Smooth Support Vector Machines for Classification and Regression

Lee, Yuh-Jye

National Taiwan University of Science and Technology

Joint work with Olvi Mangasarian, W.-F. Hsieh, C.-M. Huang, and Sun-Yun Huang

Research Seminar “Mathematical Statistics”
Humboldt University, Berlin, Germany

January 24, 2007

Outline

- Binary classification problem
- Conventional Support Vector Machines
- Kernel trick and nonlinear SVM
- SSVM: Smooth Support Vector Machines
  - For classification and regression problems
- Newton Armijo algorithm for SSVMs
  - A global convergent algorithm at a quadratic rate
- Reduced Support Vector Machines:
  - Deal with massive datasets
- Conclusions

Binary Classification Problem

(A Fundamental Problem in Data Mining)

- Find a decision function (classifier) to discriminate two categories data sets.
- Supervised learning in Machine Learning
  - Decision Tree, Neural Network, k-NN and Support Vector Machines, etc.
- Discrimination Analysis in Statistics
  - Fisher Linear Discriminator
- Successful applications:
  - Marketing, Bioinformatics, Fraud detection

Binary Classification Problem

Given a training dataset

$$S = \{(x^i, y_i) \big| x^i \in R^n, y_i \in \{-1, 1\}, i = 1, \ldots, m\}$$

$$x^i \in A_+ \Leftrightarrow y_i = 1 \;\; \& \quad x^i \in A_- \Leftrightarrow y_i = -1$$

Main goal:

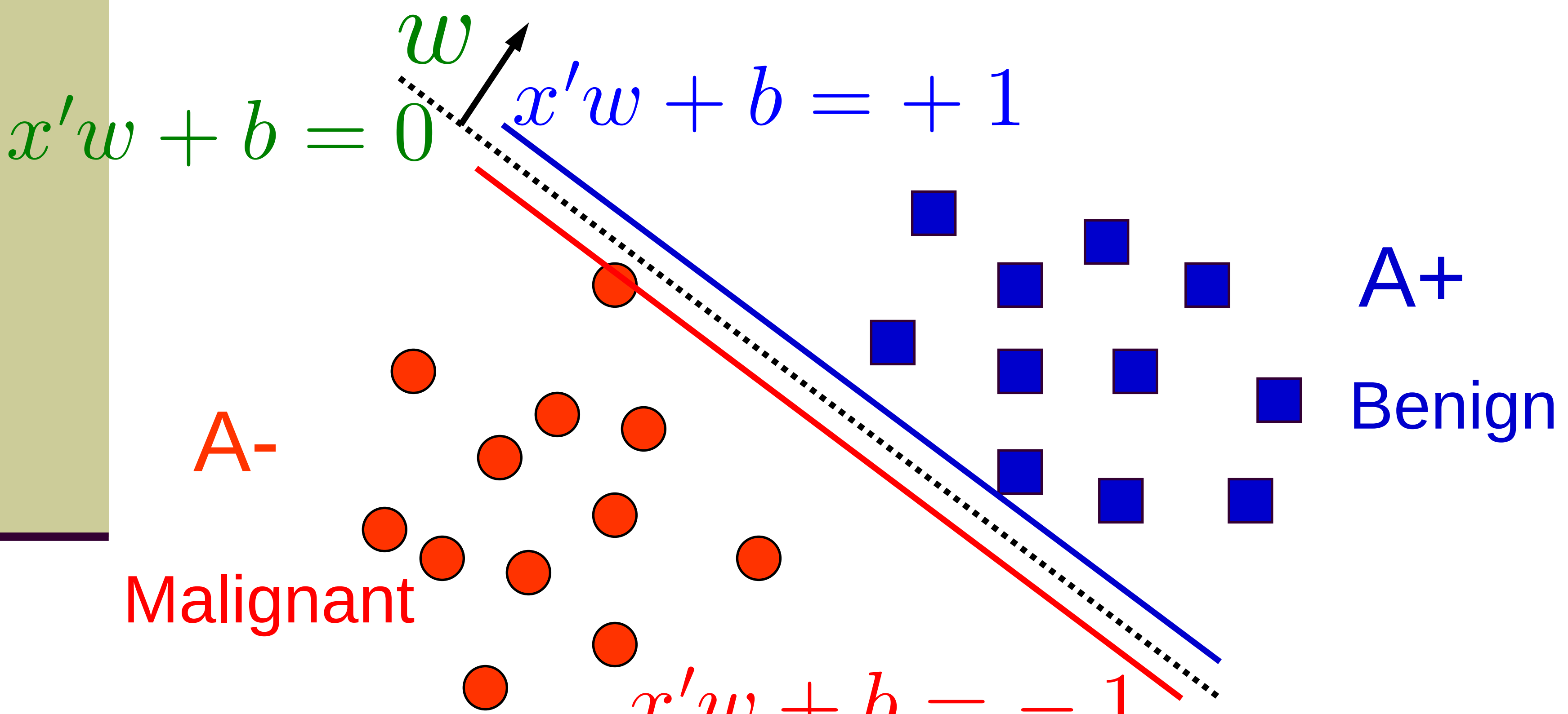
Predict the unseen class label for new data

