplace



Positive definite kernel is “inner product”

- Inner product defines positive definite kernel

Proposition

H: vector space with inner product $\langle \cdot, \cdot \rangle$, Φ: $\Omega \to H$: any map.
Define

$$k(x, y) \coloneqq \langle \Phi(x), \Phi(y) \rangle.$$

Then $k(x, y)$ is necessarily positive definite.

$$\sum_{i,j=1}^{n} c_i c_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \sum_{i,j=1}^{n} \langle c_i \boldsymbol{x}_i, c_j \boldsymbol{x}_j \rangle = \left\langle \sum_i c_i \boldsymbol{x}_i , \sum_j c_j \boldsymbol{x}_j \right\rangle = \left\| \sum_i c_i \boldsymbol{x}_i \right\|^2 \geq 0.$$

Review: inner product
(1) [symmetry] $(x, y) = (y, x)$
(2) [linearity] $(ax, y) = a(x, y)$ $a \in \mathbf{R}$, $(x_1 + x_2, y) = (x_1, y) + (x_2, y)$
(3) [positivity] $\|x\|^2 \coloneqq (x, x) \geq 0$. $\|x\| = 0 \Leftrightarrow x = 0$.

- Positive definite kernel defines an inner product

Theorem (Aronszajn)

If $k\colon \Omega \times \Omega \to \mathbf{R}$ is positive definite, there is a Hilbert space H_k with inner product $\langle , \rangle_{H_k}$, consisting of functions on Ω such that

1) $k(\cdot, x) \in H$ for all $x \in H$,
2) $\{\sum_{i=1}^{n} c_i\, k(\cdot, x_i) \in H \mid c_i \in R, x_i \in \Omega\}$ is dense in H,
3) (Reproducing property)

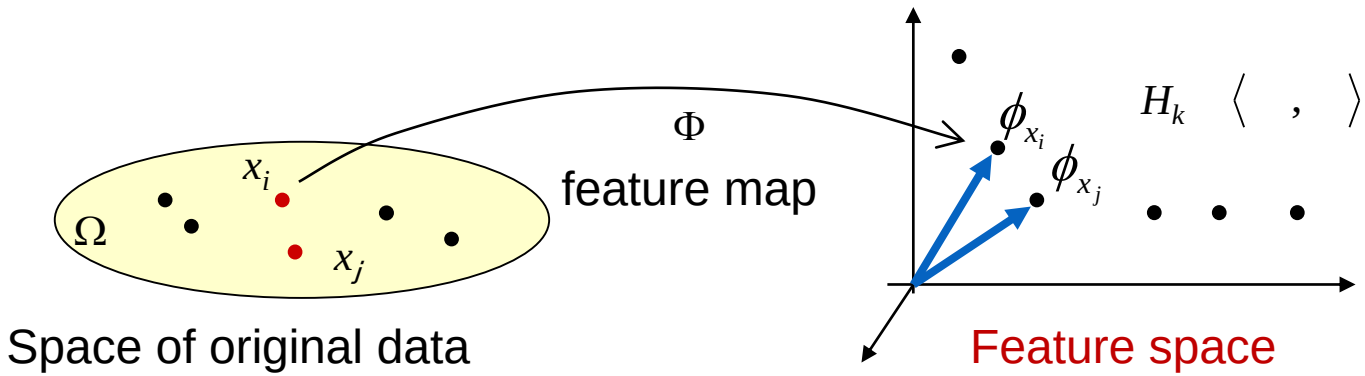
$$\langle f, k(\cdot, x)\rangle_{H_k} = f(x) \qquad \text{for any } x \in \Omega, f \in H_k.$$

Notation: $k(\cdot, x)$: function of the first argument with x fixed.

H_k is called a Reproducing Kernel Hilbert Space (RKHS) defined by k.

*_Hilbert space_ is a vector space with inner product (+ the norm is complete).

Kernel trick



By Aronszajn's theorem, for positive definite k,

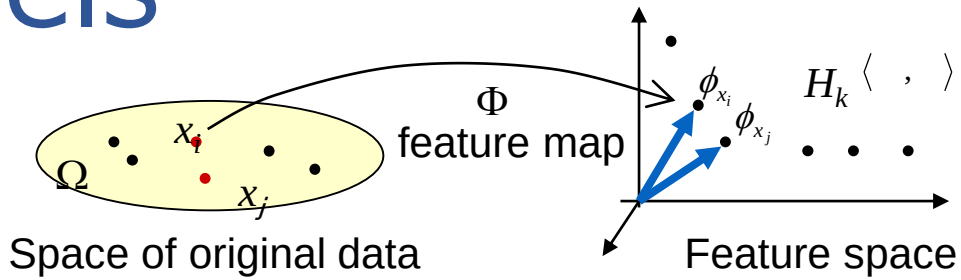
$\Phi(x) \coloneqq k(\cdot, x)$: mapping $\Omega \to H$ Feature map

$$\langle \Phi(x), \Phi(y) \rangle = k(x, y)$$ Kernel trick

Plug $f = k(\cdot, y)$ to the reproducing property.

Data analysis with kernels

- Prepare kernel k
- Feature space= RKHS
- Feature map:



$$\Phi\colon \Omega \to H, \qquad x \mapsto k(\cdot, x)$$

$$X^{(1)}, \ldots, X^{(n)} \mapsto \; k\left(\cdot\,, X^{(1)}\right), \ldots, k\left(\cdot\,, X^{(n)}\right)$$

- Kernel trick: $\langle \Phi\left(X^{(i)}\right), \Phi\left(X^{(j)}\right) \rangle = k\left(X^{(i)}, X^{(j)}\right)$
- In practice, all we need is the Gram matrices .
  - In many data analysis methods, the inner products among data suffices.
  - Not dependent on the dimensionality of original data, once Gram matrix is computed.
  - Applicable to non-vectorial data

Example 1: Kernel Ridge Regression

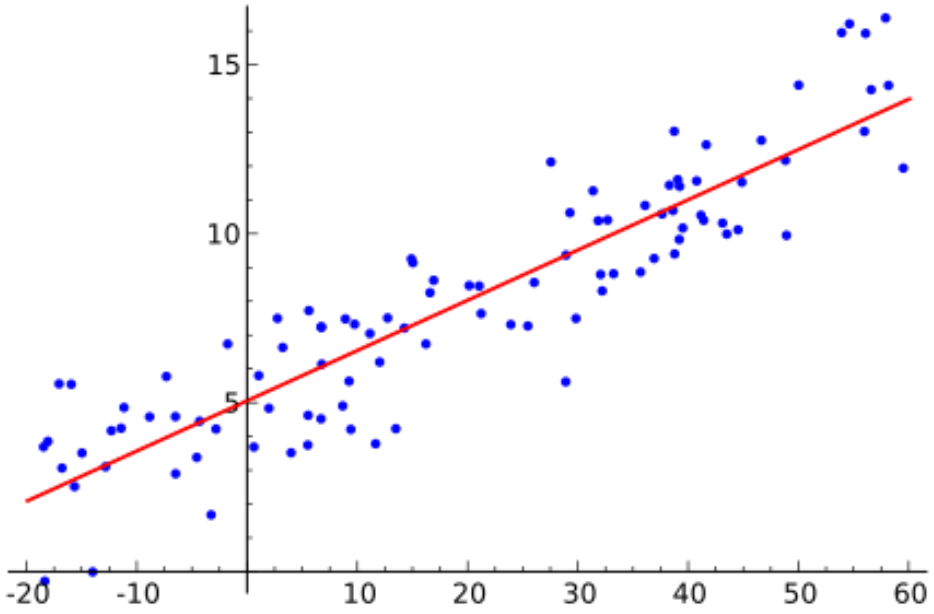
- Ridge regression

  Linear regression with L_2-penalty

$$\min_{a} \quad \frac{1}{n}\sum_{i=1}^{n}(Y_i - a^T X_i)^2 + \lambda\|a\|^2$$

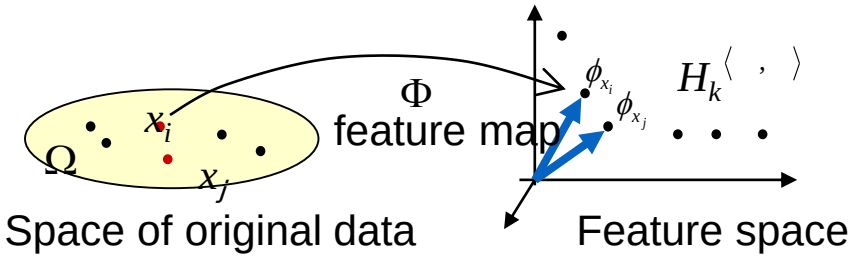
  Solution: $\hat{a} = (X^T X + \lambda I_n)^{-1} X^T Y$

  Often used when $X^T X$ is (almost) non-invertible.



- Kernel Ridge Regression

Linear: $\min_{a} \frac{1}{n}\sum_{i=1}^{n}(Y_i - a^T X_i)^2 + \lambda\|a\|^2$



Kernel: $\min_{f\in H} \frac{1}{n}\sum_{i=1}^{n}(Y_i - \langle f, \Phi(X_i)\rangle_H)^2 + \lambda\|f\|_H^2$ RR in feature space

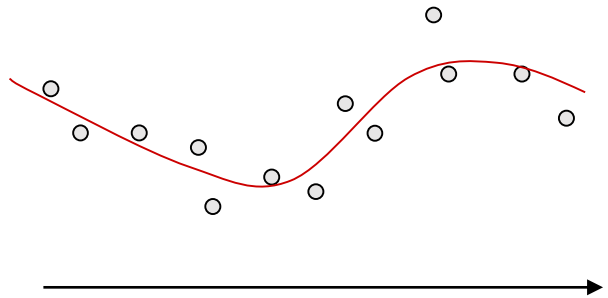
Reproducing property

$\|f\|_H^2$ is related to the smoothness of f
→ Section 4

Equivalently

$$\min_{f\in H} \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - f(X_i)\right)^2 + \lambda\|f\|_H^2$$

Nonlinear regression



- Kernel ridge regression

$$\hat{f}(x) = \mathbf{k}(x)^T (K + n\lambda I_n)^{-1} \mathbf{Y}_n$$

$K_{ij} = k(X_i, X_j)$ Gram matrix ($n \, x \, n$ matrix)

$\mathbf{Y}_n = (Y_1, \ldots, Y_n)^T$

$\mathbf{k}(x) = \big(k(x, X_1), \ldots, k(x, X_n)\big)^T$

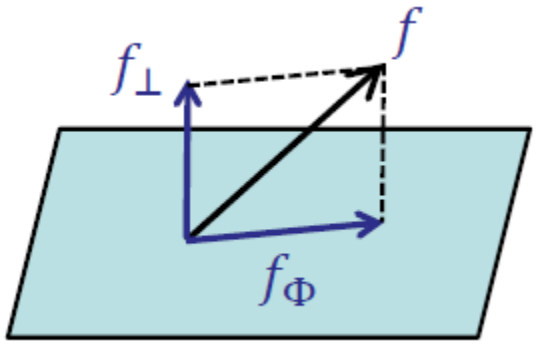
- K is positive semi-definite, but may have very small eigenvalues.
- Regularization coefficient λ ensures the invertibility
- A larger λ gives a smoother solution.

- How to solve it?

Key: $f = \sum_{i=1}^{n} c_i \Phi(X_i) = \sum_{i=1}^{n} c_i k(\cdot, X_i)$ is sufficient.
[Reprensenter Theorem]

∵) Orthogonal decomposition $f = f_\Phi + f_\perp$. $f_\Phi = \sum_{i=1}^{n} c_i \Phi(X_i)$

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \langle f, \Phi(X_i) \rangle_H)^2 + \lambda \|f\|_H^2$$



- $\langle f, \Phi(X_i) \rangle_H = \langle f_\Phi, \Phi(X_i) \rangle_H$ [Orthogonality]
- $\|f\|_H^2 = \|f_\Phi\|_H^2 + \|f_\perp\|_H^2$ [Pythagorean]

$f = f_\Phi$ (i.e. $f_\perp = 0$) gives a smaller value for the objective function.

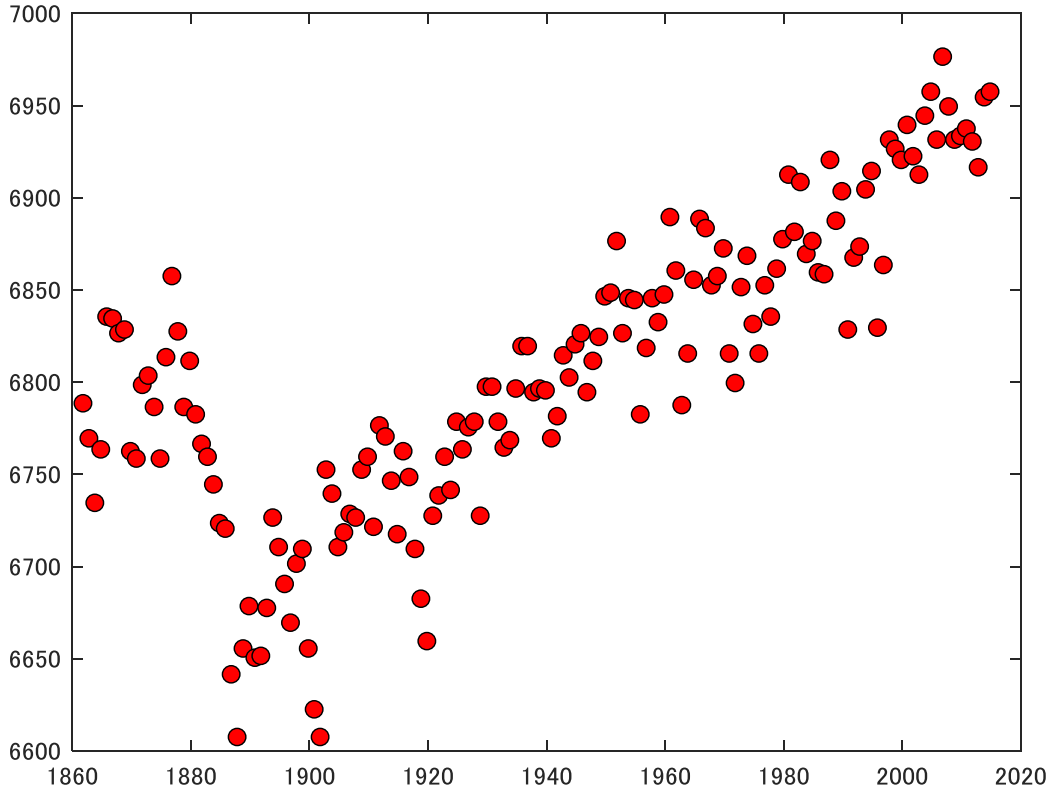
Plug-in → $\frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \sum_j c_j k(X_i, X_j)\right)^2 + \lambda \sum_{ij} c_i c_j k(X_i, X_j)$

