电子科技大学
University of Electronic Science and Technology of China

Multiple Kernel Learning

Reporter: Zhong Zhang

Data Mining Lab, Big Data Research Center, UESTC
Email：junmshao@uestc.edu.cn
http://staff.uestc.edu.cn/shaojunming

Outline

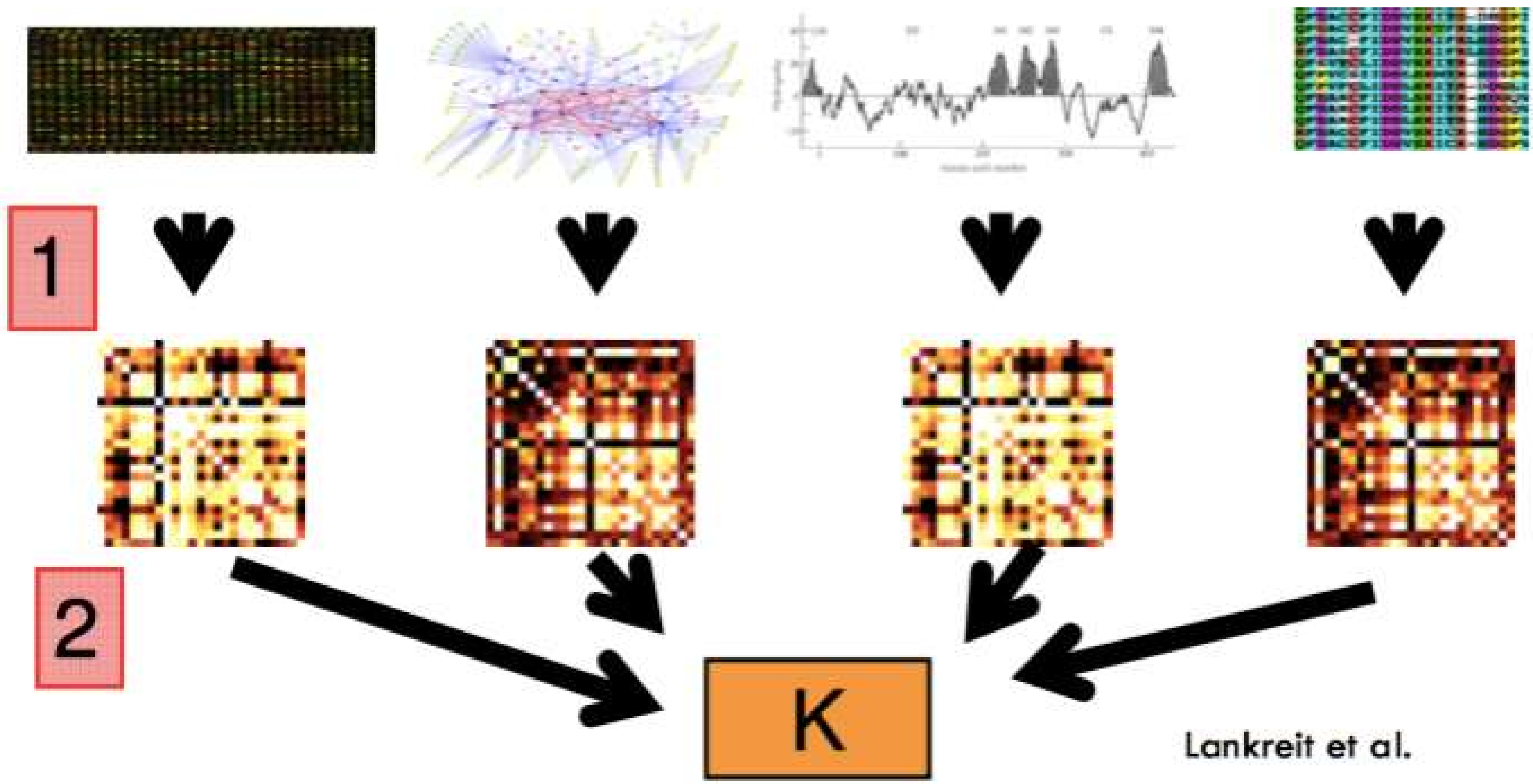
- Introduction
- Kernel
- Multiple kernel learning overview
- Summary
- Reference

Part 1 Introduction

Introduction

数据挖掘实验室
Data Mining Lab

- **Multiple Kernel Learning(MKL)**



Introduction

数据挖掘实验室
Data Mining Lab

- **Advantage of MKL**
  - a) learn optimal kernel and parameter from data automatically
  - b) combining data from different sources