Find a function $f : R^n \to R$ by learning from data

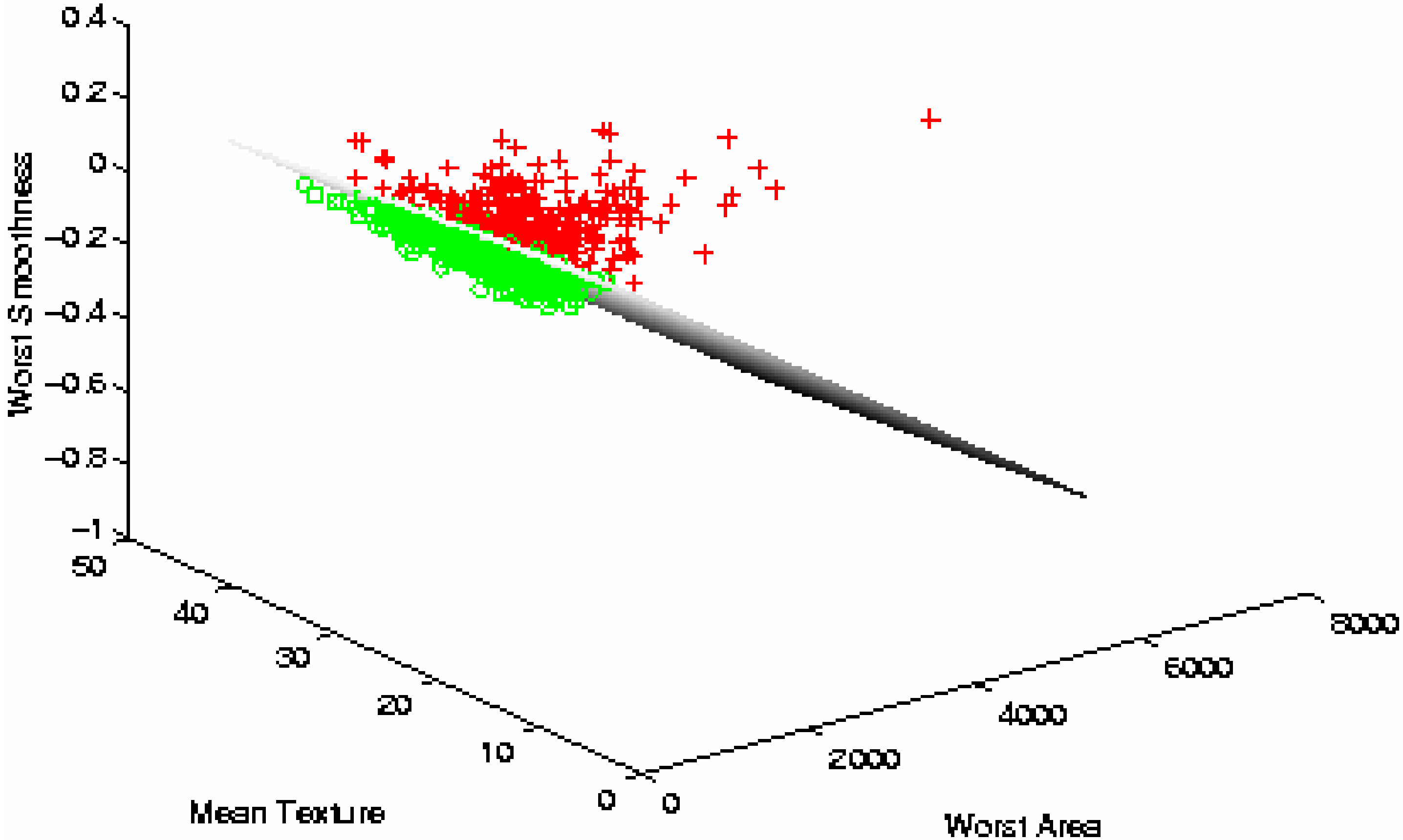
$$f(x) \geqslant 0 \Rightarrow x \in A_+ \;\; and \quad f(x) < 0 \Rightarrow x \in A_-$$

The simplest function is linear: $f(x) = w'x + b$

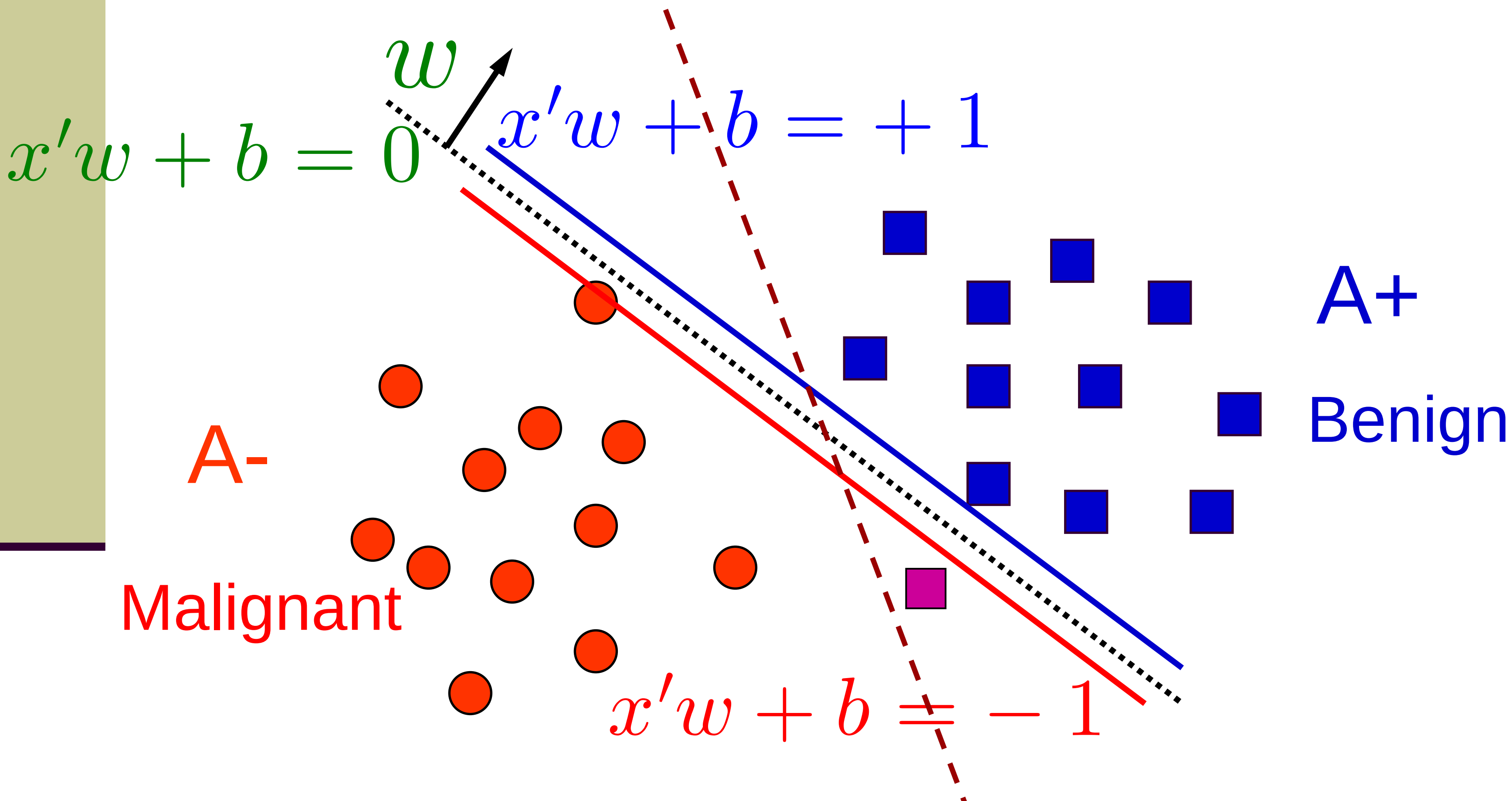
Binary Classification Problem
Linearly Separable Case