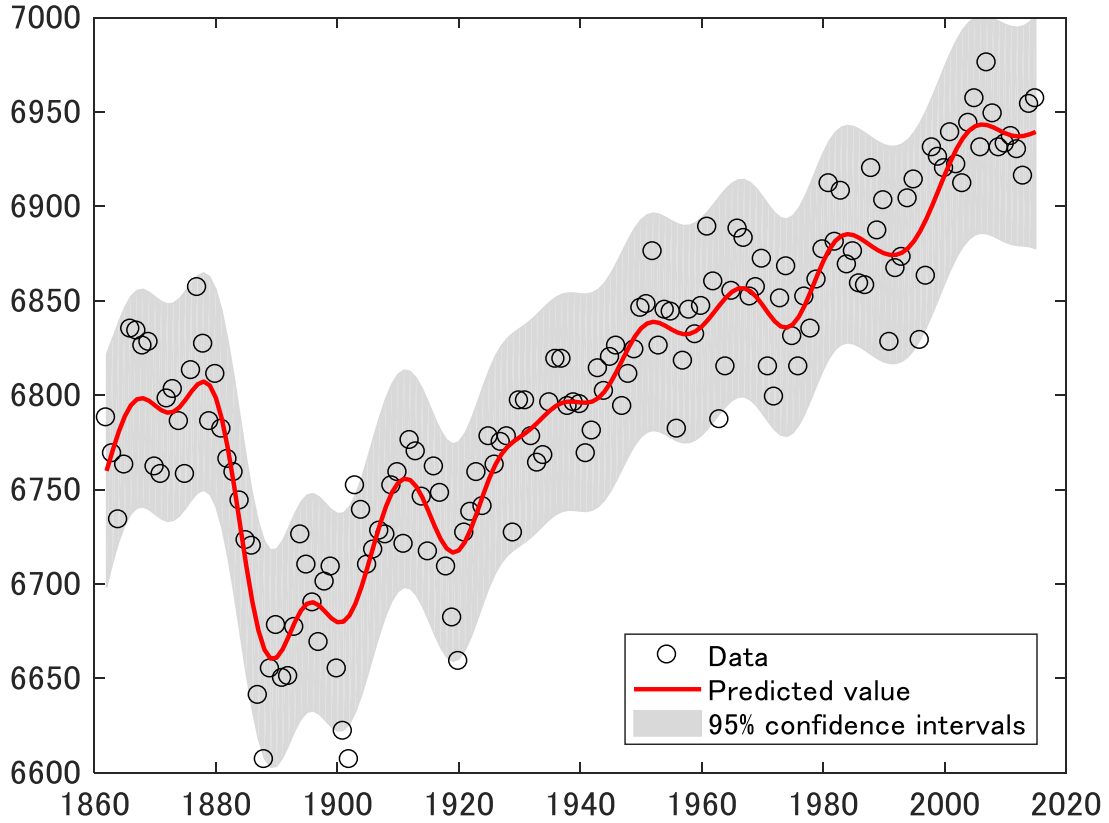
$$\hat{c} = (K + n\lambda I_n)^{-1} \mathbf{Y}_n, \quad K_{ij} = k(X_i, X_j)$$

$\mathbf{Y}_n = (Y_1, \ldots, Y_n)^T$

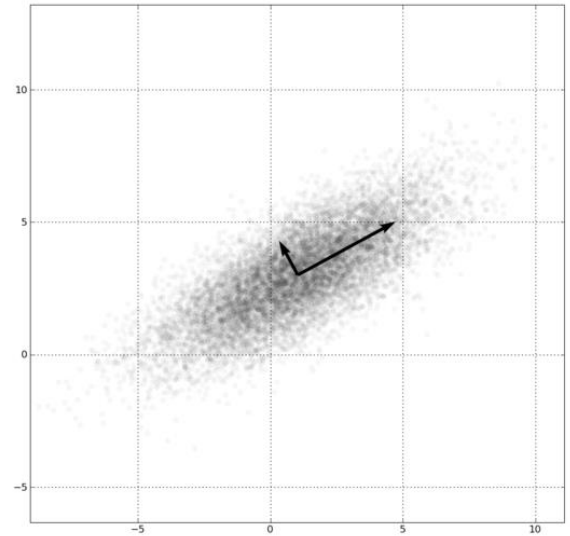
Exercise: Compute $\langle f, \Phi(X_i) \rangle_H$ and $\|f\|_H^2$ for $f = \sum_{i=1}^{n} c_i \Phi(X_i)$

Example: application of kernel ridge regression to the sea level data.





Example 2: Kernel PCA



- Principal Component Analysis (PCA)

  Projection to a subspace with large variance.

  Principal direction a : $\max_{a:\|a\|=1} \sum_{i=1}^{n} \left(a^T (X_i - \bar{X})\right)^2$

  $\bar{X} = \frac{1}{n} \sum_{j=1}^{n} X_j$

  - Reduced to an eigenproblem.

- Kernel PCA (Schölkopf et al 1999) : PCA on a feature space

  Principal direction f : $\max_{f:\|f\|_H=1} \sum_{i=1}^{n} (\langle f, \Phi(X_i) - \bar{\Phi}(X) \rangle)^2$,

  $\bar{\Phi}(X) = \frac{1}{n} \sum_{j=1}^{n} \Phi(X_j)$

- Solution

  Reprensenter theorem: $f = \sum_{i=1}^{n} c_i(\Phi(X_i) - \overline{\Phi(X)})$ is sufficient. (Decompose as $f = f_\Phi + f_\perp$. Then, $\langle f, \Phi(X_i)\rangle_H = \langle f_\Phi, \Phi(X_i)\rangle_H$ and $\|f\|_H^2 = \|f_\Phi\|_H^2 + \|f_\perp\|_H^2$.

$$\max_f \quad \sum_{i=1}^{n}(\langle f, \Phi(X_i) - \bar{\Phi}(X)\rangle)^2 = c^T\tilde{K}^2c, \qquad \|f\|_H^2 = c^T\tilde{K}c = 1$$

where $\tilde{K}_{ij} = k(X_i, X_j) - \frac{1}{n}\sum_{b=1}^{n} k(X_i, X_b) - \frac{1}{n}\sum_{a=1}^{n} k(X_a, X_j) + \frac{1}{n^2}\sum_{a.b=1}^{n} k(X_a, X_b)$

Centered Gram matrix

By solving an eigenproblem, f is obtained.

- Algorithm: [Exercise: derive this.]

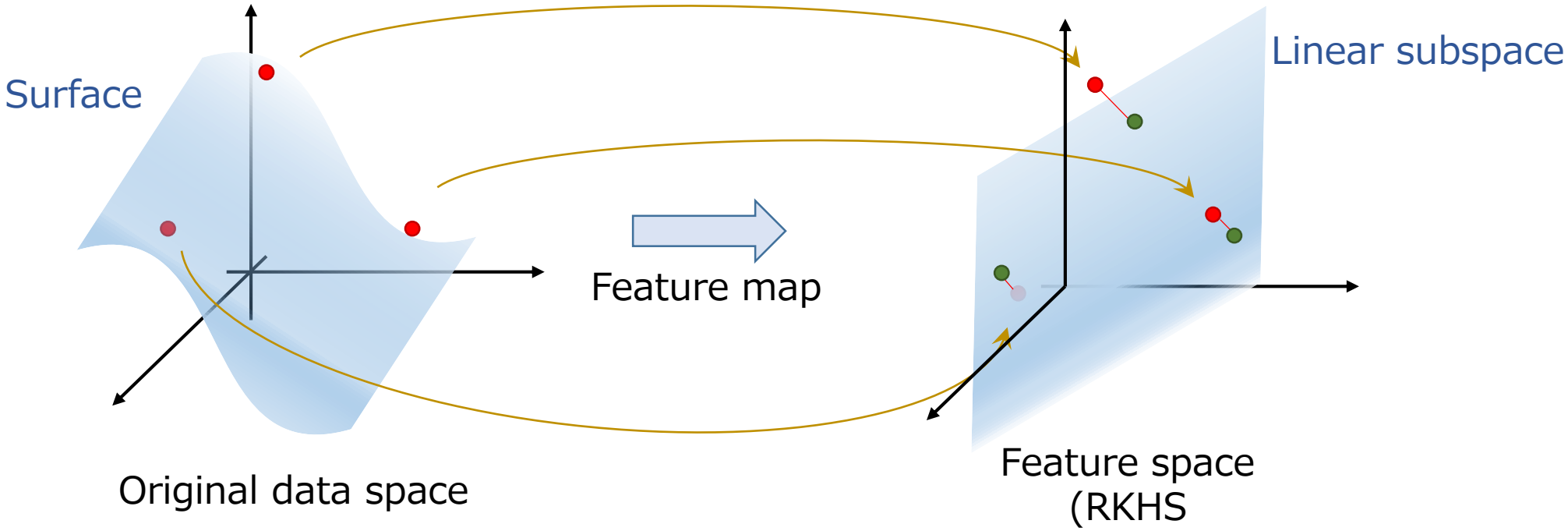
Kernel PCA

$\tilde{K}$: centered Gram matrix.

$$\tilde{K} = \sum_{i=1}^{n} \lambda_i u_i u_i^T,$$

$\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ eigenvalues, u_i : unit eigenvector

p-th principal component of X_i $= \sqrt{\lambda_p}\, u_{pi}$

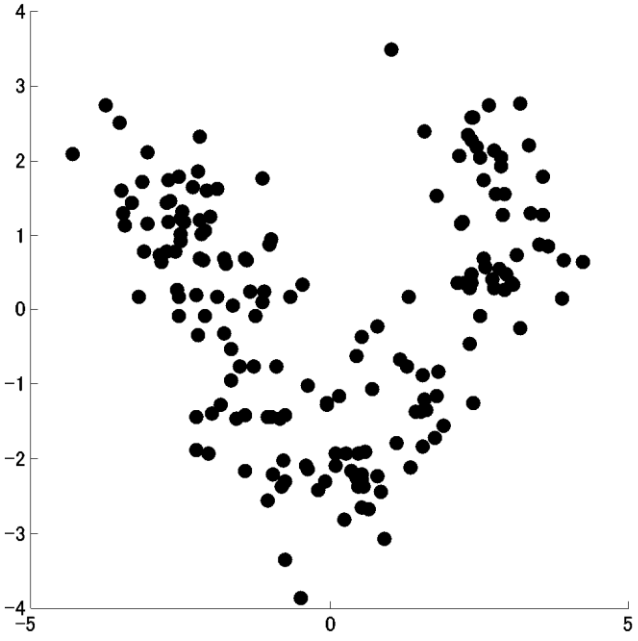

Surface
Linear subspace
Feature map
Original data space
Feature space
(RKHS

Example

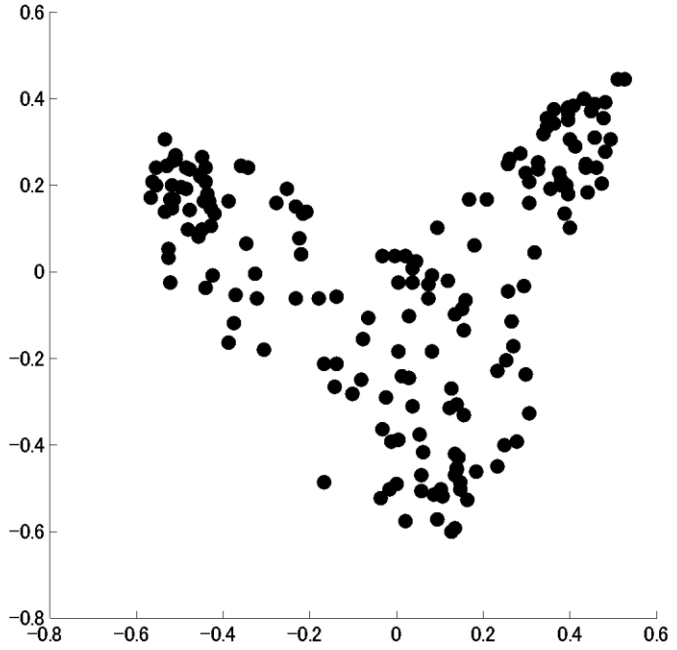
- Wine data (UCI repository)

  13 chemical measurements of three types of wine. 178.
  (The three classes are not used for KPCA)

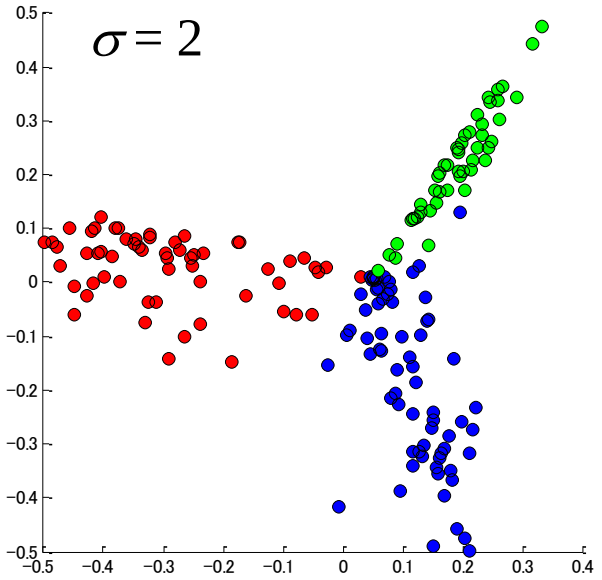
Linear PCA

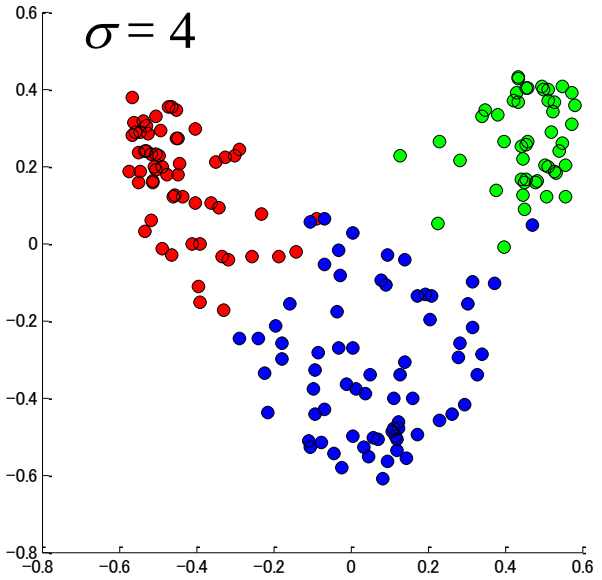


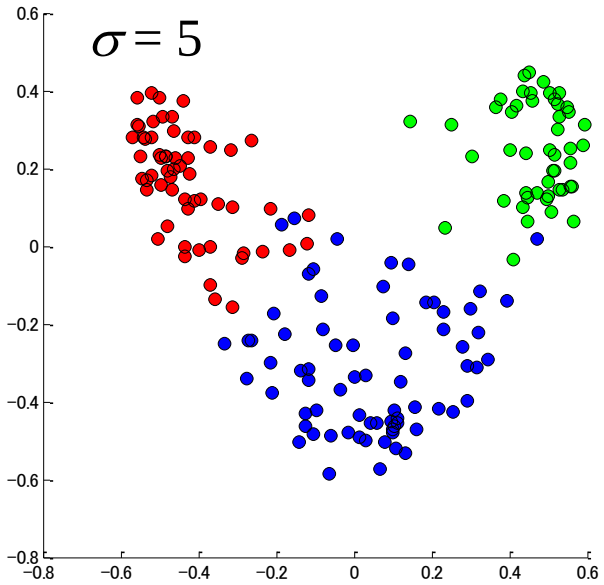
Kernel PCA (Gaussian kernel)



Kernel PCA (Gaussian)







$$k_G(x, y) = \exp\left(-\|x - y\|^2 / \sigma^2\right)$$

Common properties of kernel methods

- Linear method is applied on the feature space. [Kernelization]
- Typically, only the inner products

$$\langle \Phi(X^{(i)}), \Phi(X^{(j)}) \rangle = k(X^{(i)}, X^{(j)})$$ [Kernel trick]

  are used for writing the objective function of the method.
- The solution is almost always given in the form

$$f = \sum_{i=1}^{n} c_i \Phi(X_i)$$ [Representer theorem]

  → Everything is written by Gram matrices (size = data size).
- Original space does not appear. Non-vectorial data can be handled.