Introduction

数据挖掘实验室
Data Mining Lab

- **MKL Application**

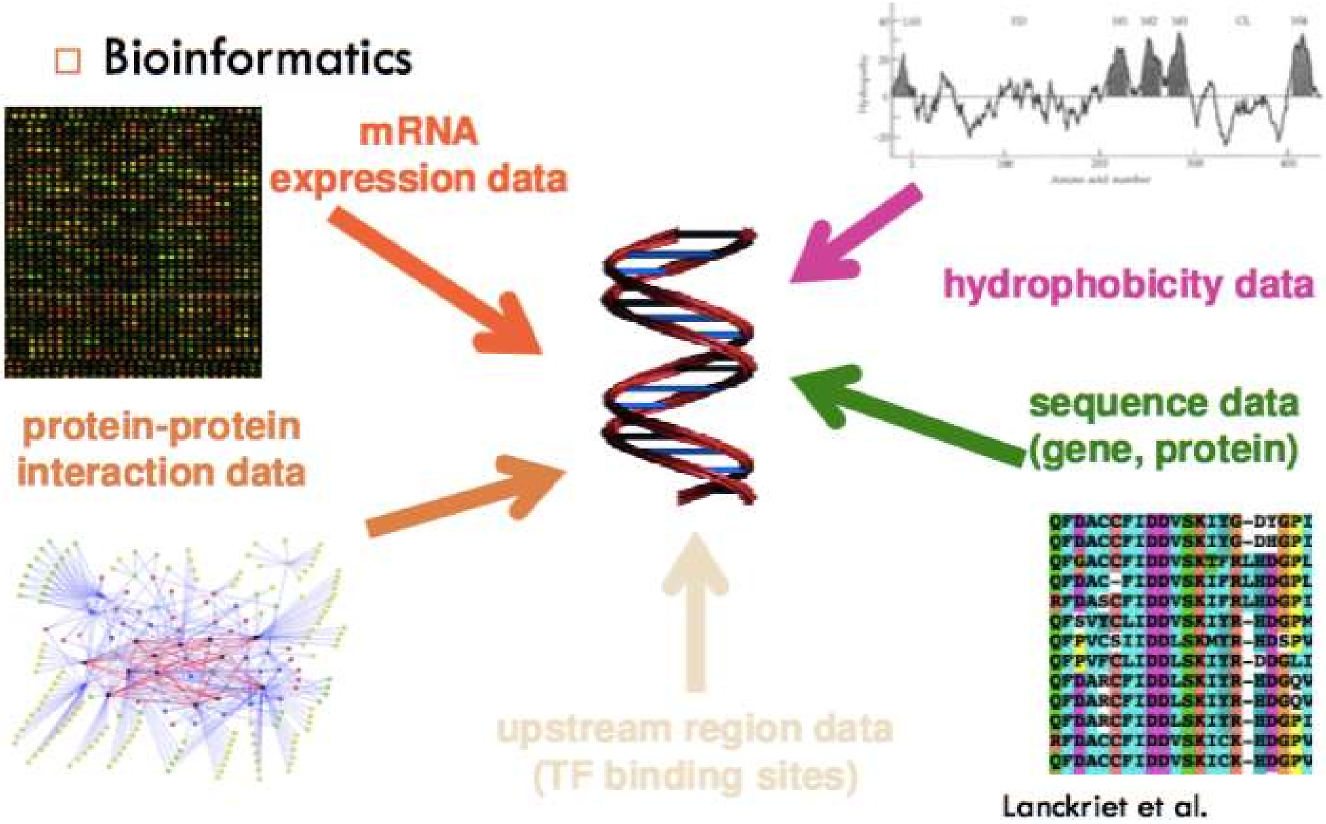
Image categorization and retrieval

- Hundreds of feature types (SIFT, HOG, GIST)
- Select and combine features for improved prediction accuracy and speed

