On the robust nonlinear curve fitting

Shotaro Akaho

The National Institute of Advanced Industrial Science and Technology (AIST)

2014/03/14

Motivation and Results

A large dimensional problem in a small dimensional space

- **Input vs feature space metric**: kernel method is a strong tool to deal with nonlinear problems by linear methods, but metric structure of input space is broken
  $\Rightarrow$ General framework to incorporate input space metric
- **Robustness**: L_p regularization ($p \leq 1$) is popular for the sparseness, but we focus more on L_p cost function for robustness
  $\Rightarrow$ Sparse property of the optimal solution
- **PCA vs MCA**: kernel PCA does not always give satisfiable results
  $\Rightarrow$ Comparative results (discussion) of kernel PCA and MCA

Acknowledgement

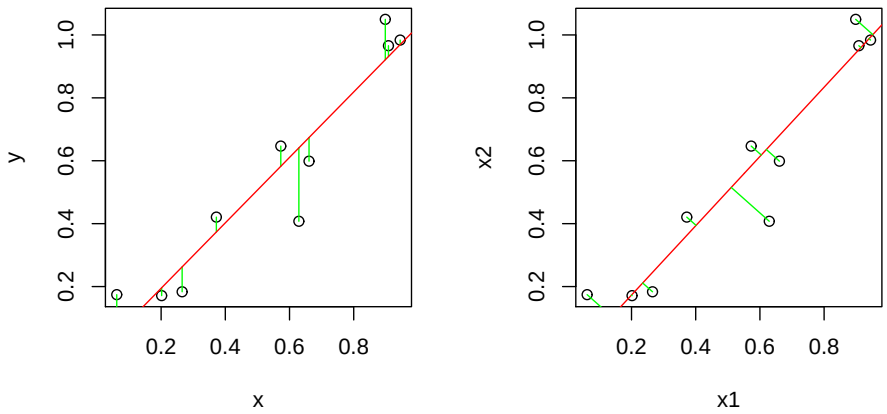
A part of the work in this presentation is a joint work with Jun Fujiki (Fukuoka univ.) Hideitsu Hino (Tsukuba univ.) and Noboru Murata (Waseda univ.), as well as informal discussion with Kenji Fukumizu (ISM)

Table of contents

Table of contents

Fitting methods

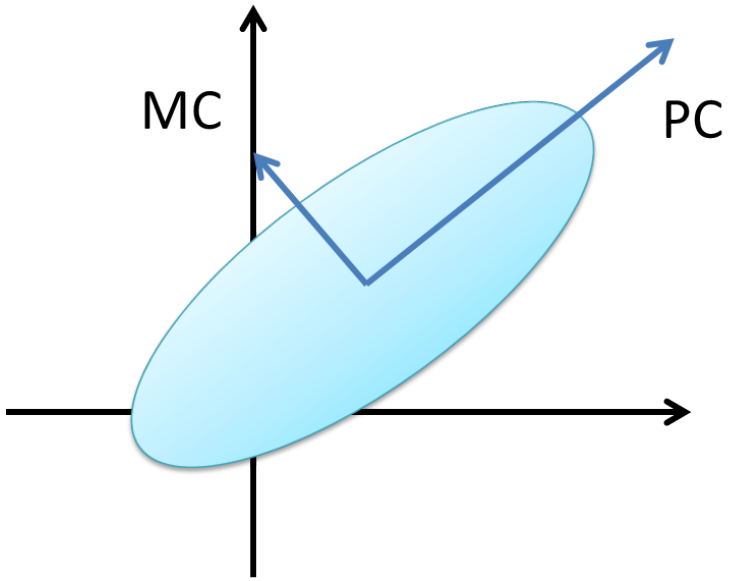
- **Purpose of fitting**: object recognition, denoising etc.
- There are (at least) two kinds of line/hypersurface fitting to sample points
  - **Regression** $y = f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$
    Distinctive treatment between y and $\mathbf{x}$
    Minimization of $E[(y - f(\mathbf{x}))^2]$
  - **Dimension reduction** $\mathbf{a}^T\mathbf{x} = 0$
    All components of $\mathbf{x}$ are treated equally
    Minimization of distance between points and line



Fitting by dimension reduction

- Minimization of distance between points and subspace (MCA)
- Equivalently, find the subspace that preserves variance of data points as much as possible (PCA)
- Equivalence of Principal Component Analysis (PCA) and Minor Component Analysis (MCA)
- Solution is obtained by solving eigenvalue problem

$$(X^T X)\mathbf{a} = \lambda \mathbf{a}$$



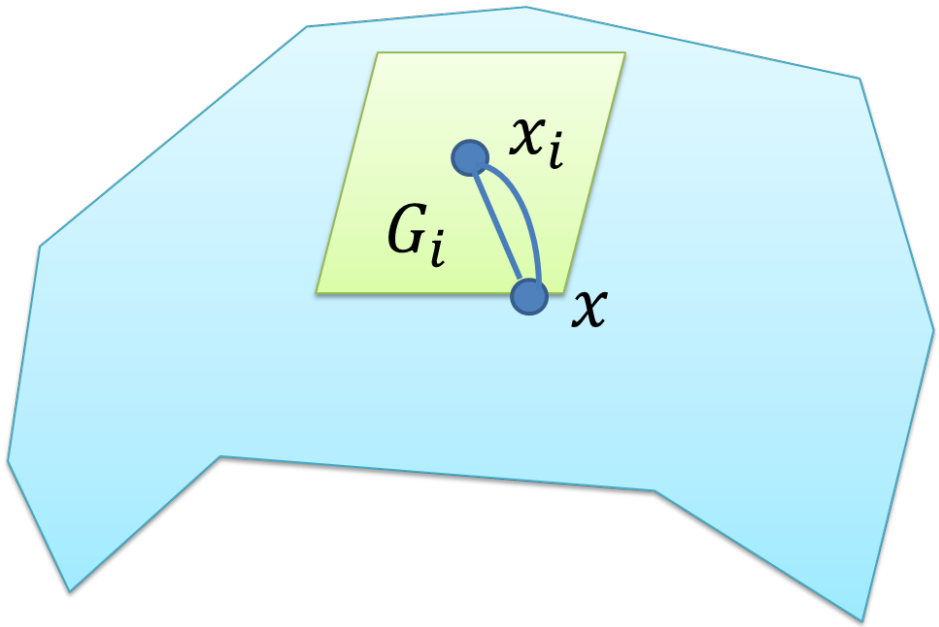
Extension to Riemannian space

In many applications, we need to consider non-Euclidean space

- Hypersphere (directional data, geology, economics; Fujiki+2007)
- Grassmann-Stiefel manifold (subspace data, independent component analysis; Nishimori+2005)
- Statistical manifold (statistics, optimization, control; Akaho2004)