Choice of kernel

Choice of kernel (or parameter of kernel) is important for good performance.

e.g.) Gaussian kernel: σ in $k_G(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$

- Recipes:
  - Supervised learning (e.g. KRR, SVM)

    → Cross-validation

  - Unsupervised learning (e.g. KPCA)
    - No general methods
    - A final goal may provide a method.

      e.g. KPCA as a preprocessing → Final regression performance

- Use Bayesian framework (marginal likelihood)
- Based on the background knowledge: roughness, frequency, etc.
- Multiple Kernel Learning

  Leaning of a convex combination of kernels

$$K_c(x, y) = c_1 k_1(x, y) + \cdots + c_M k_M(x, y)$$

See a review paper: Gönen, M. and E. Alpaydın (JMLR 2011)

3. Brief introduction to Support Vector Machine

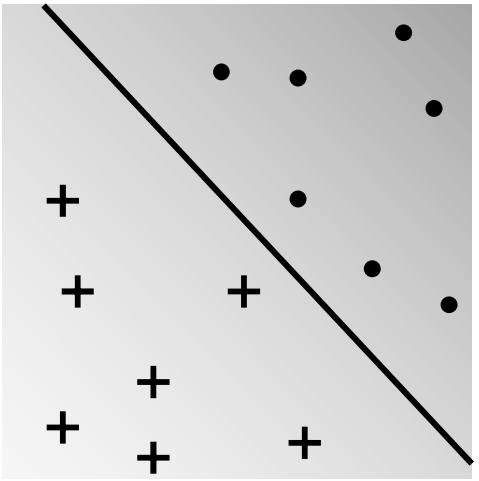
Binary classification

- Training data

Input data

Class label

$$\mathbf{X} = \begin{pmatrix} X_1^{(1)} & \cdots & X_m^{(1)} \\ \vdots & \ddots & \vdots \\ X_1^{(N)} & \cdots & X_m^{(N)} \end{pmatrix} \qquad Y = \begin{pmatrix} Y^{(1)} \\ \vdots \\ Y^{(N)} \end{pmatrix} \in \{\pm 1\}^N$$



- Linear classification

  Classifier $h_{w,b}(x) = \mathrm{sgn}(w^T x + b)$

  Objective $h_{w,b}(X^{(i)}) = Y^{(i)}$ for all (or most) i.

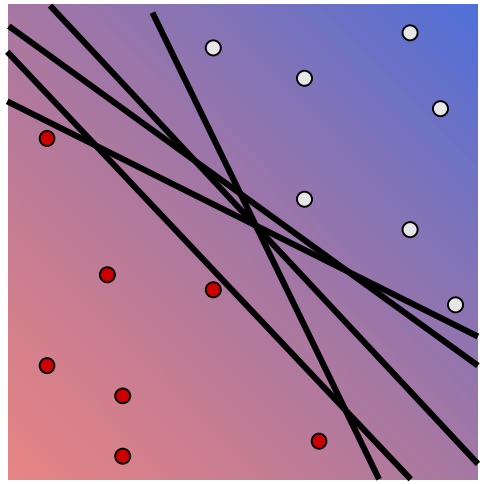
Large margin classifier

- Assumption: Linearly separable

  There exist w, b such that

  $$\operatorname{sgn}\left(w^T X^{(i)} + b\right) = Y^{(i)} \qquad \forall i.$$

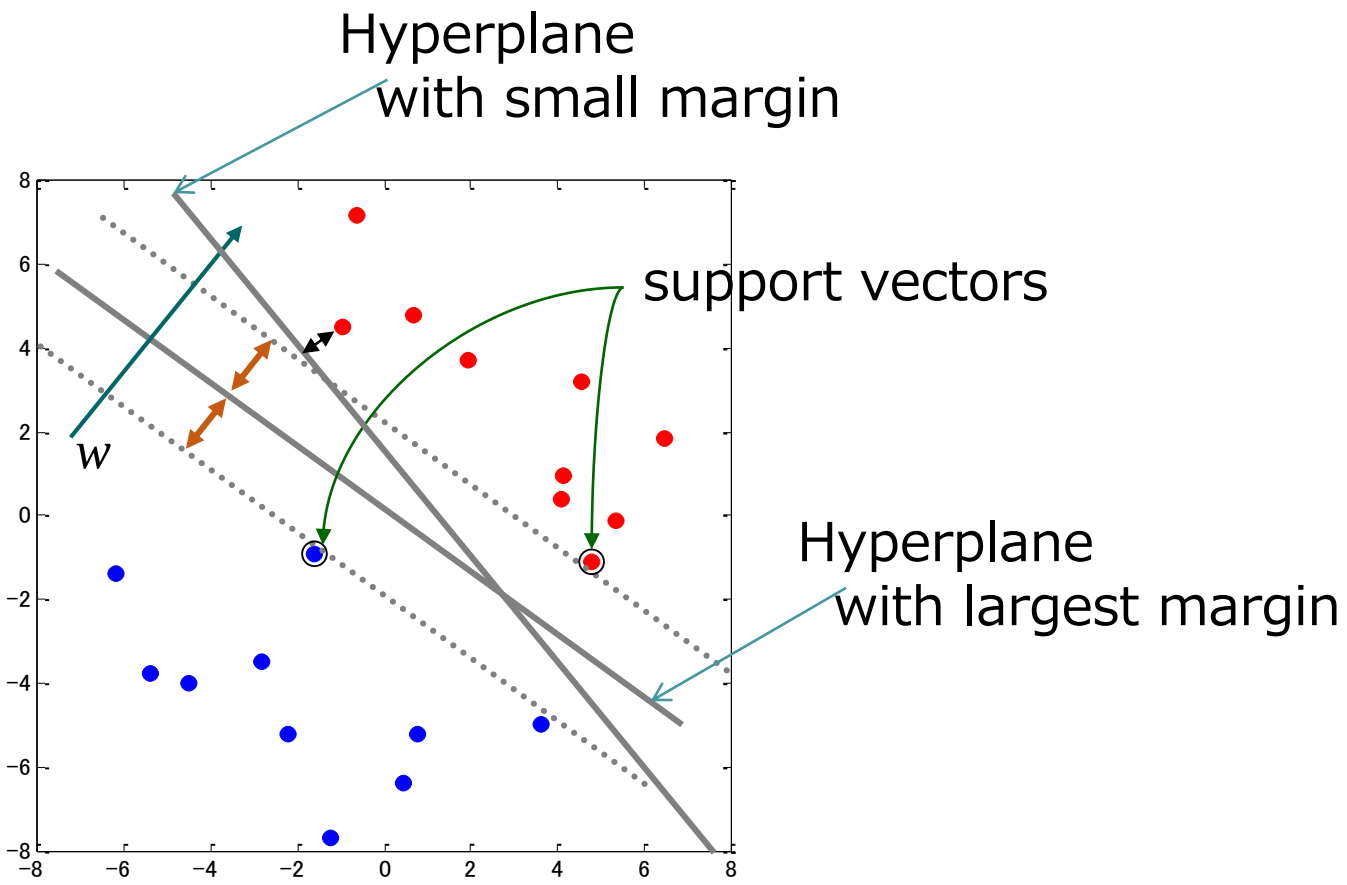
  The solutions are infinite.
  Which is the best?



- Large margin criterion:

  Select the one to give the largest margin.

  - Margin = distance of the two classes in the direction of w.
  - The very middle of the two planes.


Hyperplane
with small margin
support vectors
w
Hyperplane
with largest margin
8
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
6
8

- Measuring the margin

  To fix a scale, assume

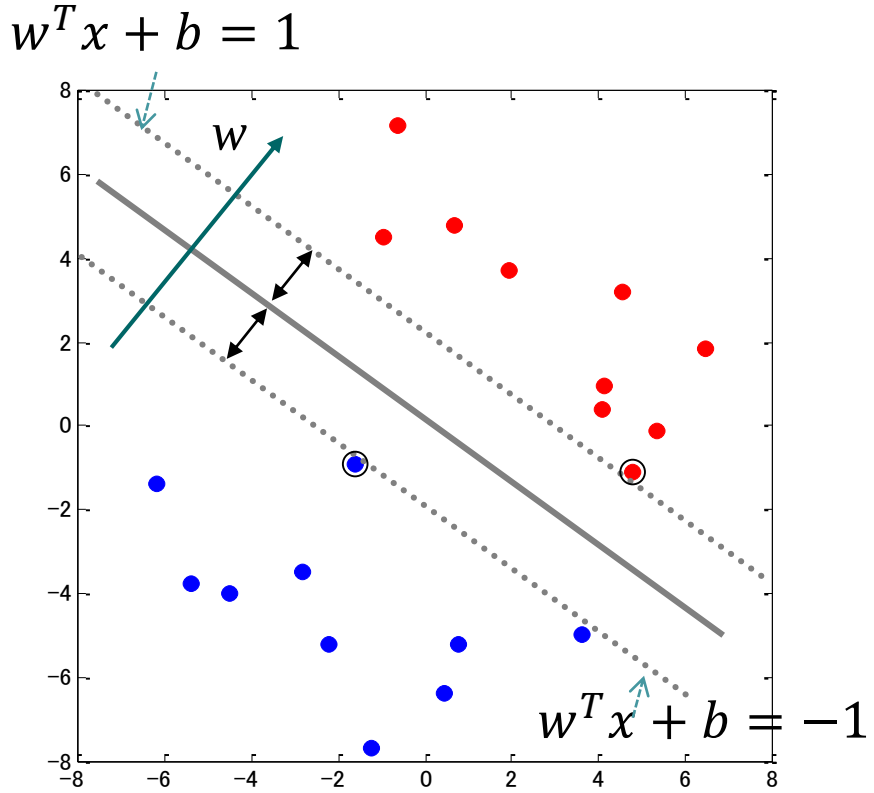
  $\min(w^T X^{(i)} + b) = 1$ for the closest point with $Y^{(i)} = +1$,

  $\min(w^T X^{(i)} + b) = -1$ for the closest point with $Y^{(i)} = -1$.

Then,

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

[Exercise: prove it]



- Max margin classifier

$$\max \frac{1}{\|w\|} \quad \text{subject to} \quad \begin{cases} w^T X^{(i)} + b \geq 1 & \text{if } Y^{(i)} = +1 \\ w^T X^{(i)} + b \leq -1 & \text{if } Y^{(i)} = -1 \end{cases}$$

Equivalently,

$$\min_{w,b} \|w\|^2 \quad \text{subject to} \quad Y^{(i)}\left(w^T X^{(i)} + b\right) \geq 1 \ (\forall\, i).$$

Linear SVM (hard margin)

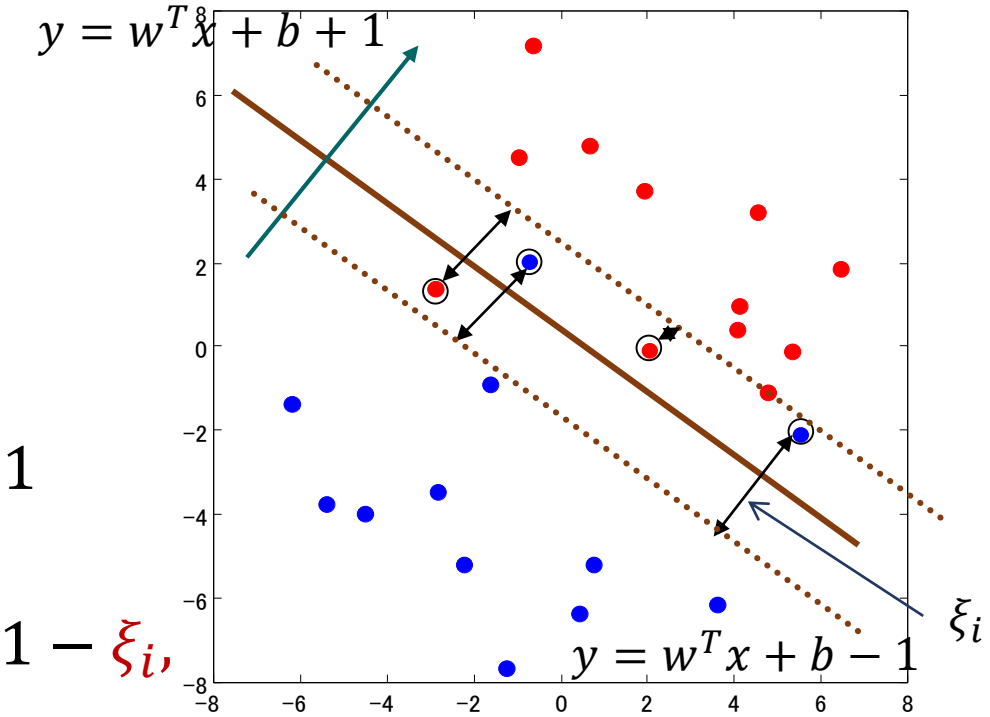
- Quadratic Program (QP): Min of a quadratic function with linear inequalities.
  → Convex. No local minima!
- QP solver is provided in many software libraries.
- SMO (Sequential Minimal Optimization), a simpler algorithm for SVM, is more popular.

Soft margin SVM

- Linear separability is impractical
  → Relax it.