Introduction

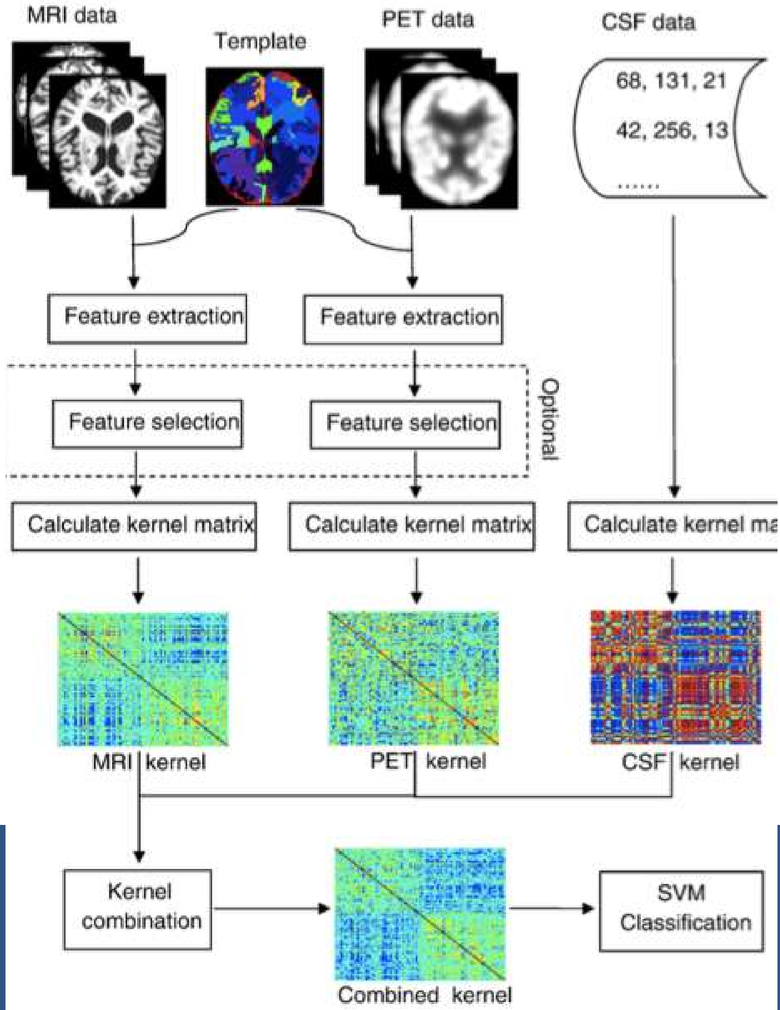
数据挖掘实验室
Data Mining Lab

- **MKL Application**



Introduction

- **MKL Application**

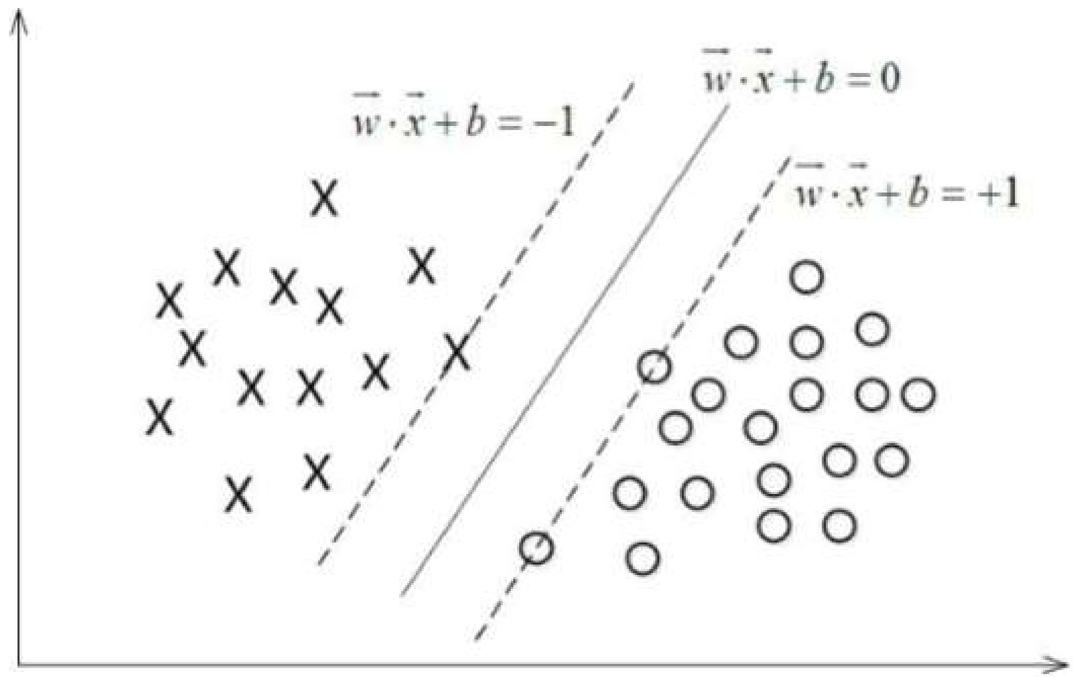


Part 2 Kernel

Kernel

- **Review of SVM**

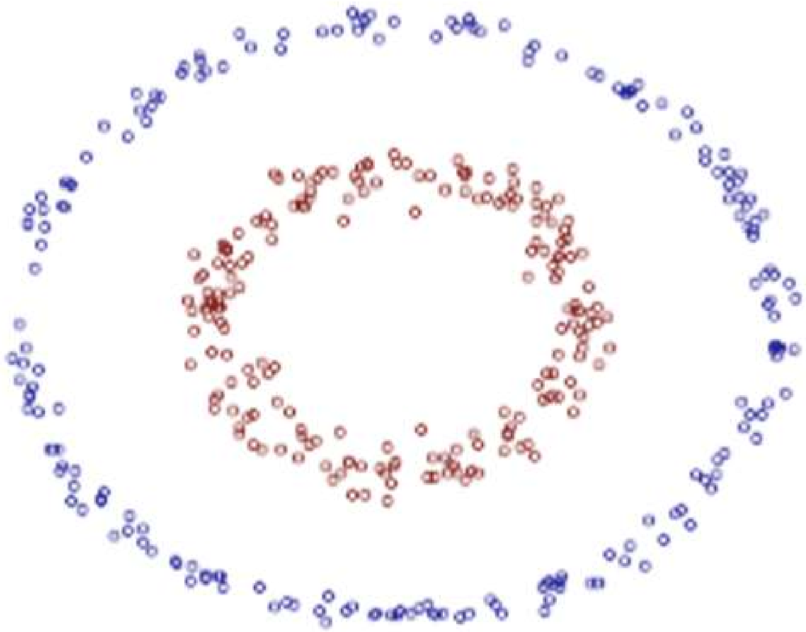
Linearly separable data distribution



Kernel

- **Review of SVM**

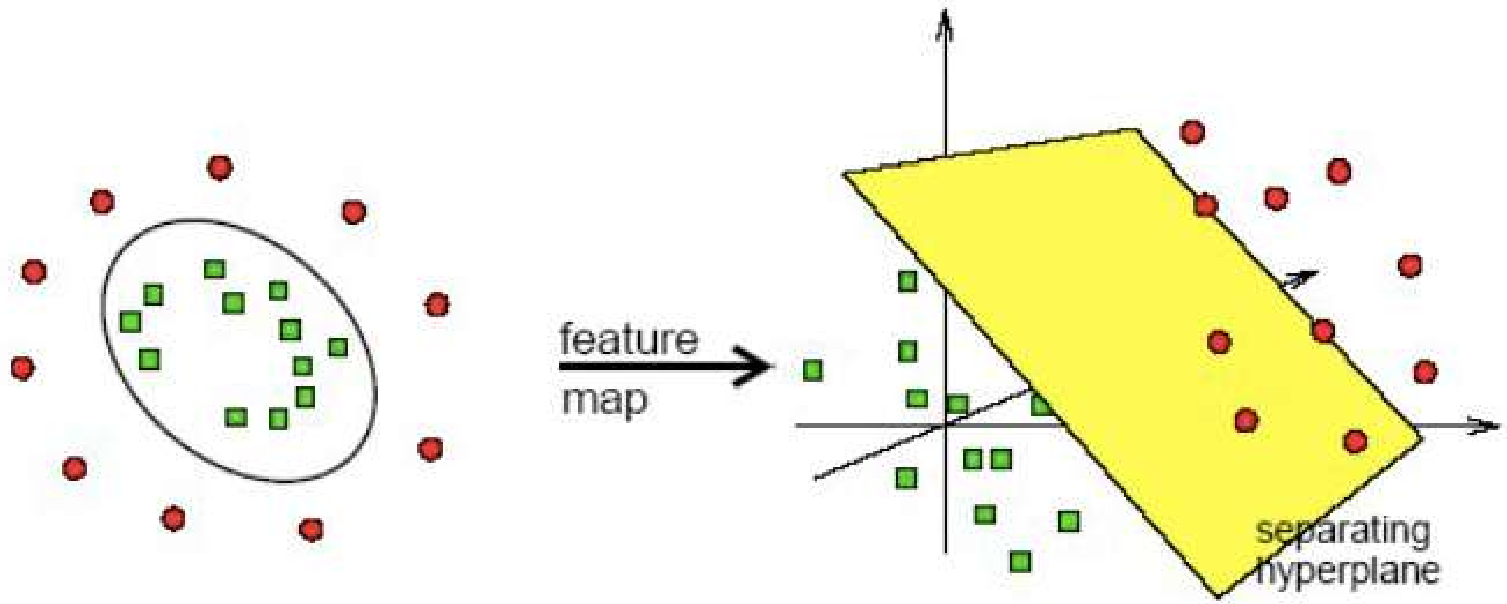
Not linearly separable data distribution