Extension to Riemannian space

- Riemannian space with metric $G(\mathbf{x})$
- Extension to distance to the length of geodesic (hard to evaluate)
- Local approximation by the norm on the tangent space at $\mathbf{x}_i$
  $\|\mathbf{x} - \mathbf{x}_i\|^2_{G_i} = (\mathbf{x} - \mathbf{x}_i)^T G_i (\mathbf{x} - \mathbf{x}_i)$, $G_i = G(\mathbf{x}_i)$



- In spite of this approximation, dimension reduction problem cannot be solved by a simple eigenvalue problem.

Table of contents

Quadratic curve fitting

- Quadratic curve fitting

$$a_0 + a_1 x_1 + a_2 x_2 + a_3 x_1^2 + a_4 x_1 x_2 + a_5 x_2^2 = 0$$

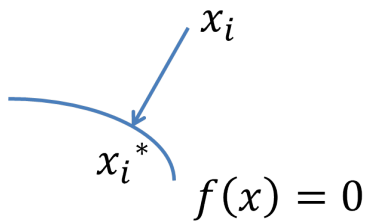
- Feature map reduces the nonlinear problem to linear problem
  $\mathbf{x} \mapsto \phi(\mathbf{x}) = (1, x_1, x_2, {x_1}^2, x_1 x_2, {x_2}^2)$
- Linear fitting on 6 dimensional feature space

Input space versus feature space

- Linear method on feature space minimizes the distance in feature space.
- However, the distance is not equal to the distance between points and the curve in the input space
- There are not many researches focusing on input space (Schölkopf 1999)

Locally linear approximation of function

- Projection from a point to a curve is difficult in general (local minimum/maximum, saddle)



- Linear approximation of $f(\mathbf{x}_i^*) = 0$ around $\mathbf{x}_i$ (Akaho1993)

$$0 = f(\mathbf{x}_i^*) \simeq f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T (\mathbf{x}_i^* - \mathbf{x}_i)$$

- The closest point $\mathbf{x}_i^*$ to $\mathbf{x}_i$ w.r.t. metric G_i, satisfying the constraint above is given by

$$\|\mathbf{x}_i^* - \mathbf{x}_i\|_{G_i}^2 = \frac{f(\mathbf{x}_i)^2}{\|\nabla f(\mathbf{x}_i)\|_{G_i^{-1}}^2}$$

Locally linear approximation of function

- Applying to $f(\mathbf{x}) = \mathbf{a}^T \phi(\mathbf{x})$ leads to the sum of input space distance

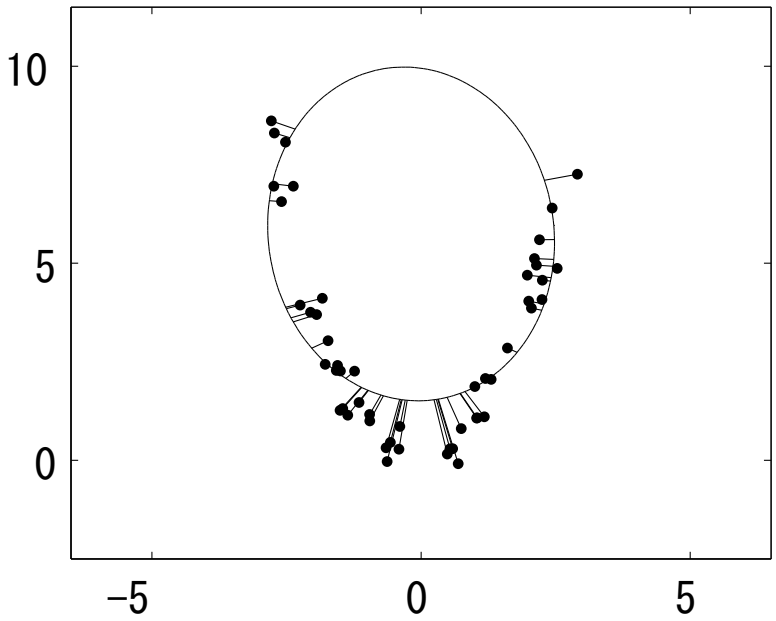
$$\sum_i \|\mathbf{x}_i^* - \mathbf{x}_i\|_{G_i}^2 = \sum_i \frac{\mathbf{a}^T \phi_i \phi_i^T \mathbf{a}}{\mathbf{a}^T \nabla \phi_i G_i^{-1} \nabla \phi_i^T \mathbf{a}}, \quad \phi_i = \phi(\mathbf{x}_i)$$