Breast Cancer Diagnosis Application
97% Tenfold Cross Validation Correctness
494 Benign, 286 Malignant

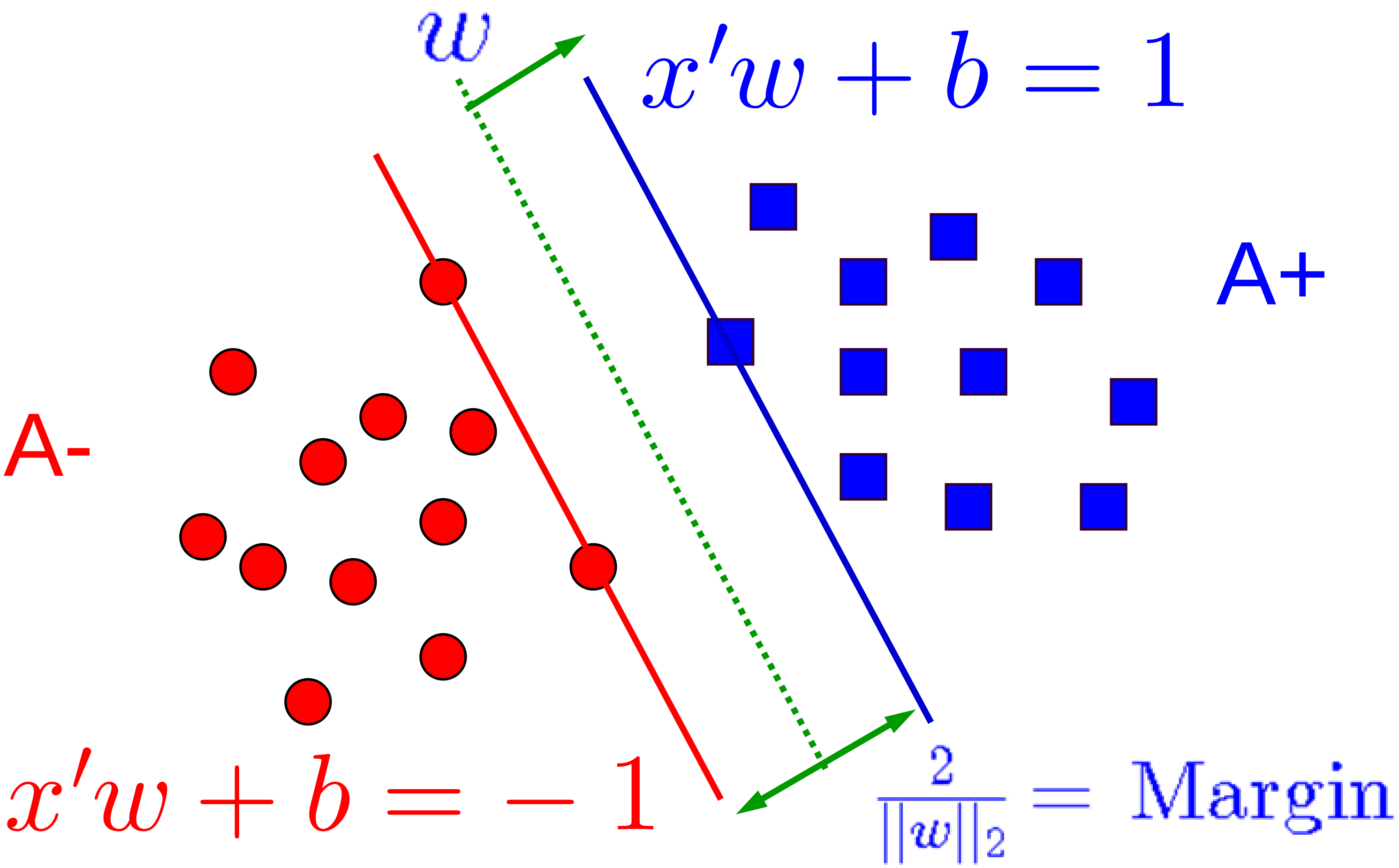


Binary Classification Problem
Linearly Separable Case



Support Vector Machines

Maximizing the Margin between Bounding Planes



Why Use Support Vector Machines?

Powerful tools for Data Mining

- SVM classifier is an optimally defined surface
- SVMs have a good geometric interpretation
- SVMs can be generated very efficiently
- Can be extended from linear to nonlinear case
  - Typically nonlinear in the input space
  - Linear in a higher dimensional "feature space"
  - Implicitly defined by a kernel function
- Have a sound theoretical foundation
  - Based on Statistical Learning Theory

Summary of Notations

Let $S = \{(x^1, y_1), (x^2, y_2), \ldots (x^m, y_m)\}$ be a training dataset and represented by matrices

$$A = \begin{bmatrix} (x^1)' \\ (x^2)' \\ \vdots \\ (x^m)' \end{bmatrix} \in R^{m \times n}, \; D = \begin{bmatrix} y_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & y_m \end{bmatrix} \in R^{m \times m}$$

$$\begin{array}{rcrll} A_i w + b & \geqslant & +1, & for & D_{ii} = +1, \\ A_i w + b & \leqslant & -1, & for & D_{ii} = -1 \end{array}$$

equivalent to

$D(Aw + \mathbf{1}b) \geqslant \mathbf{1}$, where $\mathbf{1} = [1, 1, \ldots, 1]' \in R^m$.