Kernel

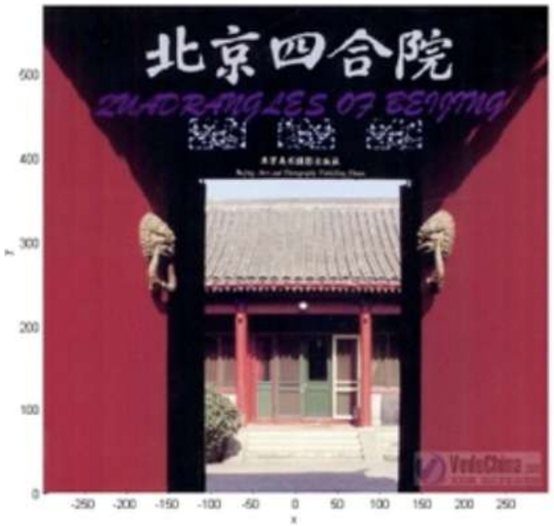
- **Review of SVM**

Separation may be easier in higher dimensions

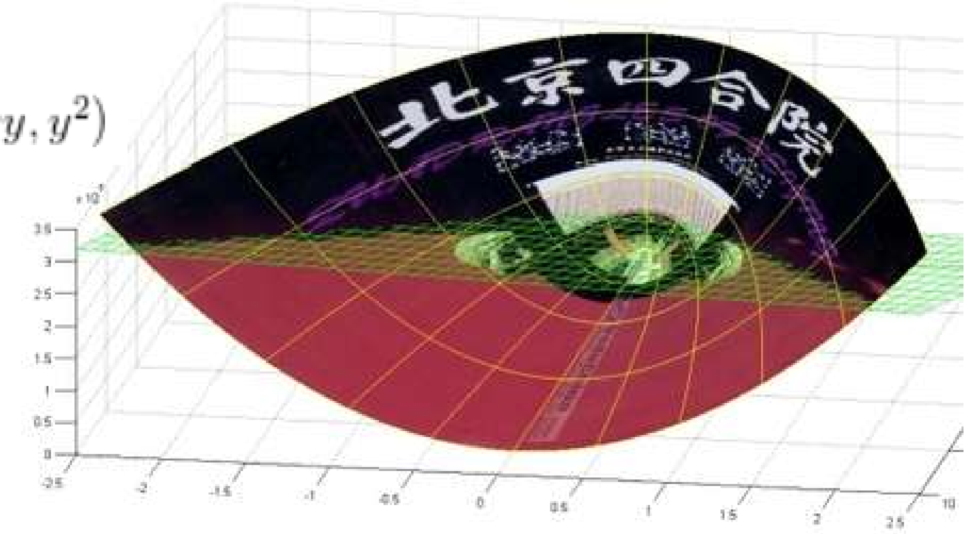


Kernel

- **Review of SVM**

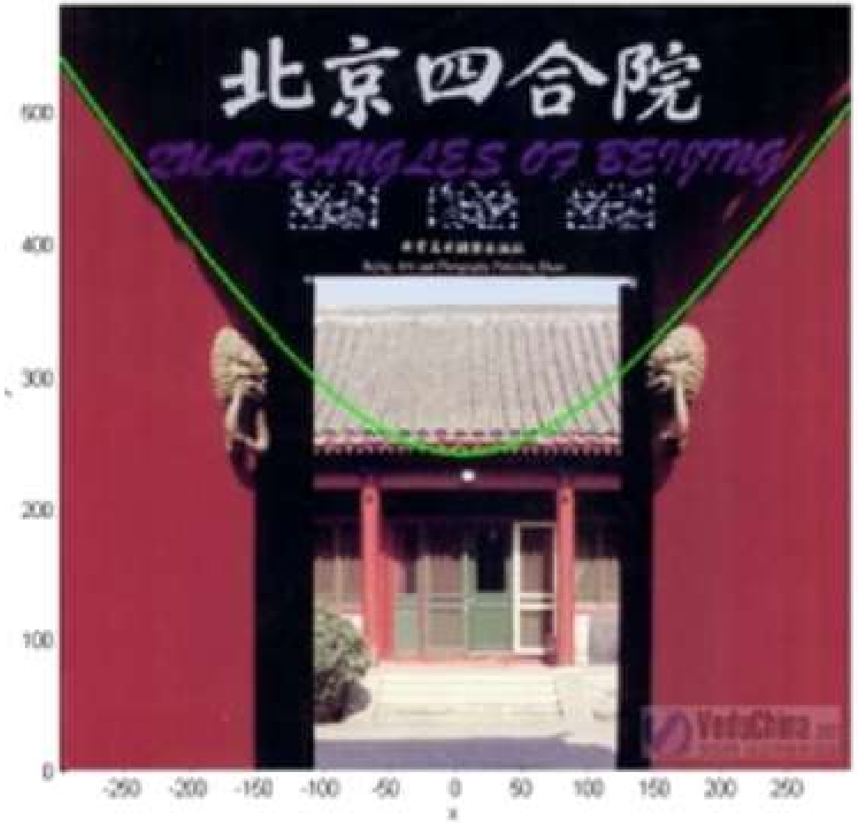


$$P(x, y) = (x^2, \sqrt{2}xy, y^2)$$



Kernel

- **Review of SVM**



Kernel

- **Review of SVM**

Discriminant function:

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b.$$

SVM can be trained by solving the quadratic optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ \text{with respect to} \quad & \mathbf{w} \in \mathbb{R}^S,\ \ \xi \in \mathbb{R}_+^N,\ \ b \in \mathbb{R} \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i \quad \forall i \end{aligned}$$

Kernel

- **Review of SVM**

Lagrangian dual function:

$$\text{maximize} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \underbrace{\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle}_{k(\mathbf{x}_i, \mathbf{x}_j)}$$

$$\text{with respect to} \quad \alpha \in \mathbb{R}_{+}^{N}$$

$$\text{subject to} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$C \geq \alpha_i \geq 0 \quad \forall i$$

Rewrite discriminant function: $f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b.$

Kernel

- **Positive-definite Kernel**

Definition: Let $\mathcal{X}$ be a nonempty set, sometimes referred to as the index set. A symmetric function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ is called a positive definite (p.d.) kernel on $\mathcal{X}$ if

$$\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0$$

hold for any $n \in \mathbb{N}$, $x_1,\ldots,x_n \in \mathcal{X}$, $c_1,\ldots,c_n \in \mathbb{R}$

Kernel

数据挖掘实验室
Data Mining Lab

- **Positive-definite Kernel**

Some general properties [edit]

- For a family of kernels $(K_i)_{i\in\mathbb{N}}$, $\quad K_i : \mathcal{X} \times \mathcal{X} \to R$
  - The sum $\sum_{i=1}^{n} \lambda_i K_i$ is p.d., given $\lambda_1, \dots, \lambda_n \geq 0$
  - The product $K_1^{a_1} \dots K_n^{a_n}$ is p.d., given $a_1, \dots, a_n \in \mathbb{N}$
  - The limit $K = \lim_{n\to\infty} K_n$ is p.d. if the limit exists.
- If $(\mathcal{X}_i)_{i=1}^{n}$ is a sequence of sets, and $(K_i)_{i=1}^{n}$, $\quad K_i : \mathcal{X}_i \times \mathcal{X}_i \to R$ a sequence of p.d. kernels, then both $K((x_1, \dots, x_n), (y_1, \dots, y_n)) = \prod_{i=1}^{n} K_i(x_i, y_i)$ and $K((x_1, \dots, x_n), (y_1, \dots, y_n)) = \sum_{i=1}^{n} K_i(x_i, y_i)$ are p.d. kernels on $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_n$.
- Let $\mathcal{X}_0 \subset \mathcal{X}$. Then the restriction K_0 of K to $\mathcal{X}_0 \times \mathcal{X}_0$ is also a p.d. kernel.