Hard constraints: $Y^{(i)}\left(w^T X^{(i)} + b\right) \geq 1$

Soft constraints: $Y^{(i)}\left(w^T X^{(i)} + b\right) \geq 1 - \xi_i,$
$\xi_i \geq 0.$



Max margin classifier (soft margin)

$$\min_{w,b,\xi} \|w\|^2 + C\sum_i \xi_i \quad \text{subj. to} \quad Y^{(i)}\left(w^T X^{(i)} + b\right) \geq 1 - \xi_i \ (\forall\, i),$$
$$\xi_i \geq 0 \qquad (\forall\, i).$$

Nonlinear SVM

- Kernelization of linear SVM
  - $(X^{(1)}, Y^{(1)}), \ldots, (X^{(N)}, Y^{(N)})$: training data
    - $X^{(i)}$: element in a set S. (non-vectorial data is allowed)
    - $Y^{(i)} \in \{+1, -1\}$
  - k: positive definite kernel on S. H: RKHS given by k.
  - $\Phi(X^{(i)}) = k(\cdot, X^{(i)})$: feature vector.
  - Linear classifier on H:

$$f(x) = \mathrm{sgn}(\langle h, \Phi(x) \rangle_H + b) = \mathrm{sgn}(h(x) + b)$$

c.f. $f(x) = \mathrm{sgn}(w^T x + b)$

- SVM （Kernelized）
  - Objective function

$$\min_{h,b,\xi_i} \|h\|_H^2 + C \sum_i \xi_i \quad \text{subj. to} \quad Y^{(i)}\left(h(X^{(i)}) + b\right) \geq 1 - \xi_i \quad (\forall i)$$
$$\xi_i \geq 0.$$

Nonlinear SVM （soft margin）

$$\min_{c,b,\xi} \sum_i c_i c_j \, k(X^{(i)}, X^{(j)}) + C \sum_i \xi_i$$
$$\text{subj. to} \quad Y^{(i)}\left(\sum_j c_j \, k(X^{(i)}, X^{(j)}) + b\right) \geq 1 - \xi_i \quad (\forall i)$$
$$\xi_i \geq 0.$$

  - QP.
  - Dual problem is easier to solve. (Omitted in this lecture).
    → Solution is sparse → support vectors.
  - Hyperparaemters (C, kernel parameter) are chosen by cross-validation.

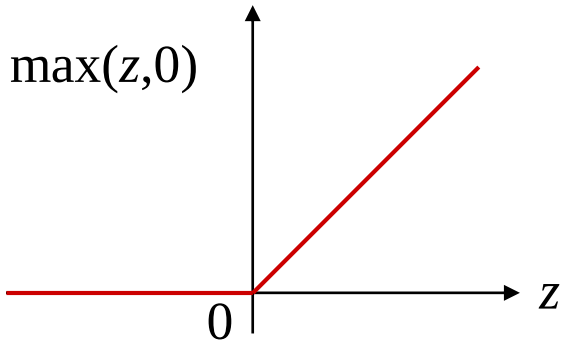
SVM and regularization

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

$$\min_{w,b} \sum_i \left(1 - Y^{(i)}\left(h(X^{(i)}) + b\right)\right)_+ + \lambda \|h\|_H^2$$

where

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



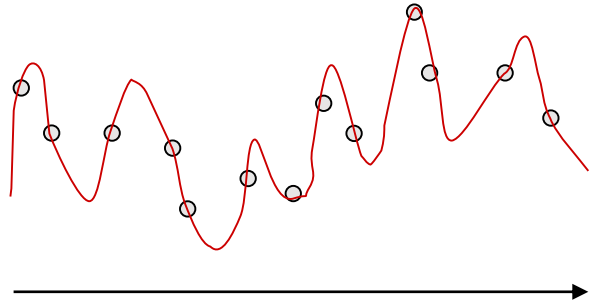
Hinge loss

Regularization

- Ill-posed problem

$$\min_f \sum_i \left(Y^{(i)} - f\left(X^{(i)}\right)\right)^2$$

If f is arbitrary, many functions f achieve zero error.

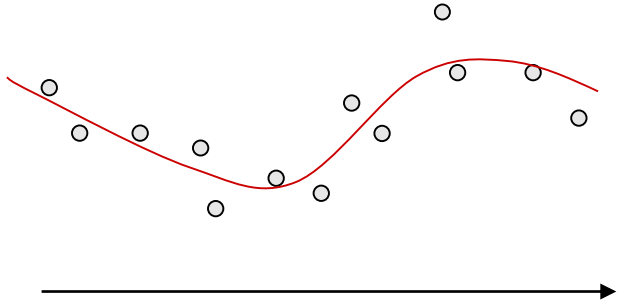


- Regularization

$$\min_f \sum_i \left(Y^{(i)} - f\left(X^{(i)}\right)\right)^2 + \lambda \|f\|_H{}^2$$

The solution is unique.

See Kernel ridge regression.



Demonstration of SVM

Many public software library.

libSVM etc

https://www.csie.ntu.edu.tw/~cjlin/libsvm/

- JavaScript

  http://cs.stanford.edu/people/karpathy/svmjs/demo/

Comparison: SVM vs Deep Learning

SVM

- Easy to use. A few hyperparameters to tune.
- Need techniques for very large data.
- Easy for structured data, e.g. graph, strings, probability measures.
- Theoretical guarantee.

DNN

- State-of-the-art performance for image classification.
- Can be used for very large data.
- Many to tune for the architecture and learning.
- Computation is heavy (GPU).

- Example of performance comparison:

  Improved support vector machine classification algorithm based on adaptive feature weight updating in the Hadoop cluster environment. By Jianfang Cao, Min Wang, Yanfei Li, Qi Zhang. *PLoS One* 2019.

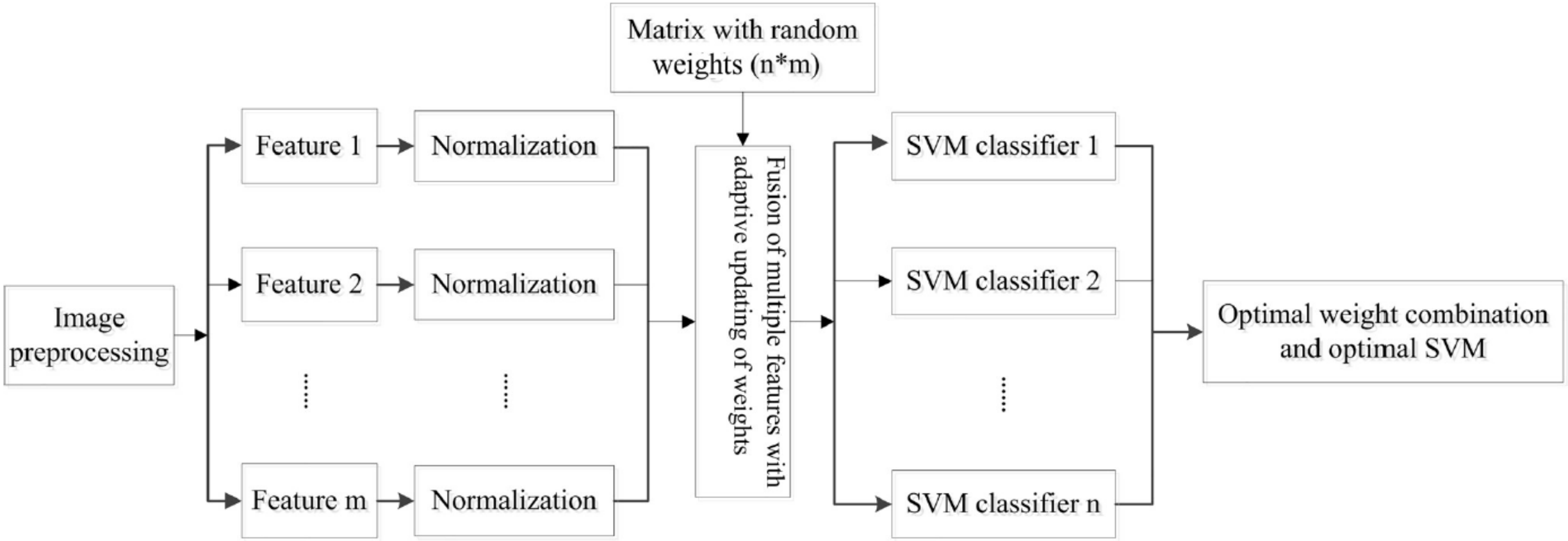


Table 2. Comprehensive comparison between the algorithm proposed in this paper and other deep learning algorithms.

Number of images	Index	Method						
		CNN	AlexNet	VGGNet	GoogleNet	ResNet	DenseNet	Method of this paper
5,000	Training time (S)	604	621	619	635	640	643	1.2
	Classification accuracy (%)	97.93	97.92	97.93	97.93	97.92	97.93	97.92
10,000	Training time (S)	1080	1090	1085	1120	1103	1032	50
	Classification accuracy (%)	96.81	96.81	96.82	96.83	96.83	96.84	96.83
25,000	Training time (S)	2700	2702	2765	2748	2801	2792	296
	Classification accuracy (%)	96.18	96.17	96.18	96.18	96.18	96.18	96.17
50,000	Training time (S)	5310	5401	5400	5376	5412	5400	1050
	Classification accuracy (%)	95.27	95.27	95.27	95.28	95.30	95.30	95.28
80,000	Training time (S)	10245	10283	10300	10299	10307	10308	1458
	Classification accuracy (%)	95.15	95.15	95.16	95.15	95.16	95.18	95.14

Data: ImageNet

- For small data (~10^4), SVM can show comparable performance

Summary of SVM

- Large margin criterion

  May not be the least error, but causes other good properties.

- Kernel can be used Nonlinear classifier is possible.

- Quadratic programming:

  The objective function is solved by the standard QP. No local minima.

- Sparse representation:

  The classifier is represented by a small number of support vectors
  (not explained in this lecture, because optimization theory is needed)

$$f(x) = \sum_{i:\text{support vector}} c_i k(x, x^{(i)}) + b$$

- Theoretical guarantee

  Theoretic guarantee on generalization error is possible by the statistical learning theory.

4. Approximation for scalability

Computation of Gram matrices

- Computation in kernel methods: linear algebra with Gram matrices

Matrix size = sample size n

$$\begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix}$$

- The dimensionality of the original space is not (less) problematic

- Computational problem for large data size n → Big Data?
  Matrix inversion, eigendecomposition: $O(n^3)$ in time, prohibitive

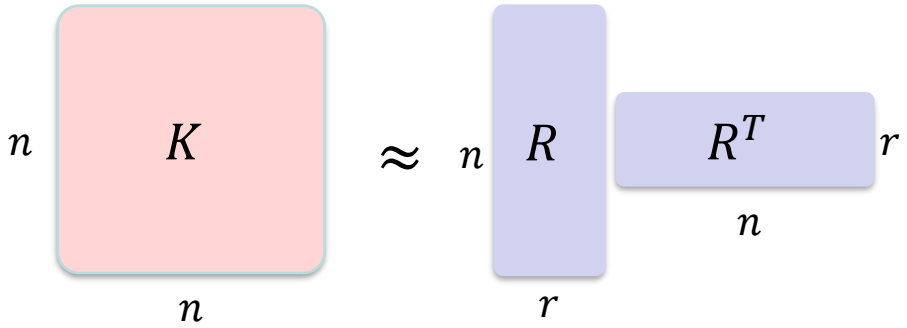
- Handling large data is important c.f. deep learning

- Approaches to approximation
  - Low rank approximation of Gram matrix
    - Incomplete Cholesky factorization (Fine and Scheinberg 2001)
    - Nyström approximation (Williams and Seeger 2001; Drineas & Mahoney 2005)
  - Random features
    - Random Fourier features (Random kitchen sink)

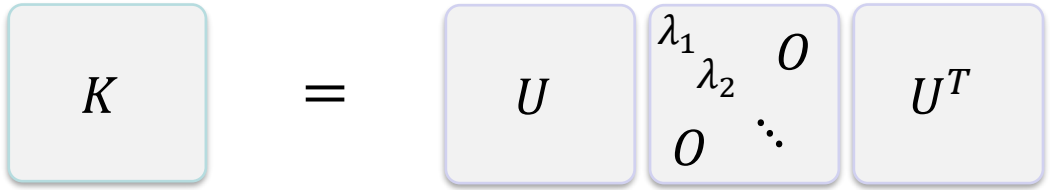
Efficient computation of specific algorithms are also studied.

Low rank approximation

- $K \approx RR^T$, R: $n \times r$ matrix $(r \ll n)$



c.f. eigendecomposition

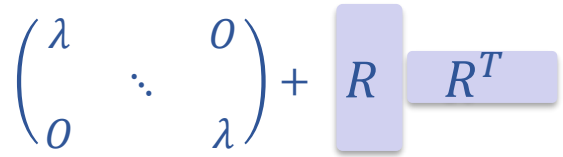


- Rank r can be small.
  Decay of eigenvalues is fast in many typical examples.
  (Widom 1963, 1964; Bach & Jordan 2002).

- Effect of low rank approximation
  - E.g. Kernel ridge regressoin

$$f(x) = Y^T(K_X + \lambda I_n)^{-1}\mathbf{k}(x) \qquad \text{time} : O(n^3)$$

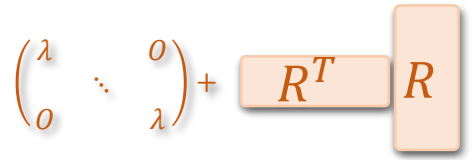
Low rank approx. → $K_X \approx RR^T$.



By Woodbury formula

$$Y^T(K_X + \lambda I_n)^{-1}\mathbf{k}(x) \approx Y^T(\underbrace{RR^T}_{n\times n} + \lambda I_n)^{-1}\mathbf{k}(x)$$

$$= \frac{1}{\lambda}\{Y^T\mathbf{k}(x) - Y^T R(\underbrace{R^T R}_{r\times r} + \lambda I_r)^{-1}R^T\mathbf{k}(x)\} \qquad \text{time} : O(r^2 n + r^3)$$



Theorem Woodbury (Sherman–Morrison–Woodbury)

$A: n \times n$ invertible matrix, $U: n \times r$, $V: r \times n$

$$(A + UV)^{-1} = A^{-1} - A^{-1}U(I_r + VA^{-1}U)^{-1}VA^{-1}.$$

[Exercise: Confirm this formula]

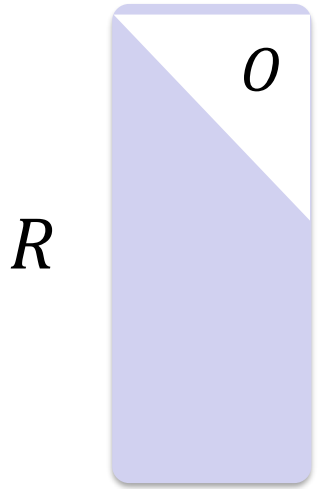
Methods of low rank approximation

- Incomplete Cholesky factorization (Fine and Scheinberg 2001)

  K ($n \times n$ positive semi-defnite)

  $$K \approx RR^T \qquad R\text{: lower triangular}$$

  - Gaussian elimination to the colums.
  - Time $O(nr^2)$ Memory $O(nr)$



Remark) Input to the algorithm is n data (x_i), not $n \times n$ matrix K.