Support Vector Machine Formulations

(Two Different Measures of Training Error)

2-Norm Soft Margin (Primal form):

$$\min_{(w,b,\xi)\in R^{n+1+m}} \quad \frac{1}{2}||w||_2^2 + \frac{C}{2}||\xi||_2^2$$

$$D(Aw + \mathbf{1}b) + \xi \geqslant \mathbf{1}$$

1-Norm Soft Margin (Primal form):

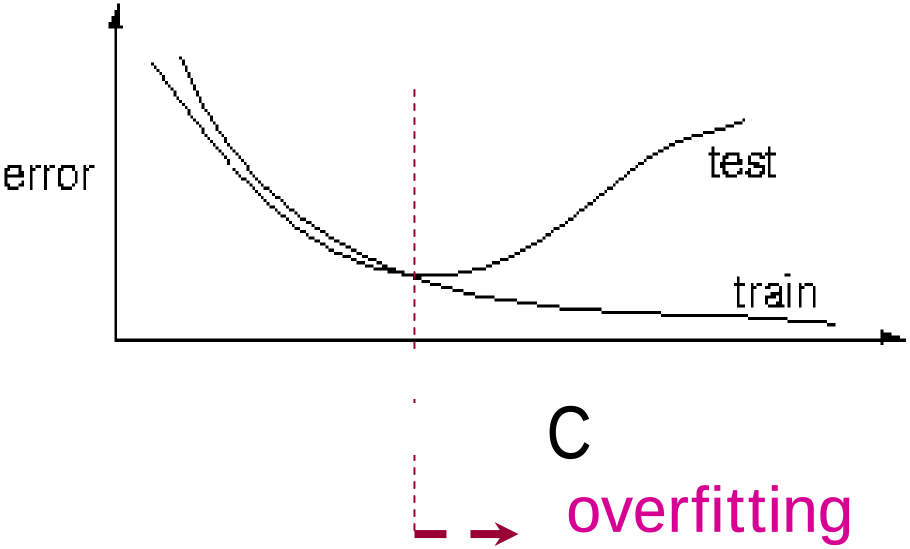
$$\min_{(w,b,\xi)\in R^{n+1+m}} \quad \frac{1}{2}||w||_2^2 + C\mathbf{1}'\xi$$

$$D(Aw + \mathbf{1}b) + \xi \geqslant \mathbf{1},\ \xi \geqslant \mathbf{0}$$

- Margin is maximized by minimizing reciprocal of margin.

Tuning Procedure

How to determine C?



The final value of parameter is the one with the maximum testing set correctness !

Support Vector Machine in Dual Form
(Motivation of the Kernel Trick)

1-Norm Soft Margin (Dual form):

$$\max_{\alpha \in R^m} \quad \mathbf{1}'\alpha - \tfrac{1}{2}\alpha' DAA'D\alpha$$
$$\mathbf{1}'D\alpha = 0,\ \mathbf{0} \leqslant \alpha \leqslant C\mathbf{1}$$

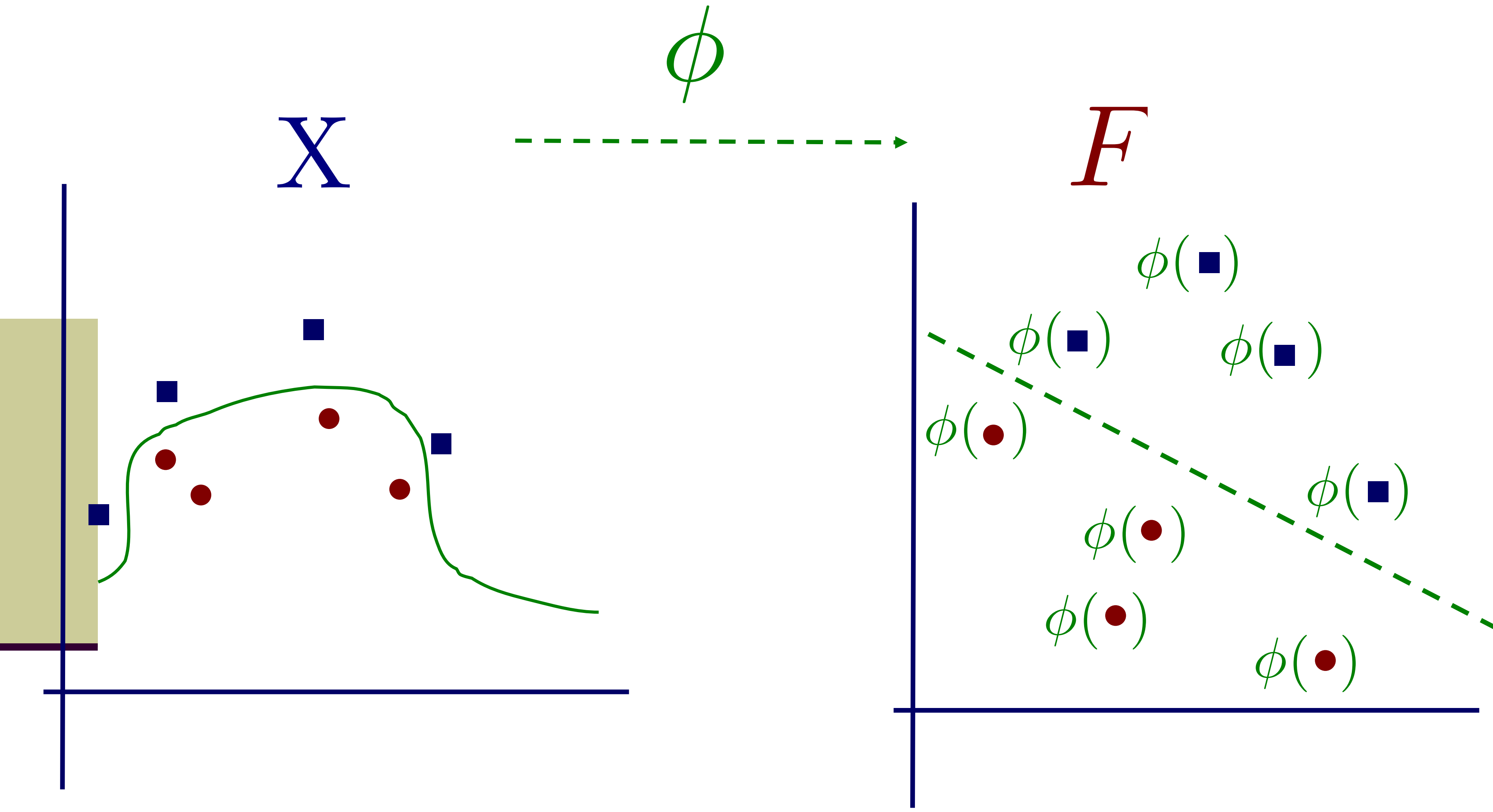
- The normal vector $w = A'D\alpha = \sum_{\alpha_j > 0}^{m} y_i \alpha_i A_i'$
- The *bias,* b is determined by *KKT* conditions
- The decision function (classifier)

$$f(x) = \alpha' DAx + b = \sum_{\alpha_i > 0}^{m} y_i \alpha_i (A_i x) + b$$

- All we need to know is the *inner products* of data

Two-spiral Dataset

(94 White Dots & 94 Red Dots)