Kernel

- **Example of p.d. Kernels**

Linear kernel: $K(x, y) = x^T y, x, y \in \mathbb{R}^d$.

Polynomial kernel: $K(x, y) = (x^T y + r)^n, x, y \in \mathbb{R}^d, r > 0$.

Gaussian kernel (RBF Kernel): $K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}, x, y \in \mathbb{R}^d, \sigma > 0$.

Laplacian kernel: $K(x, y) = e^{-\alpha\|x-y\|}, x, y \in \mathbb{R}, \alpha > 0$.

Abel kernel: $K(x, y) = e^{-\alpha|x-y|}, x, y \in \mathbb{R}, \alpha > 0$.

Kernel

- **Reproducing Kernel Hilbert Spaces(RKHS)**

Notation：

$\mathcal{X}$ is a set

H is a Hilbert space of functions $f : \mathcal{X} \longrightarrow \mathbb{R}$

$(\cdot, \cdot)_H : H \times H \longrightarrow \mathbb{R}$, the corresponding inner product on H

$e_x : H \longrightarrow \mathbb{R}$ is evaluation functional, $e_x(f) = f(x)$ for any $x \in \mathcal{X}$

Kernel

- **Reproducing Kernel Hilbert Spaces**

Definition: Space H is called a reproducing kernel Hilbert space(RKHS) if the evaluation functionals are continuous

Definition: Reproducing kernel is a function $K: \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ such that

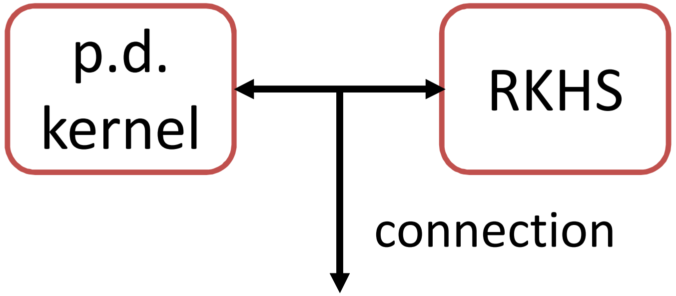
1) $K_x(\cdot) \in H$, $\forall x \in \mathcal{X}$, and

2) $(f, K_x)_H = f(x)$, for all $f \in H$ and $x \in \mathcal{X}$ (reproducing property)

Kernel

- **Reproducing Kernel Hilbert Spaces**

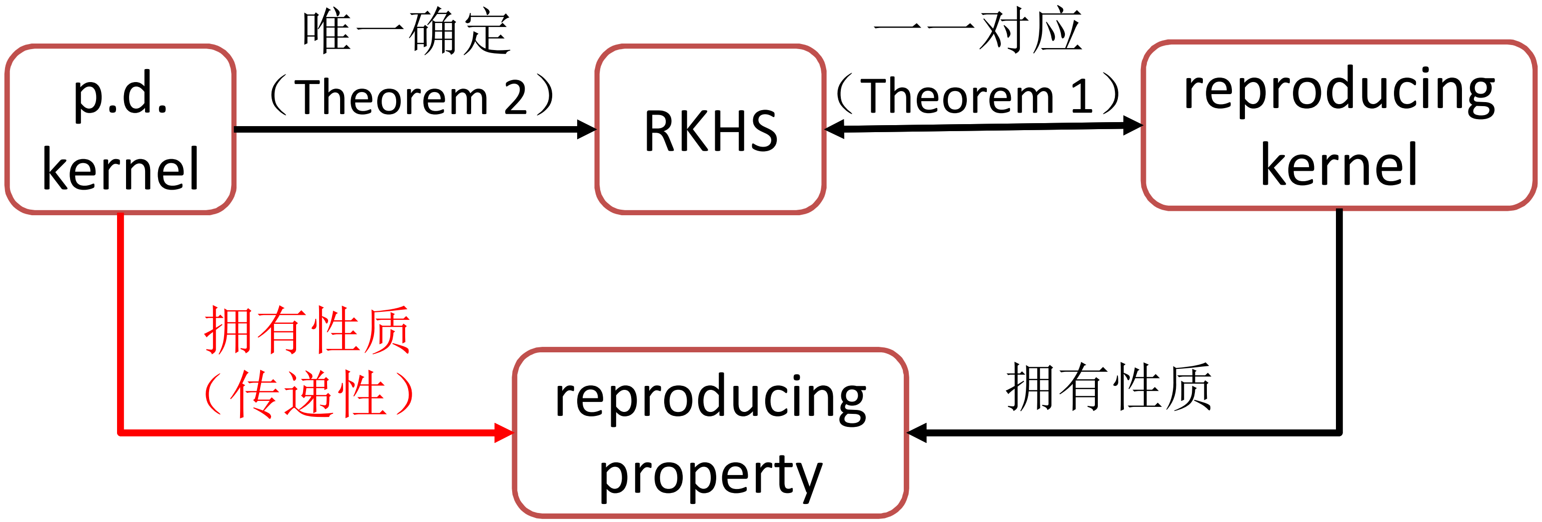
Theorem 1: Every reproducing kernel K induces a unique RKHS, and every RKHS has a unique reproducing kernel



Theorem 2: Every reproducing kernel is positive definite, and every p.d. kernel defines a unique RKHS, of which it is the unique reproducing kernel

Kernel

- **Reproducing Kernel Hilbert Spaces**



This means p.d. kernels can be constructed from inner products

Kernel

- **Feature Map**

Notation：

F is a Hilbert space, called feature space

$\phi : \mathcal{X} \longrightarrow F$ is called a feature map

$(\cdot, \cdot)_F : F \times F \longrightarrow \mathbb{R}$ the corresponding inner product on F

Kernel

- **Feature Map**

Let $F = H$, $\phi(x) = K_x$ for all $x \in \mathcal{X}$, then

$$(\phi(x), \phi(y))_F = (K_x, K_y)_H = K(x, y) \text{ (reproducing property)}$$

This is kernel trick

Note: 1)kernel function is not feature map itself

2)kernel provide a way to calculate the inner product of feature in the new feature space