- Nyström approximation (Williams and Seeger 2001; Drineas & Mahoney 2005)

$$K \approx C_p W_r^{\dagger} C_p^T$$

C_p: random sample of p columns ($p > r$).
W_r: rank-r approximation of $K(\mathbf{X}_p, \mathbf{X}_p)$ ← by eigen-decomposition

- Sampling of columns: Uniform or leverage score. (Aloui&Mahoney 2015)

- Time $O(p^3 + prn)$ Memory $O(np)$

Random Fourier Feature (Rahimi & Recht 2008)

- Target: very large data (10^5-)
- Basis: Bochner's theorem (see two slides later)

$$k(x,y) = \int \exp\left(\sqrt{-1}\omega^T(x-y)\right) d\Lambda(\omega), \quad \Lambda\text{: probability measure}$$

$\omega_1, \ldots, \omega_L \sim \Lambda$, i.i.d. Monte Carlo sampling

Approximation

$$\Rightarrow \quad k(x,y) \approx \frac{1}{L}\sum_{\ell=1}^{L} \exp\left(\sqrt{-1}\omega_\ell^T(x-y)\right)$$

For real valued kernel,

$$k(x,y) \approx \frac{1}{L}\sum_{\ell=1}^{L} \cos\left(\omega_\ell^T(x-y)\right)$$

$$= \frac{1}{L}\sum_{\ell=1}^{L} \cos\left(\omega_\ell^T x\right)\cos\left(\omega_\ell^T y\right) + \sin\left(\omega_\ell^T x\right)\sin\left(\omega_\ell^T y\right)$$

Define $Z(x) \coloneqq \frac{1}{\sqrt{L}}(\cos(\omega_1^T x), \ldots, \cos(\omega_L^T x), \sin(\omega_1^T x), \ldots, \sin(\omega_L^T x))^T$

$$k(x, y) \approx Z^T(x)Z(y)$$

- Example: RFF applied to kernel ridge regression.

KRR $\min_{f \in H_k} \sum_{i=1}^n \big(Y_i - f(X_i)\big)^2 + \lambda \|f\|_{H_k}^2$

Inserting $f(x) = \sum_{j=1}^n \alpha_j k(x, X_j)$,

$$\min_{\alpha \in \mathbf{R}^n} \sum_{i=1}^n \Big(Y_i - \sum_{j=1}^n \alpha_j k(X_i, X_j)\Big)^2 + \lambda \sum_{i,j=1}^n \alpha_i \alpha_j k(X_i, X_j)$$

- RFF: $k(X_i, X_j) \approx \sum_{a=1}^M Z_a(X_i) Z_a(X_j)$

Define $c_a \coloneqq \sum_{i=1}^n \alpha_i Z_a(X_i)$, then $f(x) = \sum_{a=1}^{2L} c_a Z_a(x)$ and

$$\min_{c \in \mathbf{R}^{2L}} (Y_i - \sum_a^{2L} c_a Z_a(X_i))^2 + \lambda \|c\|^2$$ Ridge regression with bases.

Fourier analysis of shift-invariant kernels

- Bochner's theorem (Fourier transform of shift-inv. kernel)

  $k(x, y)$: continuous, shift invariance kernel on $\mathbf{R}^m$

$$k(x, y) = \int \exp\left(\sqrt{-1}\omega^T(x - y)\right) d\Lambda(\omega)$$

  $\exists\ \Lambda$: finite nonnegative measure.

  Λ: Fourier transform of k

→ Normalize so that $\int d\Lambda(\omega) = 1$ → Probability measure.

E.g. Gaussian kernel $\exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$

→ Gaussian distribution $\exp\left(-\frac{\sigma^2}{2}\|\omega\|^2\right)$

Laplace kernel $\exp(-\alpha|x - y|)$ → Cauchy distribution $\frac{1}{\pi(\alpha^2+\omega^2)}$

Addendum: Fourier expression of RKHS

- Shift invariant kernel on $\mathbf{R}^d$

$$k(x,y) = \int \exp\left(\sqrt{-1}\omega^T(x-y)\right)\rho(\omega)d\omega\,. \qquad \text{[Bochner]}$$

$\rho(\omega)$: continuous non-negative function on $\mathbf{R}^d$.

Then, $H_k = \left\{ f \in L^2(\mathbf{R}, dx) \;\middle|\; \int \frac{|\hat{f}(\omega)|^2}{\rho(\omega)} d\omega < \infty \right\}$, $\langle f, g \rangle_{H_k} = \int \frac{\hat{f}(\omega)\overline{\hat{g}(\omega)}}{\rho(\omega)} d\omega$

where $\hat{f}$ is the Fourier transform of f : $\hat{f}(\omega) = \frac{1}{(2\pi)^m} \int f(x) \mathrm{e}^{-\sqrt{-1}\omega^T x}\, d\omega$.

[Exercise: prove the reproducing property]

- The RKHS norm $\|f\|_H$ is related to frequency property or smoothness.

$$\|f\|^2_{H_k} = \int \frac{|\hat{f}(\omega)|^2}{\rho(\omega)} d\omega$$

- Gaussian kernel

$$k_G(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \qquad \rho_G(\omega) = \frac{1}{(2\pi)^m}\exp\left(-\frac{\sigma^2\|\omega\|^2}{2}\right)$$

$$H_{k_G} = \left\{ f \in L^2(\mathbf{R}, dx) \;\middle|\; \int \left|\hat{f}(\omega)\right|^2 \exp\left(\frac{\sigma^2\|\omega\|^2}{2}\right) d\omega < \infty \right\}$$

$$\langle f, g\rangle = (2\pi)^m \int \hat{f}(\omega)\overline{\hat{g}(\omega)} \exp\left(\frac{\sigma^2\|\omega\|^2}{2}\right) d\omega$$

- Laplace kernel on $\mathbf{R}$

$$k_L(x,y) = \exp(-\beta|x-y|), \qquad \rho_L(\omega) = \frac{1}{2\pi(\omega^2+\beta^2)}$$

$$H_{k_L} = \left\{ f \in L^2(\mathbf{R}, dx) \;\middle|\; \int \left|\hat{f}(\omega)\right|^2 (\omega^2+\beta^2) d\omega < \infty \right\}$$

$$\langle f, g\rangle = 2\pi \int \hat{f}(\omega)\overline{\hat{g}(\omega)}(\omega^2+\beta^2) d\omega$$

Remarks on approximation methods

- Comparisons
  - Nyström/iCholesky depend on data. Approximation of Gram matrix
  - Random Fourier feature does NOT use data.

    Approximation of kernel function $k(x, y)$
  - RFF can be reused once an approximation (generation of ω_ℓ) is given.

- Influence of approximation.
  - How does the approximation influence on the performance?
  - E.g. KRR + Nyström / RFF

    → Generalization error converges in the same rate ($n \to \infty$) as non-approximated method by sufficiently large r.

    (Rudi et al NIPS 2015; NIPS 2017)

Large data: Doubly Stochastic Gradient
(Dai et al NIPS 2014)

- Scalable kernel method (#data > 1M) c.f. deep learning

$$\min_f E[R(f;Y,X)], \quad R(f;X,Y) \coloneqq \ell\big(Y, f(X)\big) + \frac{\nu}{2}\|f\|_{H_k}^2$$

  - Stochastic Gradient Descent on RKHS

$$\nabla_f R(f_t;X_t,Y_t) = \ell'\big(Y_t, f(X_t)\big)k(\cdot,X_t) + \nu f$$

Note: $f(X_t) = \langle f, k(\cdot,X_t)\rangle_{H_k}$

  - Random Fourier feature with one sample

$$\omega_t \sim \Lambda$$

$$\tilde{\nabla}_f R(f_t;X_t,Y_t) \coloneqq \ell'\big(Y_t, f_t(X_t)\big)\phi_{\omega_t}(\cdot)\phi_{\omega_t}(X_t) + \nu f_t$$

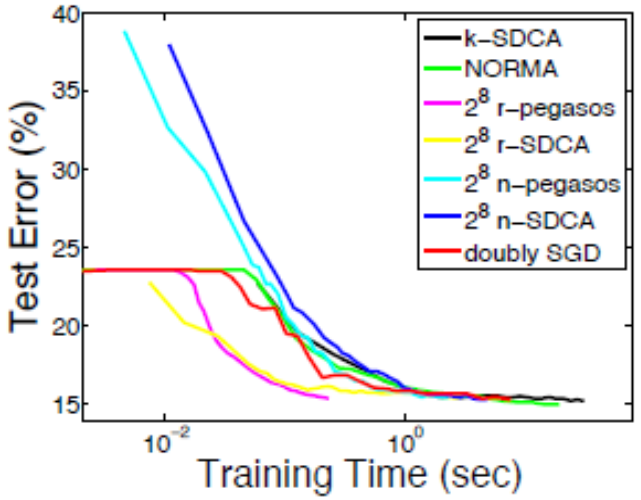
$$E\big[\tilde{\nabla}_f R(f_t;X_t,Y_t)\big] = E\big[\ell'\big(Y, f(X)\big)k(\cdot,X)\big]$$

$$f_{t+1}(\cdot) = f_t(\cdot) - \gamma_t \tilde{\nabla}_f R(f_t;X_t,Y_t)$$

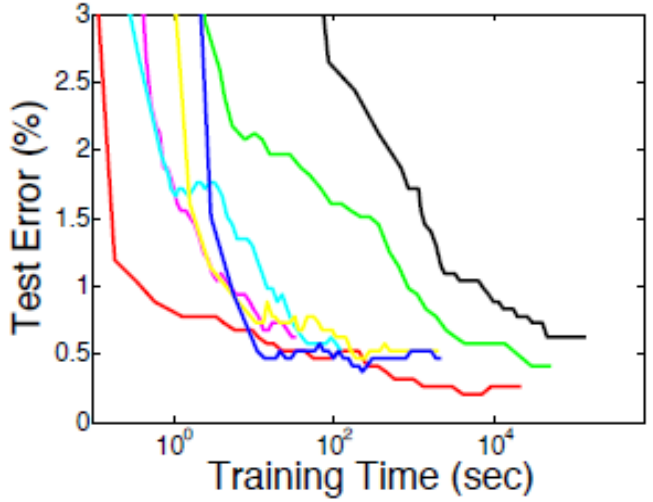
$\phi_{\omega_t}(\cdot)$ is added as a new basis function one by one.

Comparisons with other SVM solvers (binary classification)

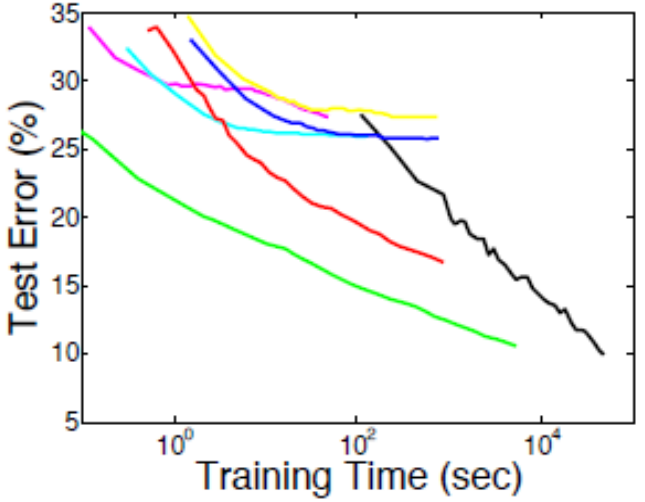
SC1: 1 Epoch
SC2: same computational cost.



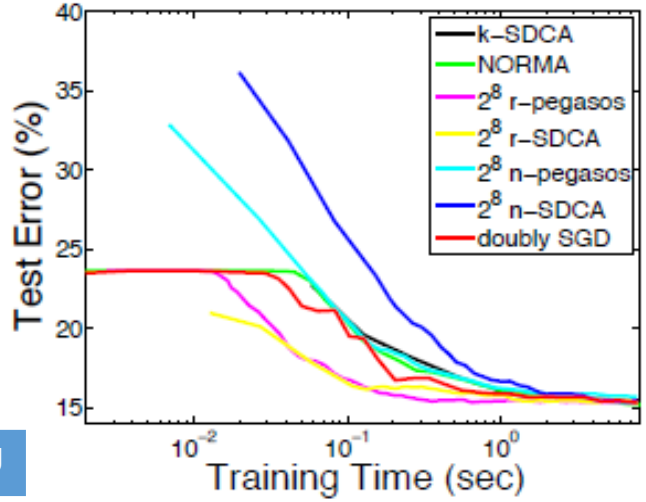
SC1: (1) Adult



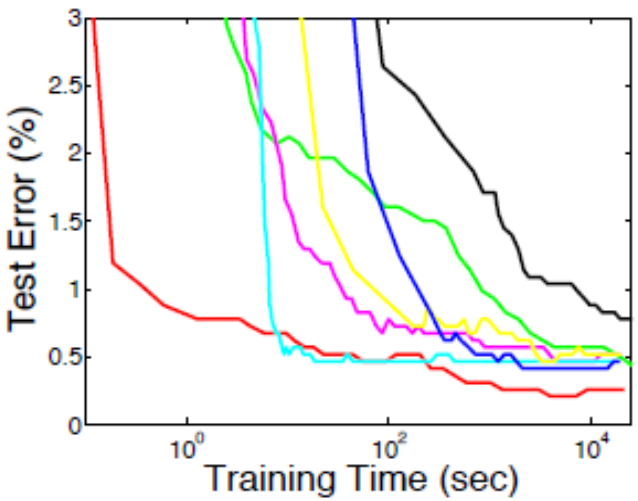
(2) MNIST 8M 8 vs. 6



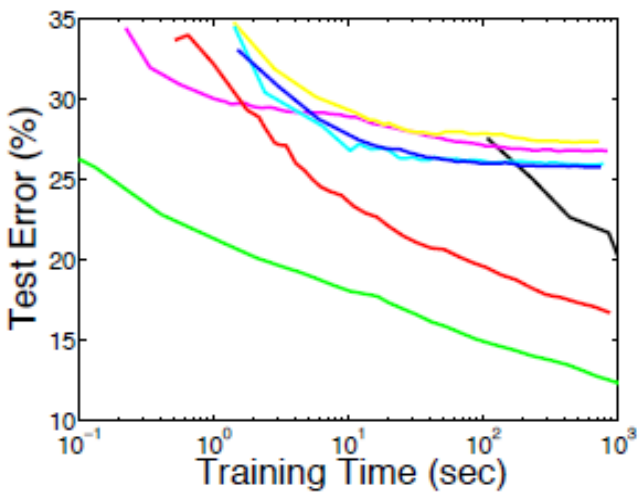
(3) Forest



SC2: (4) Adult



(5)MNIST 8M 8 vs. 6



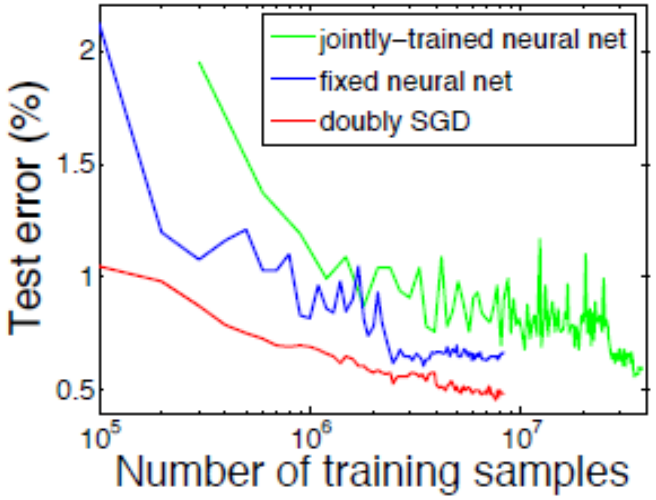
(6) Forest.