ϕ
X
F

Kernel Technique

Based on Mercer's Condition (1909)

- The value of kernel function represents the inner product of two training points in feature space

- Kernel functions merge two steps
  1. map input data from input space to feature space (might be infinite dim.)
  2. do inner product in the feature space

Examples of Kernel

$$K(A,B): R^{m\times n} \times R^{n\times l} \longmapsto R^{m\times l}$$

$A \in R^{m\times n}, a \in R^m, \mu \in R,\ \ d$ is an integer:

- Polynomial Kernel: $(AA' + \mu aa')^d_{\bullet}$

  (Linear Kernel AA': $\mu = 0, d = 1$)

- Gaussian (Radial Basis) Kernel:

$$K(A,A')_{ij} = e^{-\mu\|A_i - A_j\|_2^2},\ \ i,j = 1,\ldots,m$$

➢ The ij-entry of $K(A,A')$ represents the "similarity" of data points A_i and A_j

Nonlinear Support Vector Machines

(Applying the Kernel Trick)

1-Norm Soft Margin Linear SVM:

$$\max_{\alpha \in R^m} \; \mathbf{1}'\alpha - \tfrac{1}{2}\alpha' DAA'D\alpha \;\; s.t. \;\; \mathbf{1}'D\alpha = 0, \; \mathbf{0} \leqslant \alpha \leqslant C\mathbf{1}$$

- Applying the kernel trick and running linear SVM in the feature space without knowing the nonlinear mapping

1-Norm Soft Margin Nonlinear SVM:

$$\max_{\alpha \in R^m} \; \mathbf{1}'\alpha - \tfrac{1}{2}\alpha' DK(A, A')D\alpha$$

$$s.t. \;\; \mathbf{1}'D\alpha = 0, \; \mathbf{0} \leqslant \alpha \leqslant C\mathbf{1}$$

- All you need to do is replacing AA' by $K(A, A')$

1-Norm SVM

(Different Measure of Margin)

1-Norm SVM:

$$\min_{(w,b,\xi)\in R^{n+1+m}} \quad ||w||_1 + C\mathbf{1}'\xi$$

$$D(Aw+\mathbf{1}b)+\xi \geqslant \mathbf{1}$$

$$\xi \geqslant \mathbf{0}$$

Equivalent to:

$$\min_{(s,w,b,\xi)\in R^{2n+1+m}} \quad \mathbf{1}s + C\mathbf{1}'\xi$$

$$D(Aw+\mathbf{1}b)+\xi \geqslant \mathbf{1}$$

$$-s \leqslant w \leqslant s$$

$$\xi \geqslant \mathbf{0}$$

Good for feature selection and similar to the LASSO

Smooth Support Vector Machines

SVM as an Unconstrained Minimization Problem

$$\min_{w,b} \frac{C}{2}\|\xi\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2) \qquad \text{(QP)}$$

$$\text{s. t.} \quad D(Aw + \mathbf{1}b) + \xi \geqslant \mathbf{1}$$

At the solution of (QP): $\xi = (\mathbf{1} - D(Aw + \mathbf{1}b))_+$

where $(\cdot)_+ = \max\{\cdot, 0\}$

Hence (QP) is equivalent to the nonsmooth SVM:

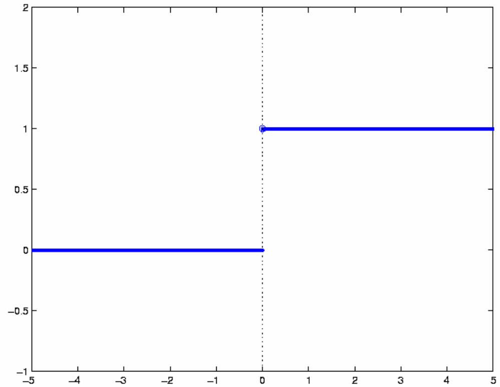
$$\min_{w,b} \frac{C}{2}\|(\mathbf{1} - D(Aw + \mathbf{1}b))_+\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2)$$

- Change (QP) into an unconstrained MP
- Reduce $(n+1+m)$ variables to $(n+1)$ variables

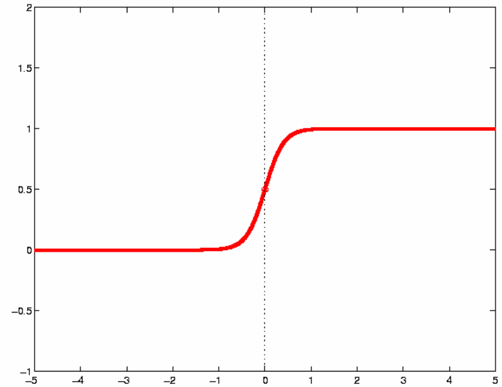
Smooth the Plus Function: Integrate $\frac{1}{(1+\varepsilon^{-\beta x})}$

$$p(x,\beta) := x + \frac{1}{\beta}\log(1+\varepsilon^{-\beta x})$$

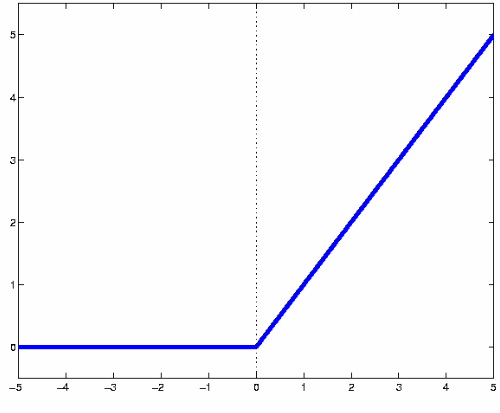
Step function: x_*



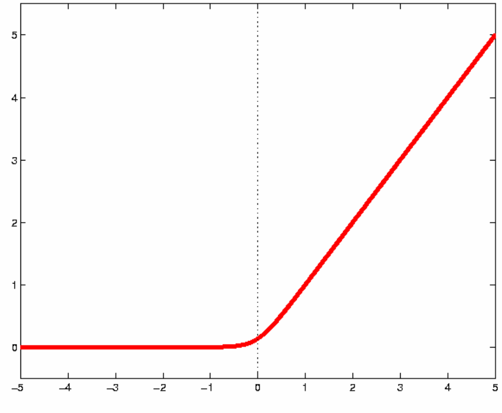
Sigmoid function: $\frac{1}{(1+\varepsilon^{-5x})}$