- Sum of ratio of quadratic forms
- Successive iteration method

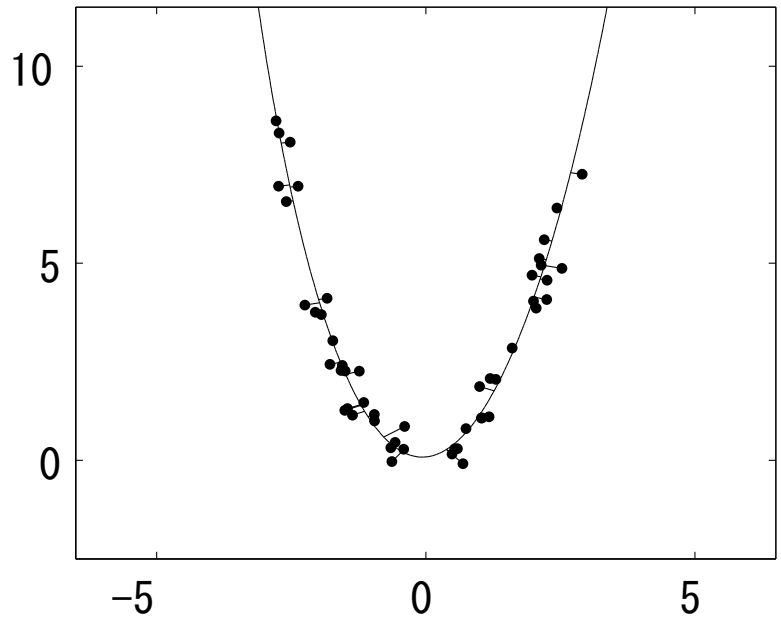
$$\mathbf{a}_{t+1} = \underset{\|\mathbf{a}\|=1}{\operatorname{argmin}} \mathbf{a}^T \left[\sum_i \frac{1}{w_i} \phi_i \phi_i^T \right] \mathbf{a}, \quad w_i = \mathbf{a}_t^T \nabla \phi_i G_i^{-1} \nabla \phi_i^T \mathbf{a}_t,$$

Example

$$u \sim \mathrm{u}[-2, 2], \quad x_1 = u + \epsilon_1, \quad x_2 = u^2 + \epsilon_2$$



Feature space



Input space

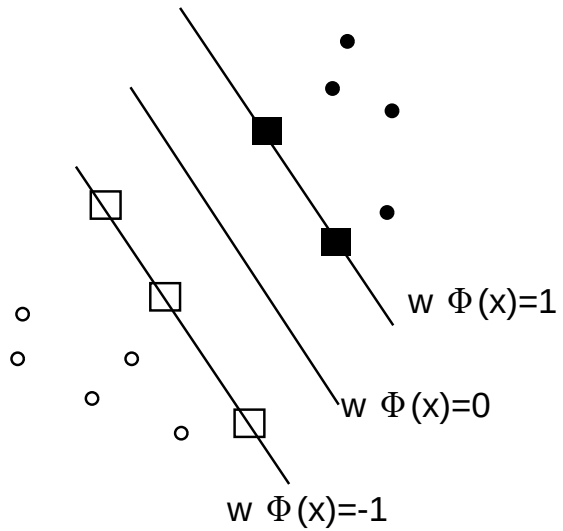
Table of contents

Toward higher dimensions

- Higher dimensional case (e.g. Reproducing Kernel Hilbert Space)
- First, we consider support vector machine (not curve fitting, but finds optimal separating hypersurface)
- Fitting problem is discussed later

Support vector machine

- SVM finds an optimal hyperplane that maximizes margin in the feature space
- Maximize the margin in the input space by the approximation of distance in the input space
- Approach: the same formulation as conventional SVM except introducing a linear approximation of distance (+ additional linear expansions; Akaho2004)

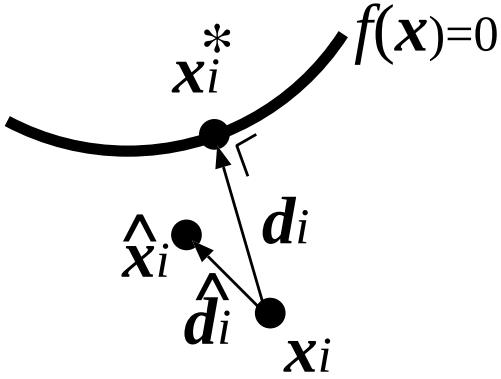


Approximation of distance

- Approximate the distance not around $\mathbf{x}_i$ but a better estimate $\hat{\mathbf{x}}_i$

$$\|\mathbf{d}_i\|^2_{G_i} = \frac{(\mathbf{a}^T\{\phi(\hat{\mathbf{x}}_i) - \nabla\phi(\hat{\mathbf{x}}_i)\hat{\mathbf{d}}_i\})^2}{\|\mathbf{a}^T\nabla\phi(\hat{\mathbf{x}}_i)\|^2_{G_i^{-1}}}$$

$$\mathbf{d}_i = \mathbf{x}_i^* - \mathbf{x}_i, \quad \hat{\mathbf{d}}_i = \hat{\mathbf{x}}_i - \mathbf{x}_i$$



- $\hat{\mathbf{x}}_i$ is initialized by $\mathbf{x}_i$ and it can be iteratively improved (discussed later)

Margin constraint

- Separating hyperplane is invariant under scalar transformation of $\mathbf{a}$, so we assume

$$\min_i \|\mathbf{d}_i\|_{G_i}^2 = \frac{1}{\|\mathbf{a}\|^2}$$

- Then maximizing margin is equivalent to minimizing $\|\mathbf{a}\|^2$ (quadratic regularization) under the above constraints with sign

$$y_i \frac{\mathbf{a}^T\{\phi(\hat{\mathbf{x}}_i) - \nabla\phi(\hat{\mathbf{x}}_i)\hat{\mathbf{d}}_i\})}{\|\mathbf{a}^T\nabla\phi(\hat{\mathbf{x}}_i)\|_{G_i^{-1}}^2} \geq \frac{1}{\|\mathbf{a}\|}$$

Linearization of constraints

- Suppose an approximate solution $\hat{\mathbf{a}}$ is given, we approximate the constraint by linear inequality of $\mathbf{a}$

$$\mathbf{a}^T[y_i\{\phi(\hat{\mathbf{x}}_i) - \nabla\phi(\hat{\mathbf{x}}_i)\hat{\mathbf{d}}_i\} - \hat{\boldsymbol{\eta}}_i] \geq \hat{g}_i$$

  where $\hat{g}_i$: scalar function, $\hat{\boldsymbol{\eta}}_i$: linear function of $\nabla\phi(\hat{\mathbf{x}}_i)$ and $\hat{\mathbf{a}}$