Data set	#Training
Adult	32K
MNIST 8vs 6	1.6M
Forest	0.5M

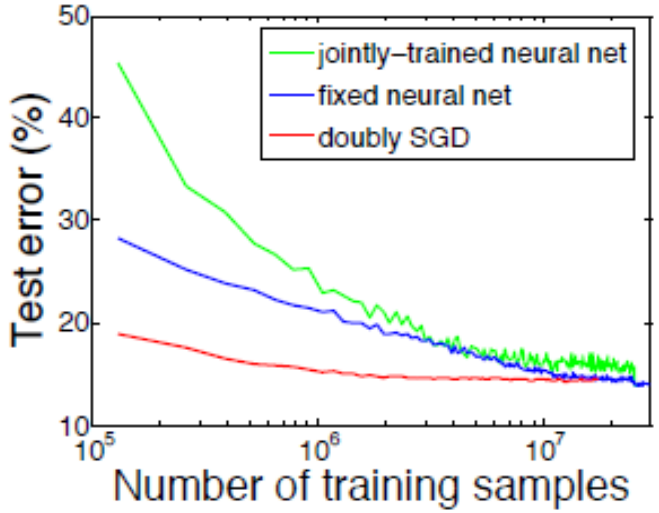
Dai et al 2015 arXiv

Comparisons with neural networks（Dai et al 2015 arXiv)

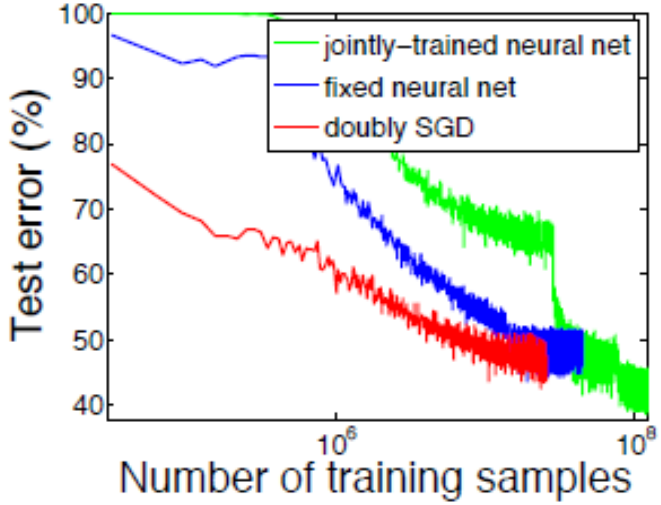
Classification



(1) MNIST 8M



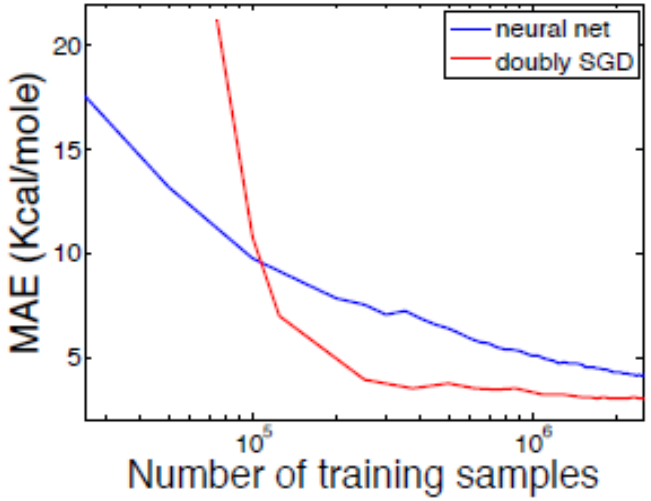
(2) CIFAR 10



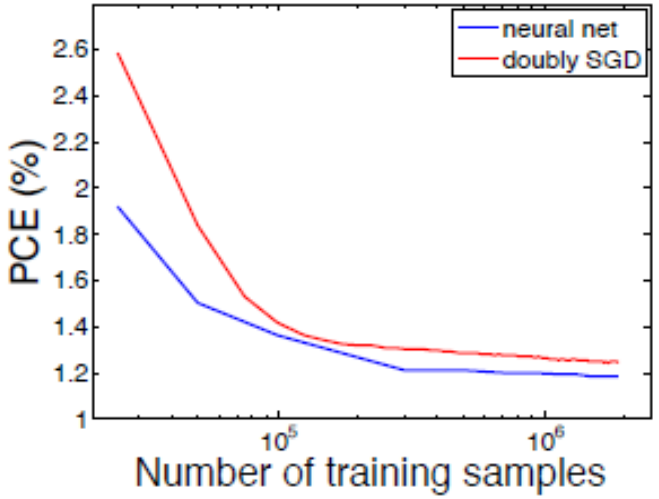
(3) ImageNet（Alex net）

Regression

Data sets	#training data	#classes
MNIST 8M	8M	10
CIFAR 10	50K	10
ImageNet	1.3M	1000
QuantumM	6K	**R**
Molecul.	2.3M	**R**



(4) QuantumMachine



(5) MolecularSpace.

6. Non-vectorial data

Structured data

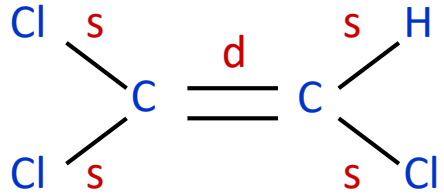
- Structured data: non-vectorial data with some structure.
  - String / sequence data (variable length):
    DNA sequence, Protein (sequence of amino acids)

    ATCATGCAATACC⋯..

    Text (sequence of words)

    A man sits alone in a cave. His hair is long. ⋯

  - Graph data
    Chemical compounds can be represented by labeled graphs.



E.g. Mapping by SVM: Chemical compound → toxic / non toxic

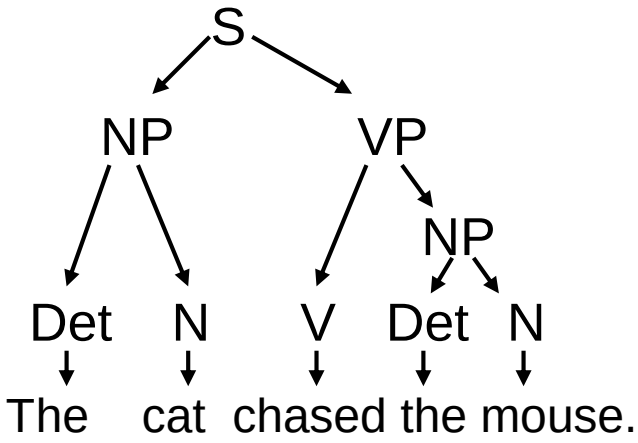
- Tree data

  Parse tree in natural language processing

  E.g. learn mapping

  Sentence → Parse tree

  (Collins & Duffty, NIPS 2002)



- Kernel methods can be applied to any type of data, once a kernel is defined.

- Many kernels uses counts of substructures (Haussler 1999).

Example: spectrum kernel

- *p*-spectrum kernel (Leslie et al 2002): positive definite kernel for string.

  $k_p(s,t) =$ Occurrences of common subsequences of length p.
  Measuring the similarity of two words.

- Example: s = "statistics" t = "pastapistan"
  3-spectrum

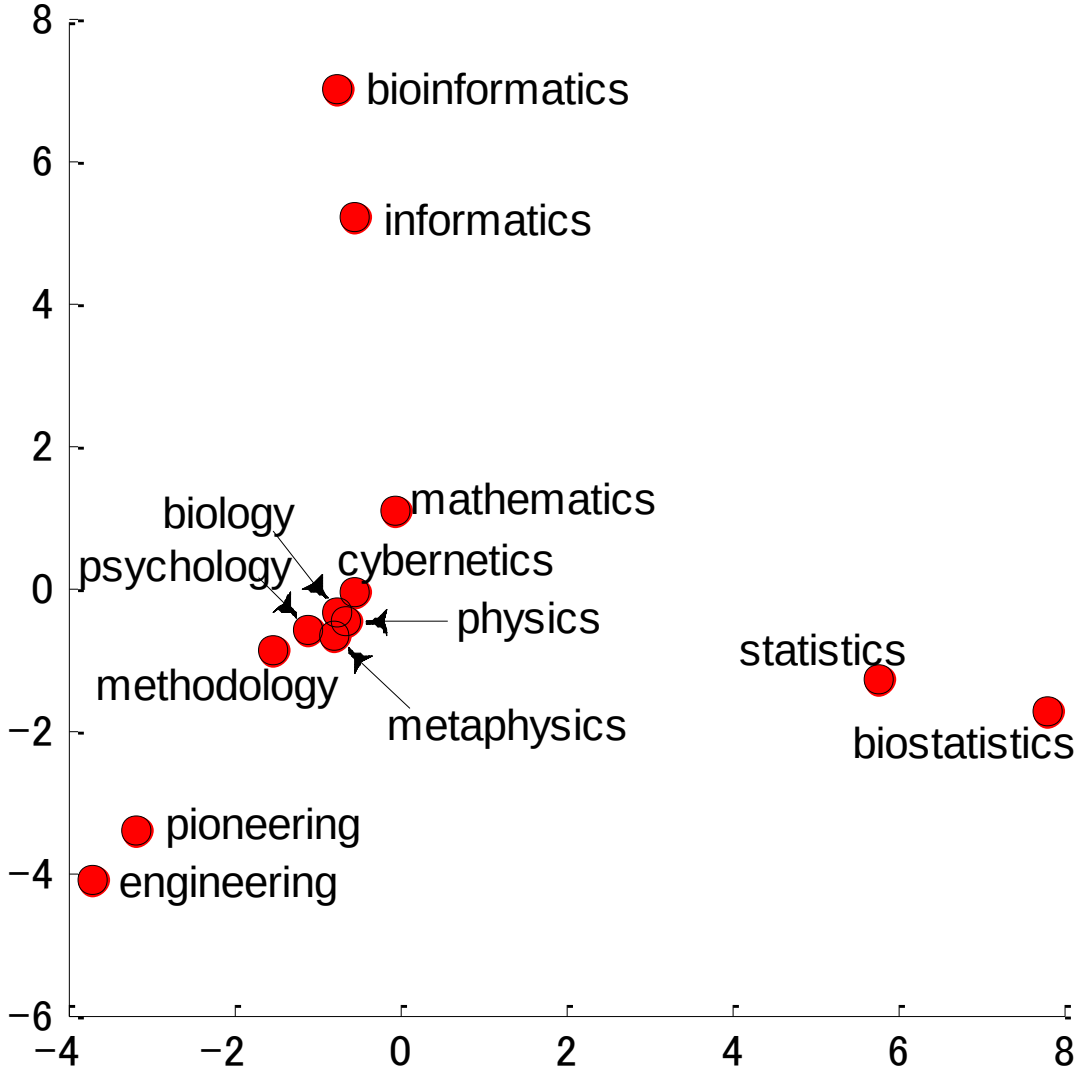
  s: sta, tat, ati, tis, ist, sti, tic, ics
  t: pas, ast, sta, tap, api, pis, ist, sta, tan

$K_3(s,t) = 1 \cdot 2 + 1 \cdot 1 = 3$

	sta	tat	ati	tis	ist	sti	tic	ics	pas	ast	tap	api	pis	tan
Φ(s)	1	1	1	1	1	1	1	1	0	0	0	0	0	0
Φ(t)	2	0	0	0	1	0	0	0	1	1	1	1	1	1

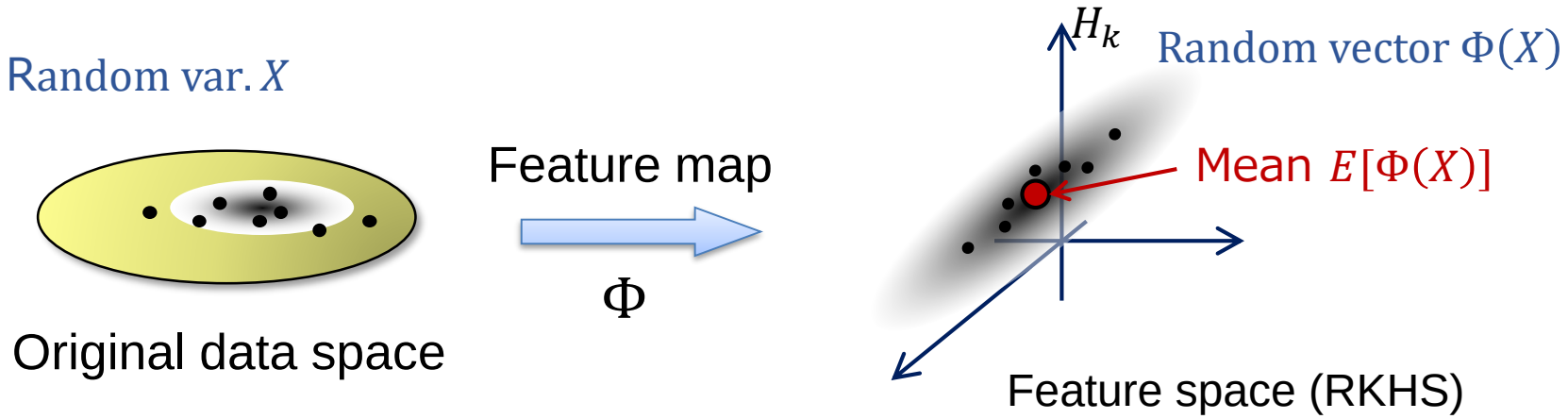
- Linear time ($O(p(|S| + |t|)$) algorithm with suffix tree is known.

- Application: 2-D plots given by kernel PCA of ‘words’ with 3-spectrum kernel



Kernel methods for probabilities

- Statistical methods: inference of probability distributions
- Kernel mean embedding



The mean $E[\Phi(X)]$ of the random feature vector $\Phi(X)$ is used for representing the probability of X

- Kernel mean embedding

$$m_X \coloneqq E[\Phi(X)] = E[k(\cdot, X)] = \int k(\cdot, x) dP(x)$$

  - Example: Gaussian kernel $\quad m_P(y) = \int \exp\left(-\frac{\|y-x\|^2}{2\sigma^2}\right) p(x)dx$ [Convolution]

  - Reproducing property

$$\langle f, m_P \rangle = E[f(X)] \qquad \text{for any } f \in H_k$$

  - Empirical estimator

$$\hat{m}_X \coloneqq \frac{1}{n}\sum_{i=1}^{n} \Phi(X_i) = \frac{1}{n}\sum_{i=1}^{n} k(\cdot, X_i)$$

  - Consistency: $\|\hat{m}_X - m_X\|_H \to 0 \quad (n \to \infty)$ for i.i.d. (X_i)

- Why does it represent a probability?
  - RKHS is infinite dimensional → infinitely many statistics
  - Higher order moments are involved.