Plus function: x_+



p-function: $p(x,5)$



SSVM: Smooth Support Vector Machine

- Replacing the plus function $(\cdot)_+$ in the nonsmooth SVM by the smooth $p(\cdot\,,\beta)$, gives our SSVM:

$$\min_{(w,b)\in R^{n+1}} \frac{C}{2}\|p((\mathbf{1}-D(Aw+\mathbf{1}b)),\beta)\|_2^2+\frac{1}{2}(\|w\|_2^2+b^2)$$

- The solution of SSVM converges to the solution of nonsmooth SVM as β goes to infinity.

Newton-Armijo Method: Quadratic Approximation of SSVM

- The sequence $\{(w^i, b_i)\}$ generated by solving a quadratic approximation of SSVM, converges to the unique solution (w^*, b^*) of SSVM at a quadratic rate.
  - Converges in 6 to 8 iterations
- At each iteration we solve a linear system of:
  - n+1 equations in n+1 variables
  - Complexity depends on dimension of input space
- It might be needed to select a stepsize

Newton-Armijo Algorithm

$$\Phi_\beta(w,b) = \frac{C}{2}\|p((\mathbf{1}-D(Aw+\mathbf{1}b)),\beta)\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2)$$

Start with any $(w^0, b_0) \in R^{n+1}$. Having (w^i, b_i),

stop if $\nabla\Phi_\beta(w^i, b_i) = 0$, else :

(i) Newton Direction :

$$\nabla^2\Phi_\beta(w^i, b_i)d^i = -\nabla\Phi_\beta(w^i, b_i)'$$

(ii) Armijo Stepsize :

$$(w^{i+1}, b_{i+1}) = (w^i, b_i) + \lambda_i d^i$$

$$\lambda_i \in \{1, \tfrac{1}{2}, \tfrac{1}{4}, \ldots\}$$

such that Armijo's rule is satisfied

globally and quadratically converge to unique solution in a finite number of steps

Comparisons of SSVM with other SVMs

Tenfold test set correctness % (best in Red)

CPU time in *seconds*

Dataset Size $m \times n$	SSVM Linear Eqns.	$\mathrm{SVM}_{\Vert\cdot\Vert_1}$ LP	$\mathrm{SVM}_{\Vert\cdot\Vert_2^2}$ QP
Cleveland Heart 297 x 13	86.13 1.63	84.55 18.71	72.12 67.55
BUPA Liver 345 x 6	70.33 1.05	64.03 19.94	69.86 124.23
Ionosphere 351 x 34	89.63 3.69	86.10 42.41	89.17 128.15
Pima Indians 768 x 8	78.12 1.54	74.47 286.59	77.07 1138.0
WPBC(24 months) 155 x 32	83.47 2.32	71.08 6.25	82.02 12.50
WPBC(60 months) 110 x 22	68.18 1.03	66.23 3.72	61.83 4.91

Two-spiral Dataset

(94 White Dots & 94 Red Dots)

Nonlinear SVM Motivation

- Linear SVM: (Linear separator: $x'w + b = 0$)

$$\min_{\xi \geqslant 0,\, w,\, b} \frac{C}{2}\|\xi\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2) \qquad \text{(QP)}$$
$$\textbf{s. t.} \quad D(Aw + \mathbf{1}b) + \xi \geqslant \mathbf{1}$$

By QP "duality", $w = A'D\alpha$. Maximizing the margin in the "dual space" gives:

$$\min_{\xi \geqslant 0,\, \alpha,\, b} \frac{C}{2}\|\xi\|_2^2 + \frac{1}{2}(\|\alpha\|_2^2 + b^2)$$
$$\textbf{s. t.} \quad D(AA'D\alpha + \mathbf{1}b) + \xi \geqslant \mathbf{1}$$

- Dual SSVM with separator: $x'A'D\alpha + b = 0$

$$\min_{\alpha,\, b} \frac{C}{2}\|p(\mathbf{1} - D(AA'D\alpha + \mathbf{1}b), \beta)\|_2^2 + \frac{1}{2}(\|\alpha\|_2^2 + b^2)$$

Nonlinear Smooth SVM

Nonlinear Classifier: $K(x', A')D\alpha + b = 0$

- Replace AA' by a nonlinear kernel $K(A, A')$:

$$\min_{\alpha, b} \frac{C}{2}\|p(\mathbf{1} - D(K(A, A')D\alpha + \mathbf{1}b, \beta)\|_2^2 + \frac{1}{2}(\|\alpha\|_2^2 + b^2)$$

- Use Newton-Armijo algorithm to solve the problem
  - Each iteration solves m+1 linear equations in m+1 variables
- Nonlinear classifier depends on the data points with nonzero coefficients :

$$K(x', A')D\alpha + b = \sum_{\alpha_j \neq 0} \alpha_j y_j K(A_j, x) + b = 0$$

Remark on Nonlinear SVMs

Dual Form *vs.* Primal Form

- Nonlinear (Conventional) SVM in Dual form:

$$\max_{\alpha \in R^m} \ \mathbf{1}'\alpha - \frac{1}{2}\alpha' D K(A, A') D\alpha$$

$$\mathbf{1}' D\alpha = 0, \ \ \mathbf{0} \leqslant \alpha \leqslant C\mathbf{1}$$

O. L. Mangasarian
Generalized support vector machines.
Advances in Large Margin Classifiers, p.135-146,
MIT Press, Cambridge, MA, 2000

Brings things back to Primal form

$$\min_{\alpha, b, \xi} \frac{C}{2} ||\xi||_2^2 + \frac{1}{2}(||\alpha||_2^2 + b^2)$$

$$D(K(A, A')D\alpha + \mathbf{1}b) + \xi \geqslant \mathbf{1}$$

Multiclass Classification Problem

Consider the problem which given *m* training examples $(x_1, y_1), \ldots, (x_m, y_m)$, where $x_i \in R^n, i = 1, \ldots, m$ and $y_i \in \{1, \ldots, k\}$ is the class of x_i .