- Quadratic optimization with linear constraint leads to

$$L(\mathbf{a}) = \mathbf{a}^T\mathbf{a} - \sum_{i=1}^{n} \alpha_i \left(\mathbf{a}^T[y_i\{\phi(\hat{\mathbf{x}}_i) - \nabla\phi(\hat{\mathbf{x}}_i)\hat{\mathbf{d}}_i\} - \hat{\boldsymbol{\eta}}_i] - \hat{g}_i\right)$$

  where α_i is a Lagrange multiplier

Support Vectors

- Differentiating $L(\mathbf{a})$ by $\mathbf{a}$, we have

$$\mathbf{a} = \sum_{i=1}^{n} \alpha_i [y_i \{\phi(\hat{\mathbf{x}}_i) - \nabla\phi(\hat{\mathbf{x}}_i)\hat{\mathbf{d}}_i\} - \hat{\boldsymbol{\eta}}_i]$$

- Sparsity (some α_i are exactly 0 as in SVM)
- Classification function $f(\mathbf{x}) = \mathbf{a}^T\phi(\mathbf{x})$:
  if we assume $\hat{\mathbf{a}}$ is a linear function of $\phi(\hat{\mathbf{x}}_i)$ and $\nabla\phi(\hat{\mathbf{x}}_i)$,

$$f(\mathbf{x}) = \sum_i \{a_i \phi(\hat{\mathbf{x}}_i)^T \phi(\mathbf{x}) + b_i \nabla\phi(\hat{\mathbf{x}}_i)^T \phi(\mathbf{x})\}$$

(Since $\hat{\boldsymbol{\eta}}_i$ is a linear function of $\hat{\mathbf{a}}$ and $\phi(\hat{\mathbf{x}}_i)$)

Kernelization

- Reproducing kernel Hilbert space
- Kernel function (strictly it should be written as an inner product)

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

- Derivative of kernel function

$$\nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') = \nabla \phi(\mathbf{x})^T \phi(\mathbf{x}'),$$

$$\nabla_{\mathbf{x}} \nabla_{\mathbf{x}'} k(\mathbf{x}, \mathbf{x}') = \nabla \phi(\mathbf{x})^T \nabla \phi(\mathbf{x}')$$

Kernel trick

- Kernel trick enables us to choose positive semidefinite function as a kernel function instead of taking inner product in the feature space
- Example (Gaussian kernel)
  - $k(\mathbf{x}, \mathbf{x}') = \exp(-\beta\|\mathbf{x} - \mathbf{x}'\|^2)$
  - $\nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') = -2\beta k(\mathbf{x}, \mathbf{x}')(\mathbf{x} - \mathbf{x}')$,
  - $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}'} k(\mathbf{x}, \mathbf{x}') = 2\beta k(\mathbf{x}, \mathbf{x}')(I - 2\beta(\mathbf{x} - \mathbf{x}')(\mathbf{x} - \mathbf{x}')^T)$,
- Kernel version of classification function

$$f(\mathbf{x}) = \sum_i \{a_i k(\hat{\mathbf{x}}_i, \mathbf{x}) + b_i \nabla k(\hat{\mathbf{x}}_i, \mathbf{x})\}$$

Estimation of $\hat{x}_i$

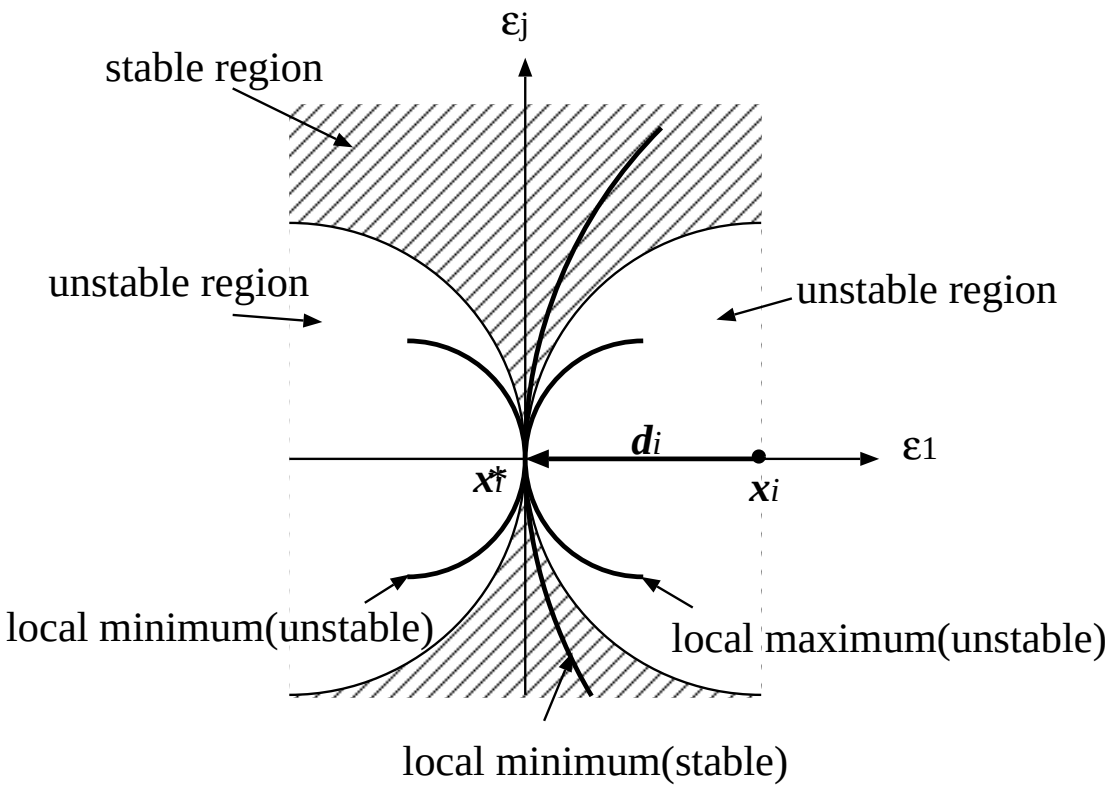
- $\hat{\mathbf{x}}_i$ can be improved for a fixed parameter of curve $\hat{\mathbf{a}}$

$$\mathbf{x}_i^{[l+1]} = h(\mathbf{x}_i^{[l]}, \hat{\mathbf{a}}, \mathbf{x}_i, G_i)$$

- It may converge to a local minimum, a local maximum, or a saddle point. How about the convergence property?
- **Property of the iterative solution**: Let $\mathbf{x}_i^*$ be an equilibrium state of the iteration step, it is a critical point of the projection point (a local minimum/maximum, or saddle point). If it is a local maximum or saddle, the algorithm is always unstable