    Example:

    Taylor series $k(u,x) = c_0 + c_1 ux + c_2 (ux)^2 + \cdots \quad (c_i > 0)$,

    e.g. $k(u,x) = e^{ux}$

    $$m_X(u) = c_0 + c_1 E[X]u + c_2 E[X^2]u^2 + \cdots$$

  - Characteristic kernel: uniquely determines a probability from m_X.

- For details, see

  Kernel Mean Embedding of Distributions: A Review and Beyond.

  Muandet, Fukumizu, Sriperumbudur and Schölkopf. (2017)

Statistical inference with KME

- Statistical tests
  - Two-sample homogeneity tests (Gretton et al NIPS 2006)

    $\mathbf{X_n} = X_1, \ldots, X_n \sim P$, i.i.d.

    $\mathbf{Y_m} = Y_1, \ldots, Y_m \sim Q$, i.i.d.

    test : P = Q ?

    Maximum mean discrepancy (MMD): $\|m_P - m_Q\|_H$

    → use $\|\hat{m}_P - \hat{m}_Q\|_H^2$ as a test statistics

  - Independent test (Fukumizu et al NIPS 2007)

    Covariance operator on RKHS: $\Sigma_{YX} \coloneqq E[\Phi(Y)\Phi(X)^T] - E[\Phi(Y)]E[\Phi(X)^T]$

    comparing P_{YX} and $P_Y P_X$.

- Implicit generative model

$$MMD(P,Q)^2 = \left\| m_P - m_Q \right\|_H^2$$

A distance measure of the probabilities.
→ Can be used as a criterion of generative model

MMD-GAN (Li et al ICML2015; Li et al NIPS2017; Bińkowski et al ICLR2018)

$$\min_{f_\theta : DNN} \left\| m_{Y=f_\theta(Z)} - m_X \right\|_H^2$$

X: training data (X_i)

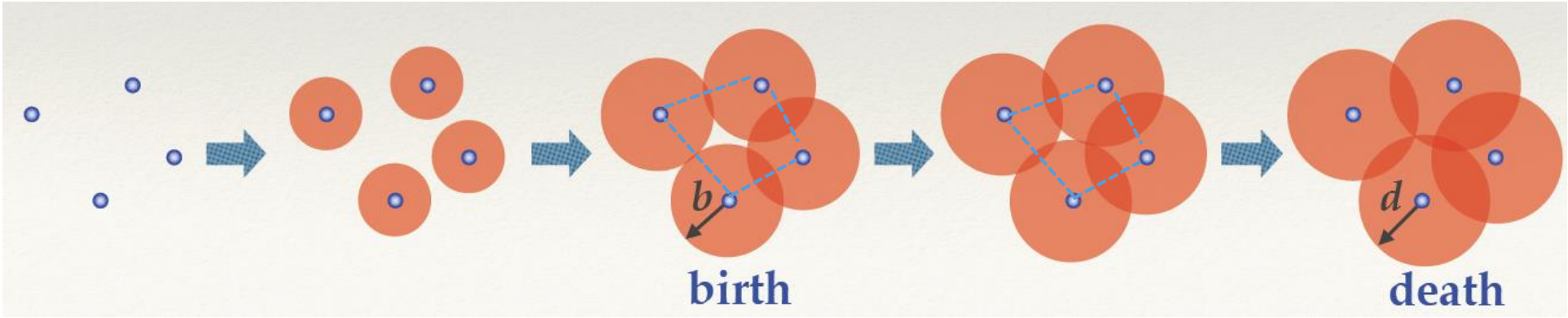
Y: generated by DNN $f_\theta(Z)$

- Bayesian inference by Kernel Bayes Rule
(Fukumizu et al NIPS 2011)

Application to topological data analysis

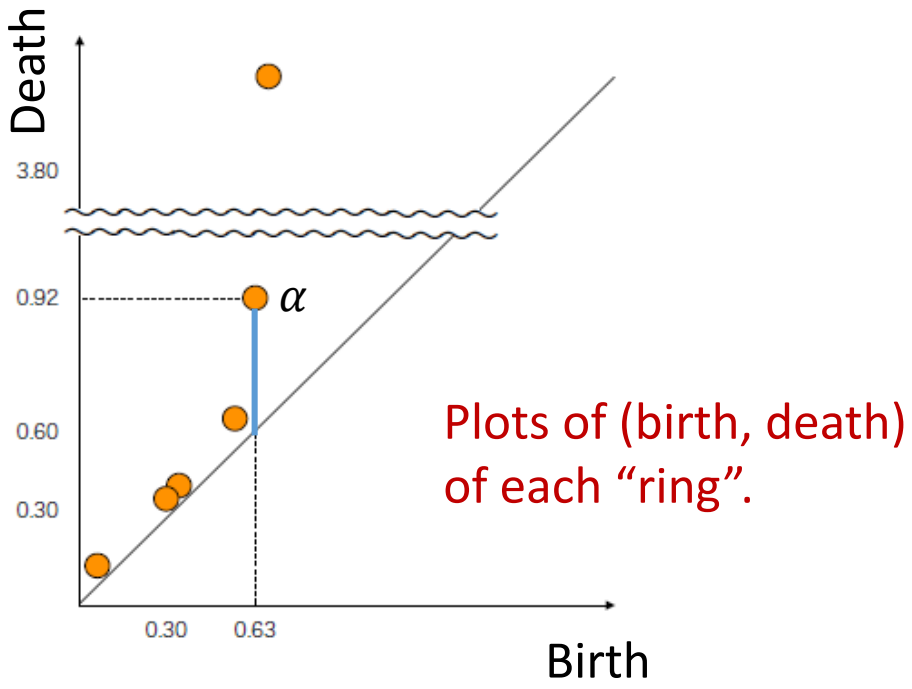
- Persistence homology
  - Expresses topological information of point cloud $\{x_i\}_{i=1}^n$.
  - Essentially represents “holes” in the union of balls $\cup_i B_{x_i}(\varepsilon)$

Birth and death of “rings”



- The birth and dead of all rings encodes the geometry of $\{x_i\}$

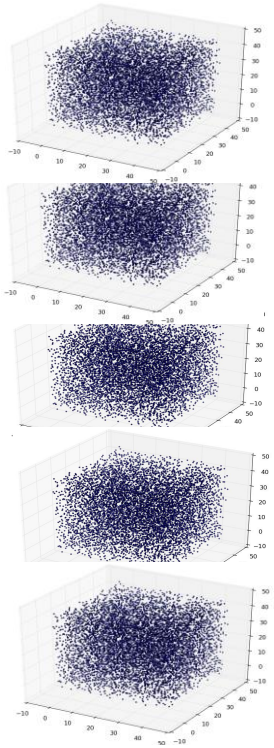
Persistence diagram (PD)



Plots of (birth, death) of each “ring”.

- Handy representation of geometry of point cloud
- A “hole” of stable structure has a long lifetime.
- PD is defined for each dimension (0-dim, 1-dim, ⋯).

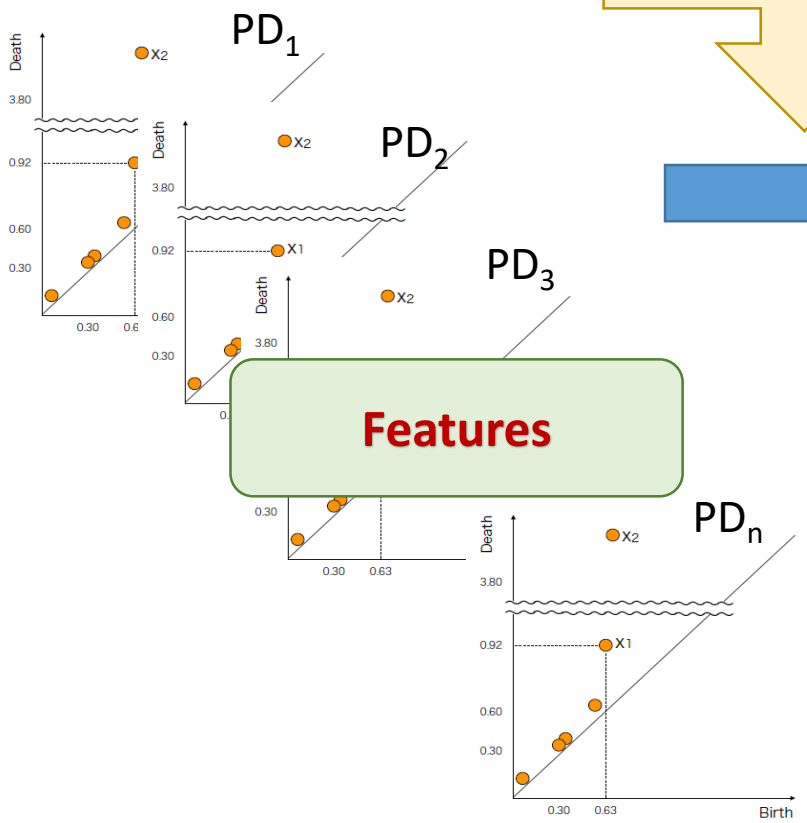
Many point clouds



Computation of PH

Software
e.g. CGAL,Dipha

Many PD's



Features

Kernel Embedding is useful!

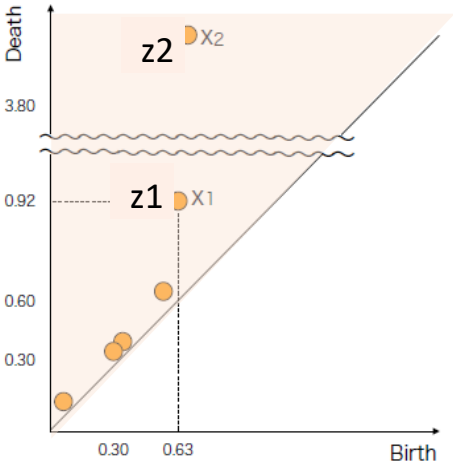
Statistical analysis of PD's

CGAL: The Computational Geometry Algorithms Library
PHAT: Distributed Persistent Homology Algorithm

Vectorizaton of PD's with kernels

- PD = Discrete measure on 2D space

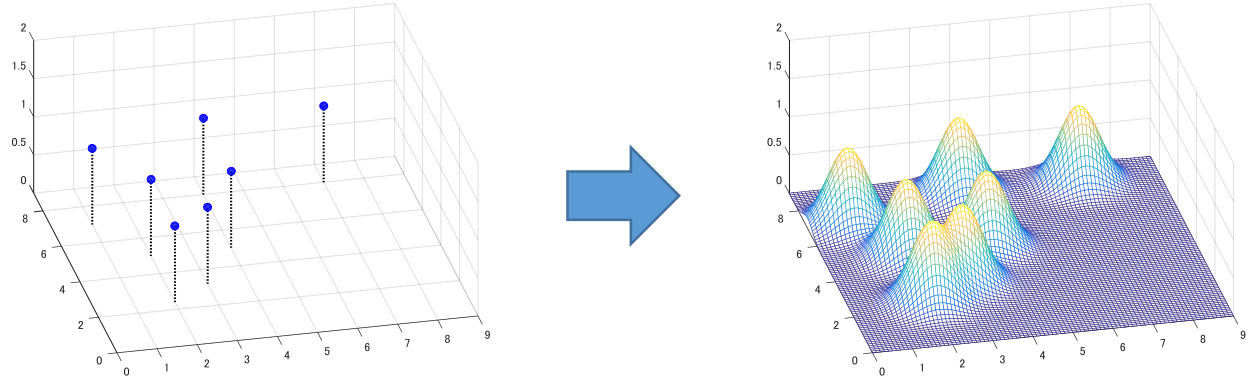
$$\mu_{PD} := \sum_{z \in PD} \delta_z$$



- Kernel embedding of PD's into RKHS

$$V_k(PD) := \sum_i k(\cdot, z_i) \quad \in H_k,$$ Vectorization

e.g. $\sum_{i=1}^{\ell} \exp\left(-\frac{\|u - z_i\|^2}{2\sigma^2}\right)$



- Characteristic kernel (e.g., Gaussian, Laplace) is used.
- By kernelization, any kernel methods can be applied. (KPCA, SVM, etc)

Liquid-glass phases of Silica (SiO_2)

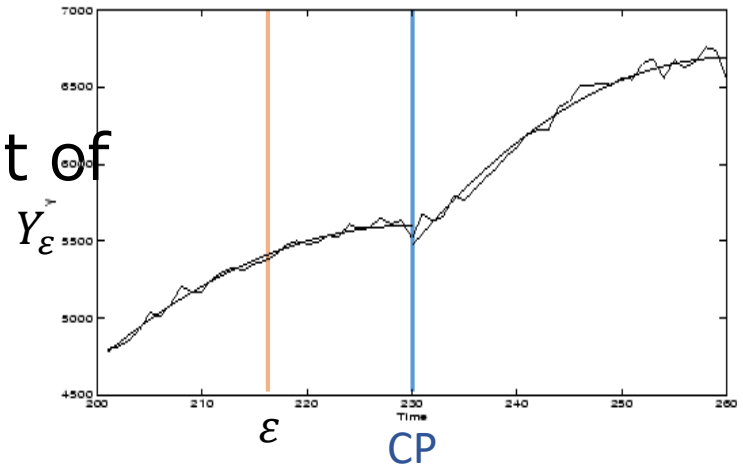
If cooled down quickly, the state changes from liquid to glass

Purpose: identify the temperature of the phase transition.

Data: Molecular dynamics simulation of SiO_2 (Nakamura et al 2015; Hiraoka et al 2016)

- 80 different temperatures (snapshot).
- PD's with the ε-ball model. Different radius are used for Si and O atoms.
- Ring structures made by many atoms have strong influence on the physical property (ring statistics)

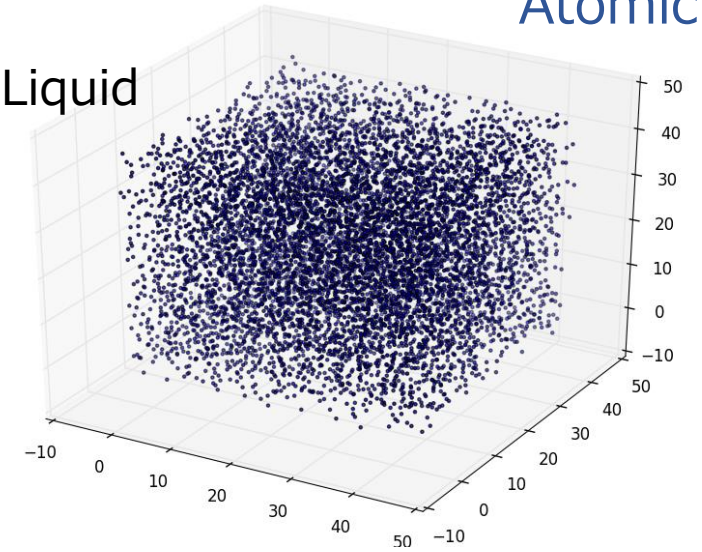
- Physicists' method: estimate discontinuous point of the derivatives of the enthalpy curve.