Main goal:

Predict the unseen class label for new data

Find *k* functions (classifiers) $f_j(x), \;\; j \in \{1, ..., k\}$ by learning form data.

$$f_j(x) \geqslant f_{j'}(x) \Rightarrow x \in \{class \;\; j\}, \;\; for \;\; all \;\; j' \neq j$$

The simplest function is linear: $f_j(x) = w_j' x + b_j$

MSSVM: Multiclass Smooth Support Vector Machine

- Single optimization formulation for Multiclass classification problem:

$$\min_{(w,b,\xi)\in R^{k(n+1+m)-m}} \frac{1}{2}\sum_{j=1}^{k}(w_j'w_j + b_j^2) + \frac{C}{2}\sum_{i=1}^{m}\sum_{j\neq y_i}(\xi_{ij})^2$$

$$subject\ \ to: \quad w_{y_i}'x_i + b_{y_i} \geq w_j'x_i + b_j + 1 - \xi_{ij}$$

- SSVM for Multiclass classification problem:

$$\min_{(v,b)\in R^{k(m+1)}} \frac{1}{2}\sum_{j=1}^{k}(v_j'v_j + b_j^2) +$$

$$\frac{C}{2}\sum_{i=1}^{m}\sum_{j\neq y_i} p((v_j' - v_{y_i}')K(A,x_i) + (b_j - b_{y_i}) + 1, \alpha)^2$$

3-class Classification Problem

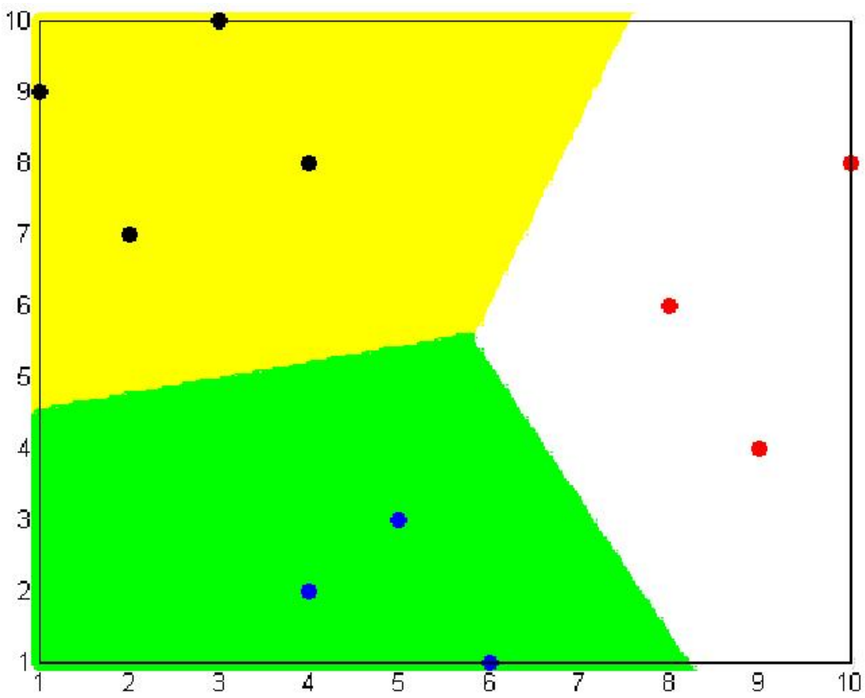
- Given three training datasets A^1, A^2 and A^3 for each distinct category respectively. The linear 3-SSVM formulation is as follows:

$$\min_{\omega \in R^{3(n+1)}} \quad \frac{1}{2}\|\omega\|_2^2 + \frac{C}{2}\|p(B\omega + \mathbf{1}, \alpha)\|_2^2.$$

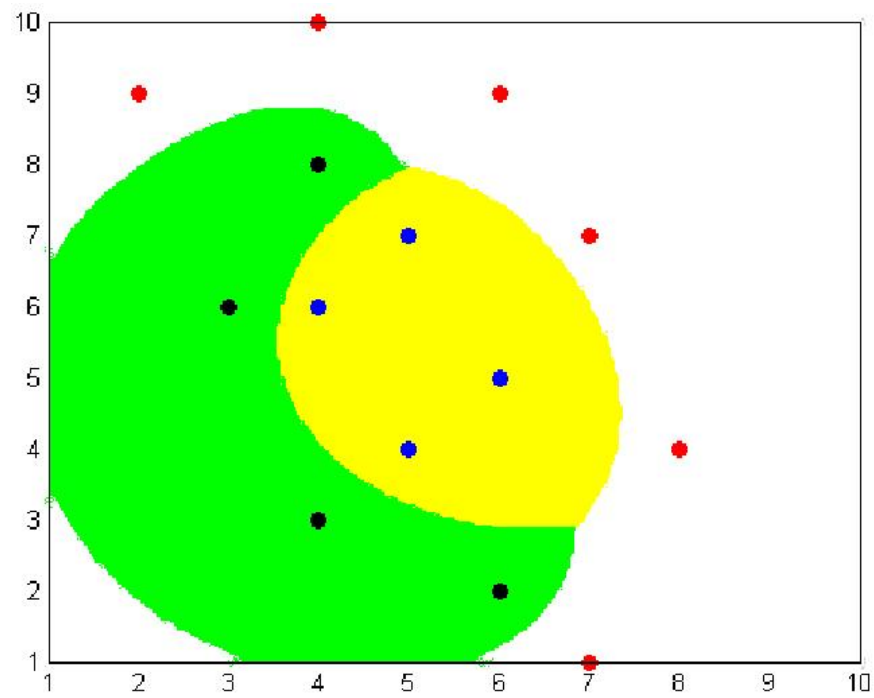
- Here the matrix $B \in R^{2m \times 3(n+1)}$ consists of A^1, A^2, and A^3 $\omega \in R^{3(n+1)}$ is the solution vector.
- We can also apply the 3-SSVM to multiclass classification problem very well. The idea is similar to the one-against-one method. We call it "***Smooth One-One-Rest***" (SOOR) method.