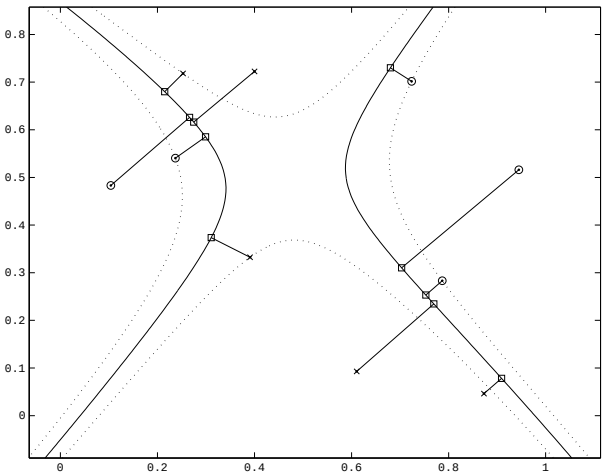
Estimation of $\hat{x}_i$

For a local minimum, the curvature of the hypersurface determines the stability (we can slow down the iteration step to avoid the instability)

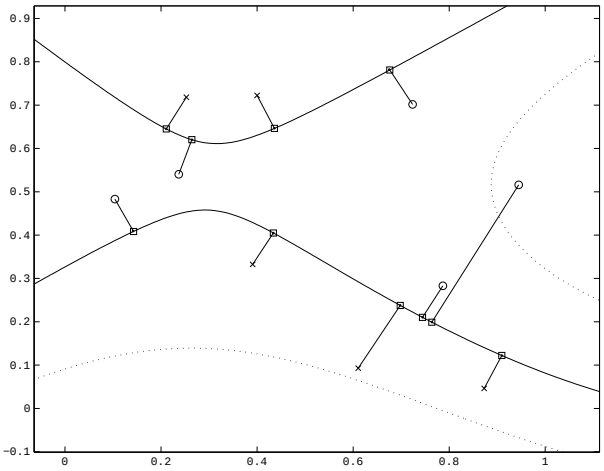


Example

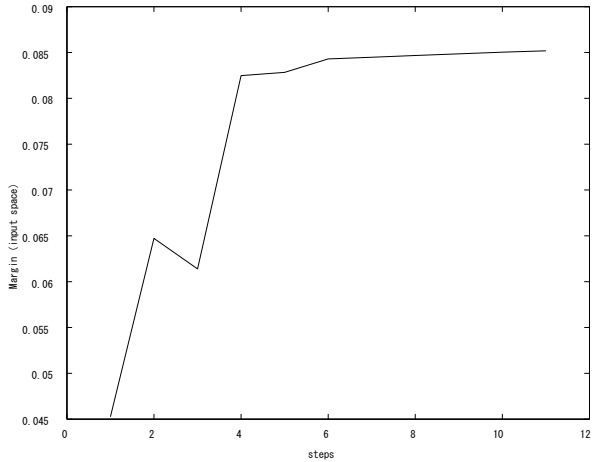
- Through experiments for synthetic and some benchmark datasets, we showed that the input space margin and generalization performance increased by a proposed method



Original



Input space



Margin

Table of contents

Robustness and sparseness

- Typical optimization formulation for fitting

$$E = \sum_i \mathrm{Cost}(\mathbf{x}_i, f) + \lambda \mathrm{Reg}(f)$$

- Regularization: we don't discuss in detail here
  SVM: L_2, Lasso: L_1 (sparse prior)
- Cost:
  -SVM: Hinge loss of $[yf(\mathbf{x}) - 1]_+$
  -Ordinary regression: Quadratic loss $(y - f(\mathbf{x}))^2$
  -Fitting by dimension reduction: Quadratic (vertical) distance
  -Here: p-th power (vertical) distance (robust + sparseness)

p-th power deviation

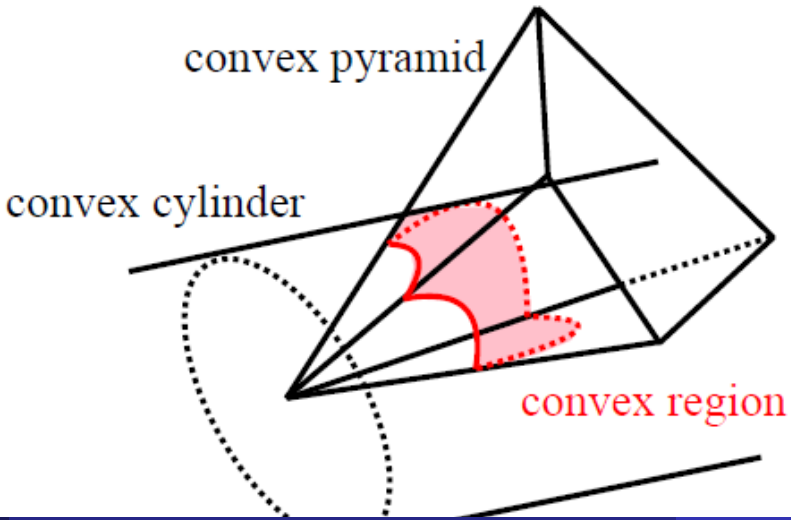
- Euclidean distance from a d-dim point $\mathbf{x}_i$ to a hyperplane $\mathbf{a}^T\mathbf{x} + a_0 = 0, \|\mathbf{a}\| = 1$ is given by $|\mathbf{a}^T\mathbf{x}_i + a_0|$
- Minimizing the mean p-th power of the Euclidean distance

$$R_p(\mathbf{a}) = \sum_{i=1}^{n} w_i |\mathbf{a}^T\mathbf{x}_i + a_0|^p$$

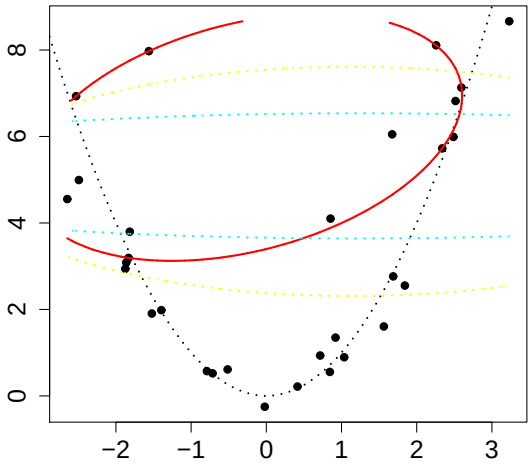
- R_1 norm (Ding2006)
- R_0 is meaningless (the same cost values)
- We can prove L_p $0 < p \leq 1$ case is completely sparse (the optimal hyperplane passes through d points)