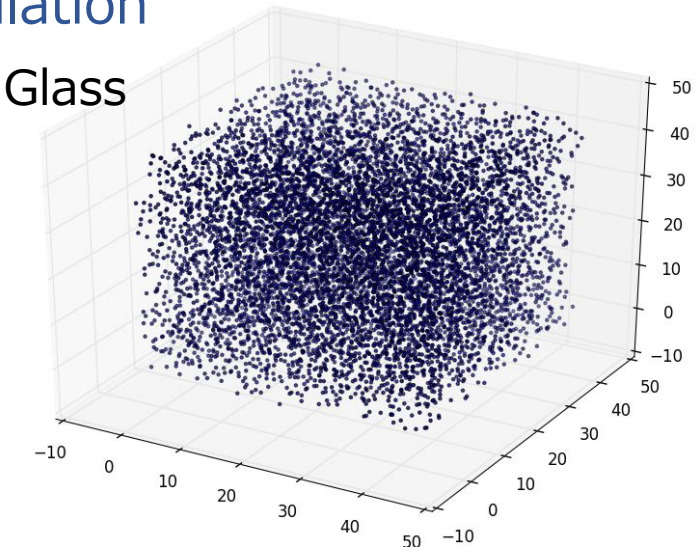
- Proposed: Kernel change point detection
  Kernel Fisher Discriminant Score (KFDS).
  (Harchoui et al NIPS2008)

Atomic arrangement by MD simulation

Liquid

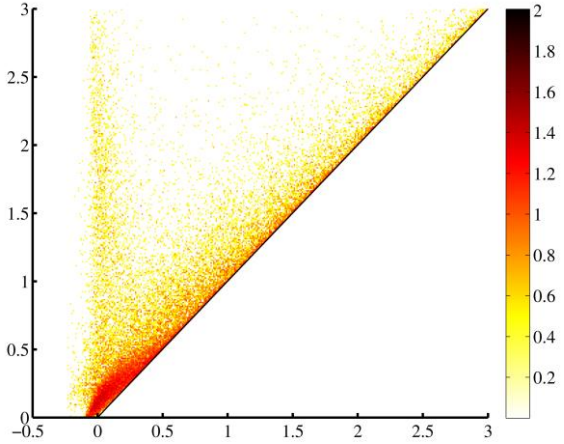


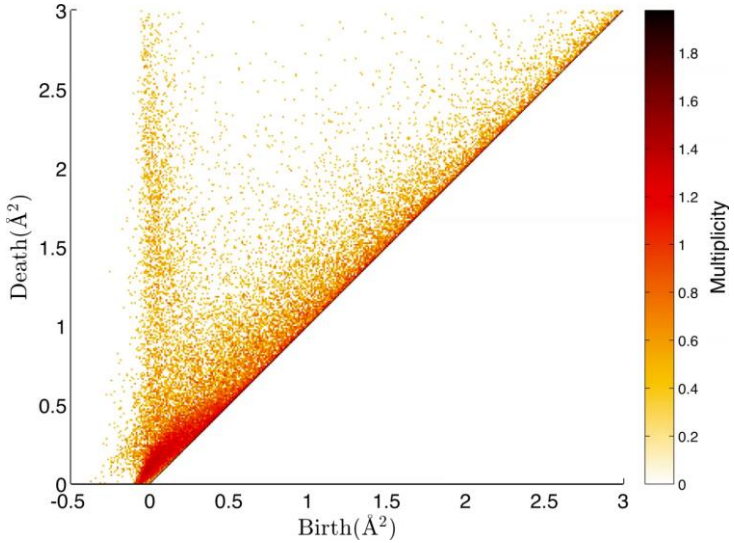
Glass



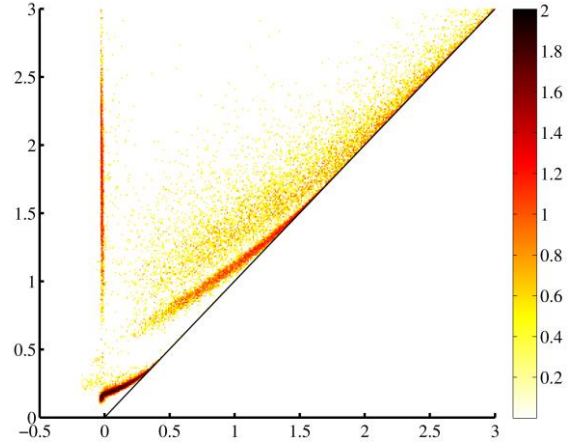
Persistence diagrams

Liquid

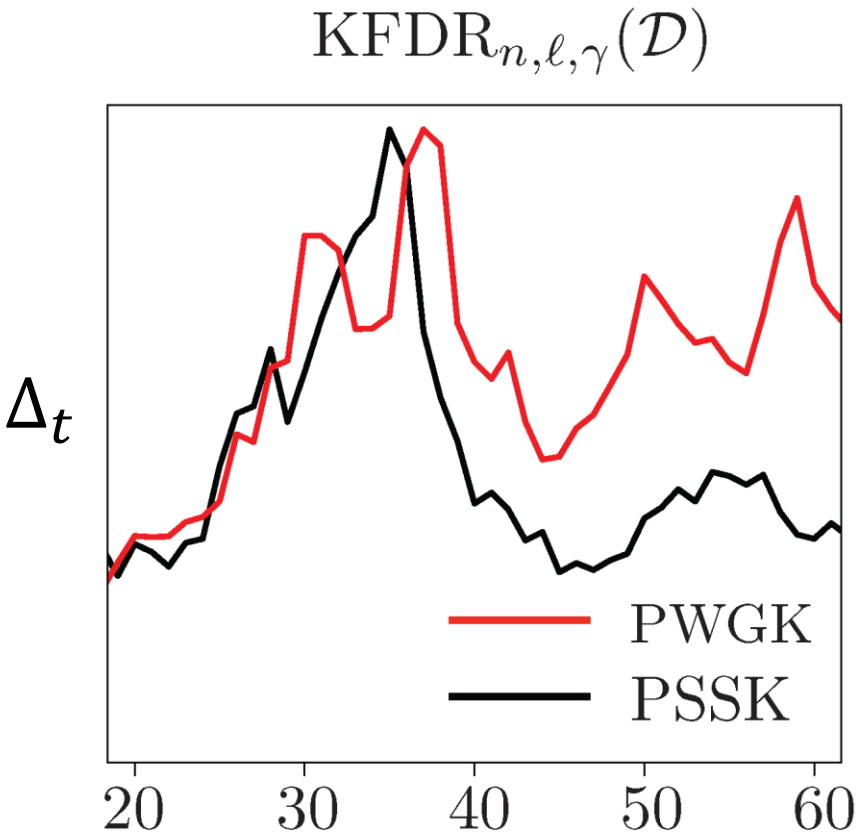




Glass



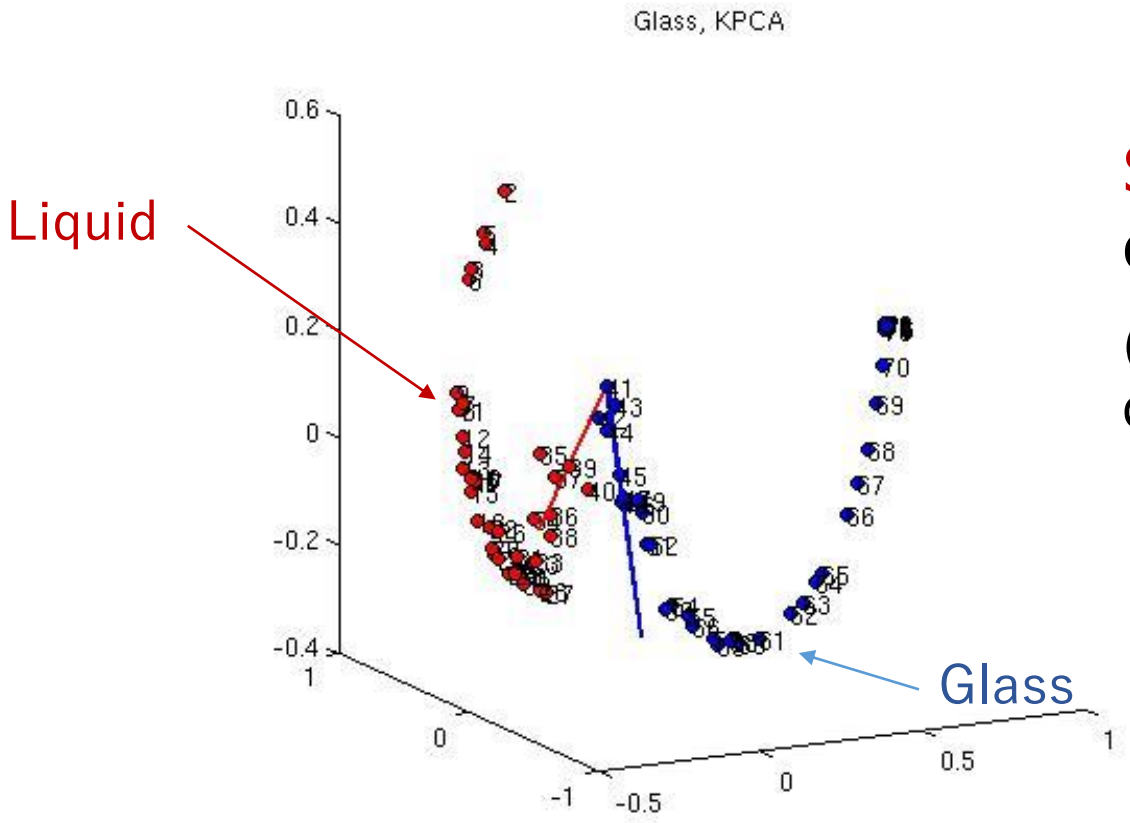
- Change point detection



Detected change point = 3100K

Enthalpy by physicist: [2000K, 3500K]

- 3-dimensional expression by Kernel PCA



Sharp change before and after the change point.

(Colored by the result of change point detection).

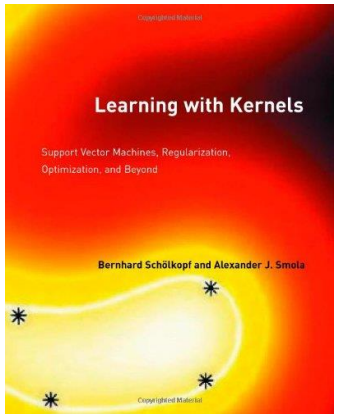
Summary

- Kernel methods: nonlinear data analysis
  - Extract nonlinear feature by kernel (feature map)
  - Computation does not depend (much) on the original space
  - Linear algebra with Gram matrices
  - Applicable to non-vectorial data if a kernel is defined.
    - String, graph, probabilities, etc

- Support vector machine
  - Large margin criterion
  - QP (convex problem), no local minima.
  - Good libraries are provided.

- Approximation for scalability
  - Computation with Gram matrix is expensive for large data.
  - Approximation
    - Low rank approximation: Nystrom, Incomplete Cholesky
    - Random Fourier feature.

References



Standard textbook

- Schölkopf, B. and A. Smola. *Learning with Kernels*. MIT Press. 2002.

Scalable approximation

- Gönen, M. and E. Alpaydın. Multiple Kernel Learning Algorithms, *Journal of Machine Learning Research*, 12(Jul):2211-2268, 2011.
- Fine, S. and K. Scheinberg. (2001) Efficient SVM Training Using Low-Rank Kernel Representations. *Journal of Machine Learning Research*, 2:243-264
- Widom, H. (1963) Asymptotic behavior of the eigenvalues of certain integral equations. *Transactions of the American Mathematical Society*, 109:278--295, 1963
- Widom, H. (1964) Asymptotic behavior of the eigenvalues of certain integral equations II. *Archive for Rational Mechanics and Analysis*, 17:215--229, 1964
- Williams, C. K. I. and M. Seeger. (2001) Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, 13:682–688.

- Drineas, P. and M. W. Mahoney. (2005). On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *J. Mach. Learn. Res*. 6 (December 2005), 2153-2175.
- Rahimi, A. and B. Recht. (2008) Random features for large-scale kernel machines. *Advances in neural information processing systems 20,* (NIPS 2007) 1177-1184
- Dai, B., B. Xie, N. He, Y. Liang, A. Raj, M.-F. Balcan, L. Song. Scalable Kernel Methods via Doubly Stochastic Gradients, *Advances in NIPS* 2014. arXiv version: arXiv:1407.5599 [cs.LG]
- Rudi, A. and Camoriano, R. and Rosasco, L. Less is More: Nystrom Computational Regularization, *Advances in Neural Information Processing Systems 28*, (NIPS 2015) 1657—1665.

Structured data

- Leslie, C., E. Eskin, A. Cohen, J. Weston and W. S. Noble. (2003) Mismatch string kernels for SVM protein classification. Advances in Neural Information Processing Systems 15, pp. 1441-1448.
- Collins, M. & N. Duffy. (2002) Convolution Kernels for Natural Language. Advances in Neural Information Processing Systems 14.
- Schlkopf, B., K. Tsuda, J-P. Vert (Editor) Kernel Methods in Computational Biology. Bradford Books. 2004.

Kernel Mean Embedding

- Muandet, K., Fukumizu, K., Sriperumbudur, B. and Schölkopf, B (2017), "Kernel Mean Embedding of Distributions: A Review and Beyond", *Foundations and Trends in Machine Learning*: Vol. 10: No. 1-2, pp 1-141. http://dx.doi.org/10.1561/2200000060

Addendum

The Institute of Statistical Mathematics

- Founded in 1944 (75th anniversary)
- All aspects of statistics (45 faculty members)

  Theory, methods, and applications.
  - Three departments
    - Dept. of Statistical Modeling
    - Dept. of Statistical Data Science
    - Dept. of Statistical Inference and Mathematics
  - Five Centers
    - Research Center for Statistical Machine Learning
    - Risk Analysis Research Center
    - Data Science Center for Creative Design and Manufacturing
    - Research Center for Medical and Health Data Science
    - School of Statistical Thinking

- “Inter-University Research Institute Corporation”
  - 17 institutes under four organizations. (MEXT)
  - To promote inter-university collaborations in each field.

- Ph.D. course (Graduate University of Advanced Studies)
  - About 35 students

- Located in Tachikawa, a western suburb of Tokyo.

- Grants for visiting students from overseas.
  - Contact me or ISM people!

International Congress of Mathematicians 2018 (Rio de Janeiro)
Research Center for Statistical Machine Learning, ISM, was featured in ICM TV