Part 3 Multiple Kernel Learning Overview

MKL overview

- **General Form of MKL**

$$k_\eta(\mathbf{x}_i,\mathbf{x}_j) = f_\eta(\{k_m(\mathbf{x}_{mi},\mathbf{x}_{mj})\}_{m=1}^{P})$$

Where

$f_\eta : \mathbb{R}^P \to \mathbb{R}$, combination function, linear or nonlinear

$k_m : \mathbb{R}^{Dm} \times \mathbb{R}^{Dm} \to \mathbb{R}$, kernel function

P , feature representations (not necessarily different) of data

D_m, dimension of the corresponding feature representation

η , parameterizes the combination function

MKL overview

数据挖掘实验室
Data Mining Lab

- **Two Types of General Form**

  - $k_\eta(\mathbf{x}_i,\mathbf{x}_j) = f_\eta(\{k_m(\mathbf{x}_{mi}, \mathbf{x}_{mj})\}_{m=1}^{P} | \eta)$

  parameters are used to combine predefined kernels, (i.e. know kernel functions and parameters before training)

  - $k_\eta(\mathbf{x}_i,\mathbf{x}_j) = f_\eta(\{k_m(\mathbf{x}_{mi}, \mathbf{x}_{mj} | \eta)\}_{m=1}^{P})$

  parameters integrated into the kernel functions are optimized during training

MKL overview

- **Three Types of Integrating Data**
  - early combination
  - intermediate combination (combine kernel)
  - late combination

MKL overview

- **Key Properties of MKL**
  - Learning method
  - Functional form
  - Target function
  - Training method
  - Base learner
  - Computational complexity

MKL overview

- **Learning Method**
  - Fixed rules

functions without any parameters and do not need any training

e.g.

$$k_\eta(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{P} k_m(\mathbf{x}_i^m, \mathbf{x}_j^m)$$

$$k_\eta(\mathbf{x}_i, \mathbf{x}_j) = \prod_{m=1}^{P} k_m(\mathbf{x}_i^m, \mathbf{x}_j^m).$$

MKL overview

数据挖掘实验室
Data Mining Lab

- **Learning Method**
  - Heuristic approaches

select the kernel weights by looking at the performance values obtained by each kernel separately

e.g.

$$k_\eta(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{P} \eta_m k_m(\mathbf{x}_i^m, \mathbf{x}_j^m)$$

accuracy obtained using only **K***m*

$$\eta_m = \frac{\pi_m - \delta}{\sum_{h=1}^{P} (\pi_h - \delta)}$$

threshold

MKL overview

数据挖掘实验室
Data Mining Lab

- **Learning Method**
  - Optimization approaches

learn the parameters by solving an optimization problem

e.g. optimize separately

$$\begin{aligned} \text{maximize} \quad & \mathrm{A}(\mathbf{K}_{\eta}^{\text{tra}}, \mathbf{y}\mathbf{y}^{\top}) \\ \text{with respect to} \quad & \mathbf{K}_{\eta} \in \mathbb{S}^{N} \\ \text{subject to} \quad & \mathrm{tr}\left(\mathbf{K}_{\eta}\right) = 1 \\ & \mathbf{K}_{\eta} \succeq 0 \end{aligned}$$

MKL overview

- **Learning Method**
  - Optimization approaches

learn the parameters by solving an optimization problem

e.g. optimize jointly

$$\min_{\mathbf{w},b,\boldsymbol{\xi}\geq\mathbf{0},\mathbf{d}\geq\mathbf{0}} \frac{1}{2}\sum_k \mathbf{w}_k^t\mathbf{w}_k + C\sum_i \xi_i + \frac{\lambda}{2}\Big(\sum_k d_k^p\Big)^{\frac{2}{p}}$$

$$\text{s. t. } y_i\Big(\sum_k \sqrt{d_k}\,\mathbf{w}_k^t\phi_k(\mathbf{x}_i)+b\Big) \geq 1-\xi_i$$

MKL overview

数据挖掘实验室
Data Mining Lab

- **Learning Method**

  - Bayesian approaches

  Interpret combination parameter as random variables, put priors on parameters

  e.g.
$$f(\mathbf{x}) = \sum_{i=0}^{N} \alpha_i \sum_{m=1}^{P} \eta_m k_m(\mathbf{x}_i^m, \mathbf{x}^m)$$

  η is modeled with a Dirichlet prior

  α is modeled with a zero-mean Gaussian with an inverse gamma variance prior

MKL overview

- **Learning Method**
  - Boosting approaches

  Iteratively a add new kernel until the performance stops improving

MKL overview

- **Function form**
  - Linear combination

$$k_\eta(\mathbf{x}_i, \mathbf{x}_j) = f_\eta(\{k_m(\mathbf{x}_i^m, \mathbf{x}_j^m)\}_{m=1}^P | \eta) = \sum_{m=1}^{P} \eta_m k_m(\mathbf{x}_i^m, \mathbf{x}_j^m)$$

restrictions on η: linear sum (i.e., $\eta \in \mathbb{R}^P$)

conic sum (i.e., $\eta \in \mathbb{R}_+^P$)

convex sum (i.e., $\eta \in \mathbb{R}_+^P$ and $\sum_{m=1}^{P} \eta_m = 1$)

MKL overview

- **Function form**
  - Linear combination

    advantage of conic and convex sums

    1) easy to extract important kernel

    2) interpret feature representation (if nonnegative)

$$\langle \Phi_\eta(\mathbf{x}_i), \Phi_\eta(\mathbf{x}_j) \rangle = \begin{pmatrix} \sqrt{\eta_1}\Phi_1(\mathbf{x}_i^1) \\ \sqrt{\eta_2}\Phi_2(\mathbf{x}_i^2) \\ \vdots \\ \sqrt{\eta_P}\Phi_P(\mathbf{x}_i^P) \end{pmatrix}^\top \begin{pmatrix} \sqrt{\eta_1}\Phi_1(\mathbf{x}_j^1) \\ \sqrt{\eta_2}\Phi_2(\mathbf{x}_j^2) \\ \vdots \\ \sqrt{\eta_P}\Phi_P(\mathbf{x}_j^P) \end{pmatrix} = \sum_{m=1}^{P} \eta_m k_m(\mathbf{x}_i^m, \mathbf{x}_j^m).$$

MKL overview

- **Function form**

  ➢ Linear combination

  Lp-norm restriction is also applicable

  e.g. L1-norm promotes sparsity on the kernel level, which can be interpreted as feature selection