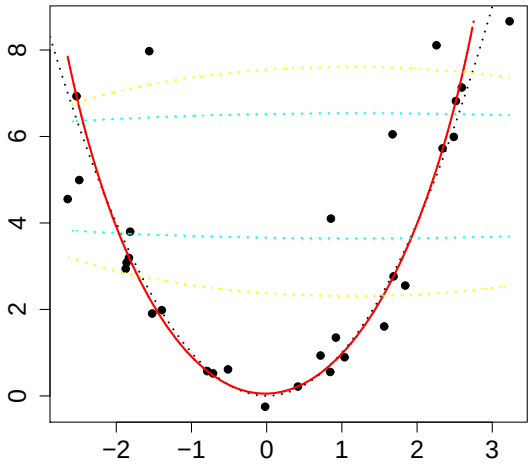
Proof sketch

- The parameter $(\mathbf{a}^T, a_0)$ is on a cylinder $Q = S^d \times R$: convex in R^{d+1} against the origin
- Parameter space is devided by n hyperplanes $\mathbf{a}^T\mathbf{x}_i + a_0 = 0$, each P is convex in R^{d+1}. $\Rightarrow P \cap Q$ is convex against the origin
- The (weighted) p-th deviation takes concave contour against origin
- The minimum of the objective function is obtained in the boundary of $P \cap Q$
- The procedure is performed recursively, and the (local) minimum is obtained at one of the vertex of P



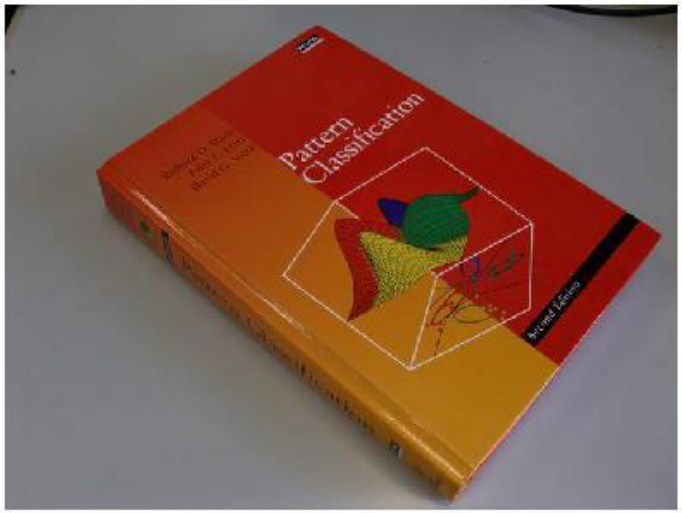
Example



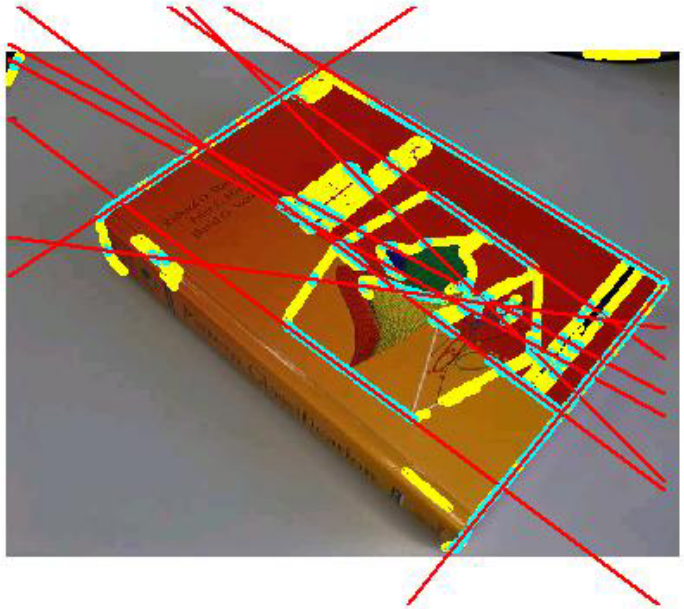


Feature space ($p = 0.5$) Input space ($p = 0.5$)
(Yellow (Feature space $p = 2$) and Blue (Input space $p = 2$))

Example



Example



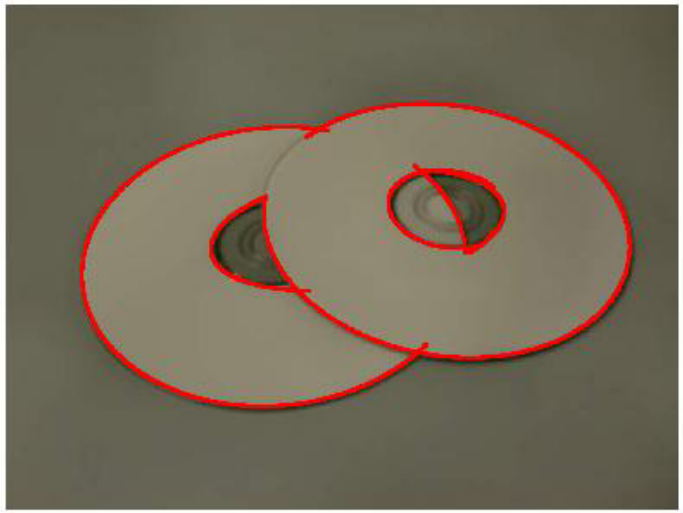


Table of contents

Curve fitting in RKHS

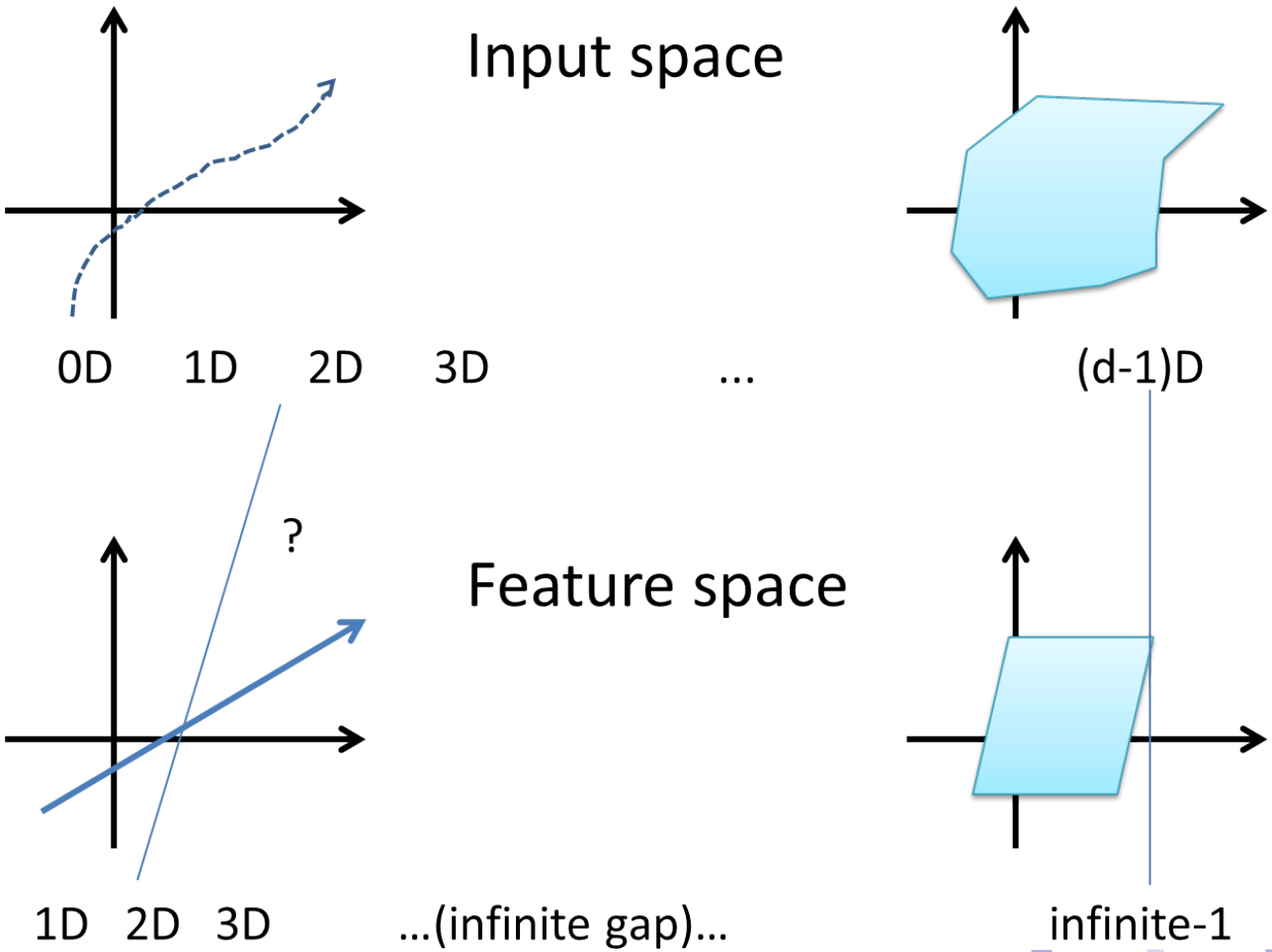
- We want to extend the curve fitting in high dimensional space
- PCA vs MCA (in low dimensional case, it is almost equivalent)
- In infinite dimensional case, the difference taking higher eigenvalues and discarding lower eigenvalues is large
- Note: finding a dimension reduction map $u = f(x)$ and finding a low dimensional structure $x' = g(x)$ in the input space are slightly different

Minor component analysis

- In the first part of my talk, we considered the hyperplane $\mathbf{a}^T \phi(\mathbf{x}) = 0$ which is obtained by MCA
- MCA is attractive because to find the optimal hyperplane was somewhat reasonable in SVM
- But in MCA case, it is hard because of too much degree of freedom

PCA vs MCA

- The image of input space in feature space is very sparse
- PCA (small number of projection axes)
- MCA (small number of noise reduction axes)



Two kinds of regularization

- Tikhonov (SVM)

$$\min_f \sum_i \mathrm{Cost}(f(\mathbf{x}_i)) + \lambda\Omega(\|f\|)$$

  $f \in$ RKHS, Ω is a nondecreasing function

- Ivanov-like (PCA, MCA) not precisely Ivanov

$$\min_f \sum_i \mathrm{Cost}(f(\mathbf{x}_i)) \quad \text{s.t.} \quad \|f\| = \text{const}$$

Representer theorem

- Representer theorem is a key theorem for RKHS, which reduces the dimensionality from infinite to finite
- **Representer theorem 1**: As for Tikhonov regularization

$$\min_f \sum_i \mathrm{Cost}(f(\mathbf{x}_i)) + \lambda\Omega(\|f\|),$$

For any Cost function, the optimal f is in the form

$$f_{opt}(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

Ivanov-like

- **Representer theorem 2**: As for Ivanov-like regularization

$$\min_f \sum_i \mathrm{Cost}(f(\mathbf{x}_i)) \quad \text{s.t.} \quad \|f\| = \text{const}$$