  L2-norm usually prevent overfiting

MKL overview

- **Function form**
  - Nonlinear combination

  Combine kernel by multiplication, power, and exponentiation…

MKL overview

- **Function form**
  - ➢ Data-dependent combination

  assign specific kernel weights for each data instance

MKL overview

数据挖掘实验室
Data Mining Lab

- **Target function**

  - Similarity-based functions

maximize the similarity between the combined kernel matrix and an optimum kernel matrix

e.g.

$$\begin{aligned} \text{maximize} \quad & \mathrm{A}(\mathbf{K}_{\eta}^{\mathrm{tra}}, \mathbf{y}\mathbf{y}^{\top}) \\ \text{with respect to} \quad & \mathbf{K}_{\eta} \in \mathbb{S}^{N} \\ \text{subject to} \quad & \mathrm{tr}\left(\mathbf{K}_{\eta}\right) = 1 \\ & \mathbf{K}_{\eta} \succeq 0 \end{aligned}$$

MKL overview

数据挖掘实验室
Data Mining Lab

- **Target function**
  - Structural risk functions

minimize the sum of a regularization term and error term

e.g.

$$\min_{\mathbf{w},b,\boldsymbol{\xi}\geq 0,\mathbf{d}\geq 0} \frac{1}{2}\sum_k \mathbf{w}_k^t\mathbf{w}_k + C\sum_i \xi_i + \frac{\lambda}{2}\Big(\sum_k d_k^p\Big)^{\frac{2}{p}}$$

$$\text{s. t. } y_i\Big(\sum_k \sqrt{d_k}\,\mathbf{w}_k^t\phi_k(\mathbf{x}_i)+b\Big) \geq 1-\xi_i$$

L1-norm, L2-norm or Lp-norm are used on the kernel weights or feature spaces

MKL overview

- **Target function**
  - ➢ Bayesian functions

  measure the quality of the resulting kernel function constructed from candidate kernels using a Bayesian formulation

  likelihood or posterior are usually used as the target function

MKL overview

- **Target function**
  - Training method
    1) One-step method
    2) Two-step method

MKL overview

数据挖掘实验室
Data Mining Lab

- **Base learner**
  - SVM(SVR)
  - kernel Fisher discriminant analysis (KFDA)
  - regularized kernel discriminant analysis (RKDA)
  - kernel ridge regression (KRR)
  - Multinomial probit and Gaussian process (GP)

MKL overview

- **Computational complexity**
  - One-step methods using fixed rules and heuristics generally do not spend much time
  - One-step methods using optimization have high computational complexity
  - Two-step methods update combination function parameters and base learner parameters in an alternating manner

MKL overview

数据挖掘实验室
Data Mining Lab

Representative References	Learning Method	Functional Form	Target Function	Training Method	Base Learner	Computational Complexity
Pavlidis et al. (2001)	Fixed	Lin. (unwei.)	None	1-step (seq.)	SVM	QP
Ben-Hur and Noble (2005)	Fixed	Lin. (unwei.)	None	1-step (seq.)	SVM	QP
de Diego et al. (2004, 2010a)	Heuristic	Nonlinear	Val. error	2-step	SVM	QP
Moguerza et al. (2004); de Diego et al. (2010a)	Heuristic	Data-dep.	None	1-step (seq.)	SVM	QP
Tanabe et al. (2008)	Heuristic	Lin. (convex)	None	1-step (seq.)	SVM	QP
Qiu and Lane (2009)	Heuristic	Lin. (convex)	None	1-step (seq.)	SVR	QP
Qiu and Lane (2009)	Heuristic	Lin. (convex)	None	1-step (seq.)	SVM	QP
Lanckriet et al. (2004a)	Optim.	Lin. (linear)	Similarity	1-step (seq.)	SVM	SDP+QP
Igel et al. (2007)	Optim.	Lin. (linear)	Similarity	1-step (seq.)	SVM	Grad.+QP
Cortes et al. (2010a)	Optim.	Lin. (linear)	Similarity	1-step (seq.)	SVM	Mat. Inv.+QP
Lanckriet et al. (2004a)	Optim.	Lin. (conic)	Similarity	1-step (seq.)	SVM	QCQP+QP
Kandola et al. (2002)	Optim.	Lin. (conic)	Similarity	1-step (seq.)	SVM	QP+QP
Cortes et al. (2010a)	Optim.	Lin. (conic)	Similarity	1-step (seq.)	SVM	QP+QP
He et al. (2008)	Optim.	Lin. (convex)	Similarity	1-step (seq.)	SVM	QP+QP
Tanabe et al. (2008)	Optim.	Lin. (convex)	Similarity	1-step (seq.)	SVM	QP+QP
Ying et al. (2009)	Optim.	Lin. (convex)	Similarity	1-step (seq.)	SVM	Grad.+QP
Lanckriet et al. (2002)	Optim.	Lin. (linear)	Str. risk	1-step (seq.)	SVM	SDP+QP
Qiu and Lane (2005)	Optim.	Lin. (linear)	Str. risk	1-step (seq.)	SVR	SDP+QP
Conforti and Guido (2010)	Optim.	Lin. (linear)	Str. risk	1-step (seq.)	SVM	SDP+QP
Lanckriet et al. (2004a)	Optim.	Lin. (conic)	Str. risk	1-step (seq.)	SVM	QCQP+QP
Fung et al. (2004)	Optim.	Lin. (conic)	Str. risk	2-step	KFDA	QP+Mat. Inv.
Tsuda et al. (2004)	Optim.	Lin. (conic)	Str. risk	2-step	KFDA	Grad.+Mat. Inv