If the Cost function satisfies decreasing property ($Cost(c_1 f(\mathbf{x})) \leq Cost(c_2 f(\mathbf{x}))$ when $c_1 \geq c_2 > 0$), the optimal f is in the form

$$f_{opt}(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

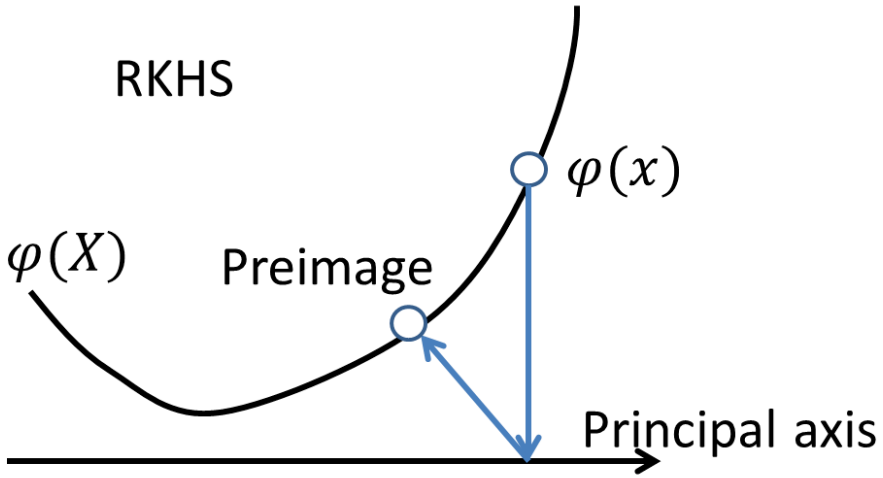
- PCA (minus of variance of $f(\mathbf{x}_i)$) satisfies, but MCA (variance of $f(\mathbf{x}_i)$) does not! Even data out of samples can be good for MCA.

Approach from PCA

- Kernel PCA does not work as one would expect (cf. manifold learning)
- For noise reduction, we need to solve "preimage problem" that is difficult to solve
- Even if preimage is found, it is not always what we want (non-smooth, more dimension is needed)
- The structures of projection and preimage are different in nonlinear case

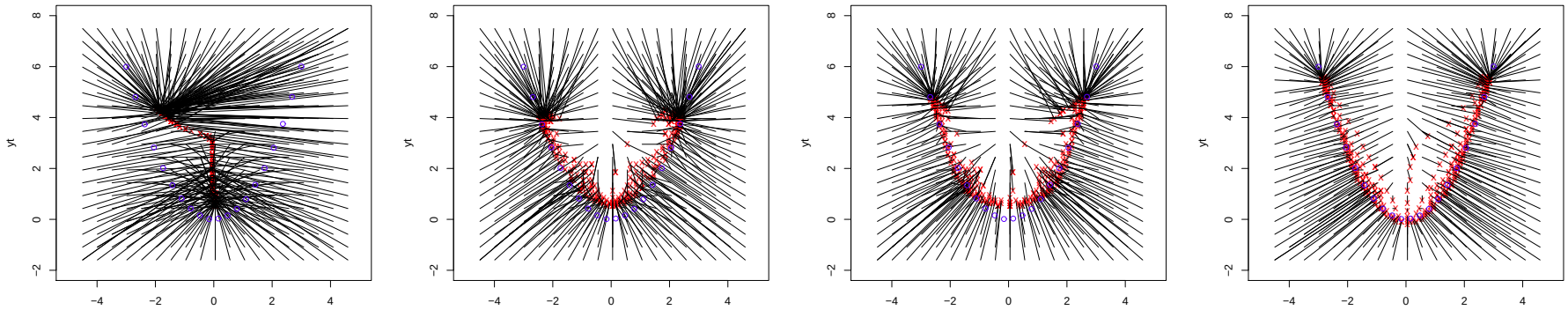
Preimage

Finding a dimension reduction map $u = f(x)$ and finding a low dimensional structure $x' = g(x)$ in the input space are slightly different in nonlinear case



Example

Preimage is somewhat unstable and not smooth

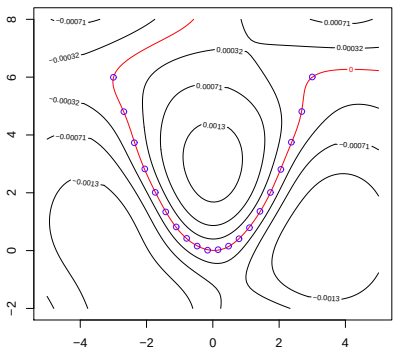
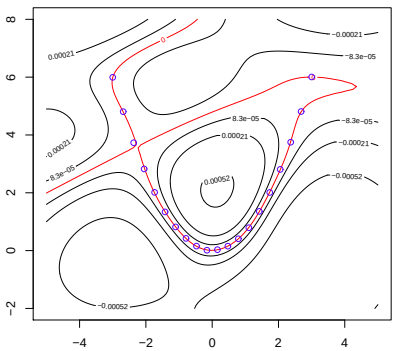
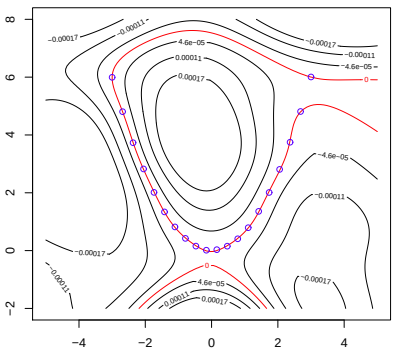
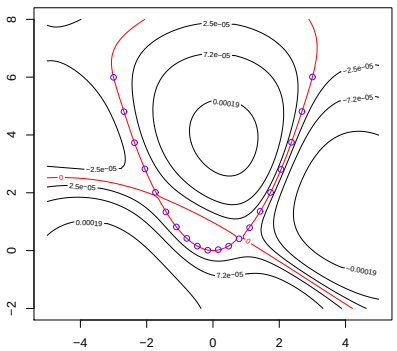


Approach from MCA

- Kernel MCA achieves smoother curve/surface
- Too many freedom (even in the sample space)

Example

MCA gives too many good curves! (their linear combination as well; Fujiki+2013)
Further out of sample points will increase the freedom
Does sparsity help to choose a good curve? $\rightarrow$ open

Concluding remarks

- The method to incorporate input space metric is proposed
- Finding the projection point to hypersurface is more stable than finding the preimage
- L_p cost function is effective when many outliers exist
- MCA has a potential to give a good curve fitting, but choosing a good curve obtaining good generalization performance is not established