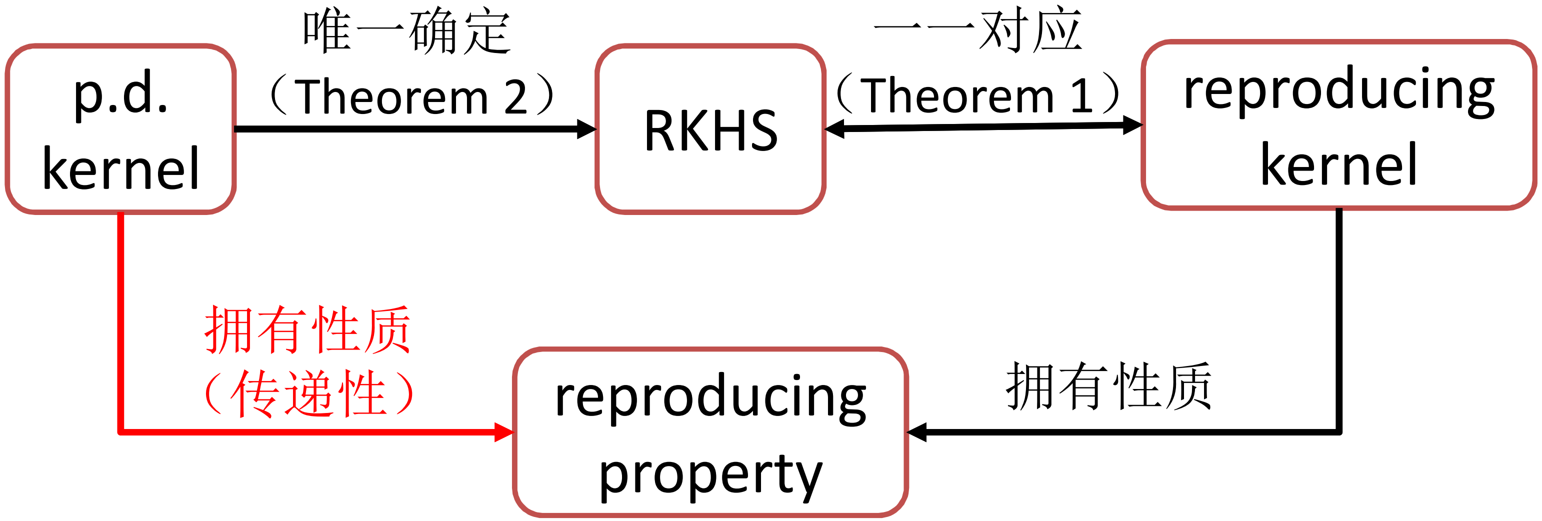
MKL overview

Representative References	Learning Method	Functional Form	Target Function	Training Method	Base Learner	Computational Complexity
Bousquet and Herrmann (2003)	Optim.	Lin. (convex)	Str. risk	2-step	SVM	Grad.+QP
Bach et al. (2004)	Optim.	Lin. (convex)	Str. risk	1-step (sim.)	SVM	SOCP
Sonnenburg et al. (2006a,b)	Optim.	Lin. (convex)	Str. risk	2-step	SVM	LP+QP
Kim et al. (2006)	Optim.	Lin. (convex)	Str. risk	1-step (seq.)	KFDA	SDP+Mat. Inv.
Ye et al. (2007a)	Optim.	Lin. (convex)	Str. risk	1-step (seq.)	RKDA	SDP+Mat. Inv.
Ye et al. (2007b)	Optim.	Lin. (convex)	Str. risk	1-step (seq.)	RKDA	QCQP+Mat. Inv.
Ye et al. (2008)	Optim.	Lin. (convex)	Str. risk	1-step (seq.)	RKDA	SILP+Mat. Inv.
Rakotomamonjy et al. (2007, 2008)	Optim.	Lin. (convex)	Str. risk	2-step	SVM	Grad.+QP
Chapelle and Rakotomamonjy (2008)	Optim.	Lin. (convex)	Str. risk	2-step	SVM	QP+QP
Kloft et al. (2010b); Xu et al. (2010a)	Optim.	Lin. (convex)	Str. risk	2-step	SVM	Analytical+QP
Conforti and Guido (2010)	Optim.	Lin. (convex)	Str. risk	1-step (seq.)	SVM	QCQP+QP
Lee et al. (2007)	Optim.	Nonlinear	Str. risk	1-step (sim.)	SVM	QP
Varma and Babu (2009)	Optim.	Nonlinear	Str. risk	2-step	SVM	Grad.+QP
Cortes et al. (2010b)	Optim.	Nonlinear	Str. risk	2-step	KRR	Grad.+Mat. Inv.
Lewis et al. (2006b)	Optim.	Data-dep.	Str. risk	1-step (sim.)	SVM	QP
Gönen and Alpaydın (2008)	Optim.	Data-dep.	Str. risk	2-step	SVM	Grad.+QP
Yang et al. (2009a)	Optim.	Data-dep.	Str. risk	2-step	SVM	Grad.+QP
Yang et al. (2009b, 2010)	Optim.	Data-dep.	Str. risk	2-step	SVM	SILP+QP
Girolami and Rogers (2005)	Bayesian	Lin. (conic)	Likelihood	Inference	KRR	Approximation
Girolami and Zhong (2007)	Bayesian	Lin. (conic)	Likelihood	Inference	GP	Approximation
Christoudias et al. (2009)	Bayesian	Data-dep.	Likelihood	Inference	GP	Approximation
Bennett et al. (2002)	Boosting	Data-dep.	Str. risk	$P \times 1$-step	KRR	Mat. Inv.
Crammer et al. (2003)	Boosting	Lin. (conic)	Str. risk	$P \times 1$-step	Percept.	Eigenvalue Prob.
Bi et al. (2004)	Boosting	Lin. (linear)	Str. risk	$P \times 1$-step	SVM	QP

Part 4 Summary

Summary

- **Kernel**



$(\phi(x), \phi(y))_F = (K_x, K_y)_H = K(x, y)$ (reproducing property)

Summary

- **Key Properties of MKL**
  - Learning method
    - Fixed rules
    - Heuristic approaches
    - Optimization approaches
    - Bayesian approaches
    - Boosting approaches

Summary

- **Key Properties of MKL**
  - ➢ Functional form
    - ➢ Linear combination
    - ➢ Nonlinear combination
    - ➢ Data-dependent combination

Summary

- **Key Properties of MKL**
  - Target function
    - *Similarity-based functions*
    - Structural risk functions
    - Bayesian functions

Summary

- **Key Properties of MKL**
  - Training method
    - One-step
    - Two-step
  - Base learner
    - SVM
    - KFDA
    - …

Summary

- **Key Properties of MKL**
  - Computational complexity
    - One-step
    - Two-step

Reference

- M. Gönen, E. Alpaydın, “Multiple Kernel Learning Algorithms” in JMLR, 2011
- Francis Bach, “Multiple Kernel Learning” PPT, 2008
- MSU CSE 902, “Multiple Kernel Learning” PPT, 2014
- https://en.wikipedia.org/wiki/Positive-definite_kernel
- https://en.wikipedia.org/wiki/Reproducing_kernel_Hilbert_space