Synthetic Datasets
(For 3-class Classification Problems)

Linear Separable



Nonlinear Separable



Support Vector Regression

(Linear Case: $f(x) = x'w + b$)

- Given the training set:

$$S = \{(x^i, y_i) |\ \ x^i \in R^n,\ \ y_i \in R,\ \ i = 1,\ \ \ldots,\ \ m\}$$

- Find a linear function, $f(x) = x'w + b$ such that $f(x^i) = w'x^i + b \approx y_i, \forall i$

- The (w, b) guarantees the smallest overall experiment error made by $f(x) = x'w + b$

ϵ-Insensitive Loss Function

(Discard the Tiny Error)

- ϵ-insensitive loss function:

$$|\xi|_\epsilon = \max\{0, |\xi| - \epsilon\} = \begin{cases} 0 & \text{if } |\xi| \leqslant \epsilon \\ |\xi| - \epsilon & \text{otherwise} \end{cases}$$

- If $\xi \in R^n$ then $|\xi|_\epsilon \in R^n$ is defined as:

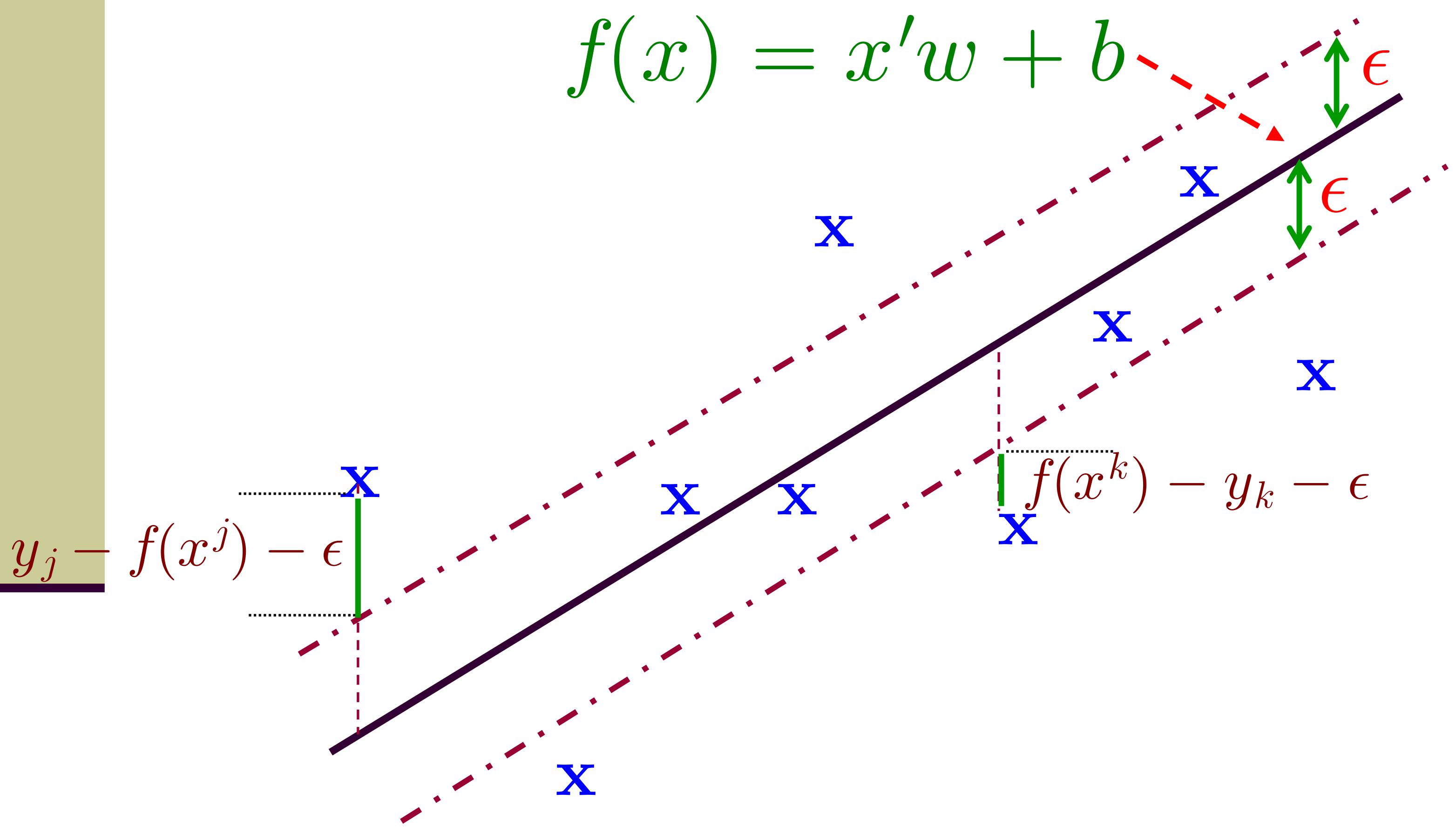
$$(|\xi|_\epsilon)_i = |\xi_i|_\epsilon \ , \ \ i = 1 \ldots n$$

- The loss made by the estimation function, f at the data point (x^i, y_i) is

$$|f(x^i) - y_i|_\epsilon = \max\{0, |f(x^i) - y_i| - \epsilon\}$$

ϵ-Insensitive Linear Regression

Find (w, b) with the smallest overall error



ϵ-insensitive Support Vector Regression Model

- Motivated by SVM:
  - $||w||_2$ should be as small as possible
  - Some tiny error should be discarded

$$\min_{(w,b,\xi)\in R^{n+1+m}} \quad \frac{1}{2}||w||_2^2 + C\mathbf{1}'\,|\xi|_\epsilon$$

where $|\xi|_\epsilon \in R^m, \;\; (|\xi|_\epsilon)_i = \max\{0, |A_i w + b - y_i| - \epsilon\}$

Reformulated ϵ- SVR as a Constrained Minimization Problem

$$\min_{(w,b,\xi,\xi^*)\in R^{n+1+2m}} \frac{1}{2}||w||_2^2 + C\mathbf{1}'(\xi+\xi^*)$$

subject to

$$\begin{aligned} y - Aw - \mathbf{1}b &\leqslant \epsilon\mathbf{1} + \xi \\ Aw + \mathbf{1}b - y &\leqslant \epsilon\mathbf{1} + \xi^* \\ \xi, \xi^* &\geqslant \mathbf{0} \end{aligned}$$

n+1+2m variables and *2m* constrains minimization problem

Enlarge the problem size and computational complexity for solving the problem

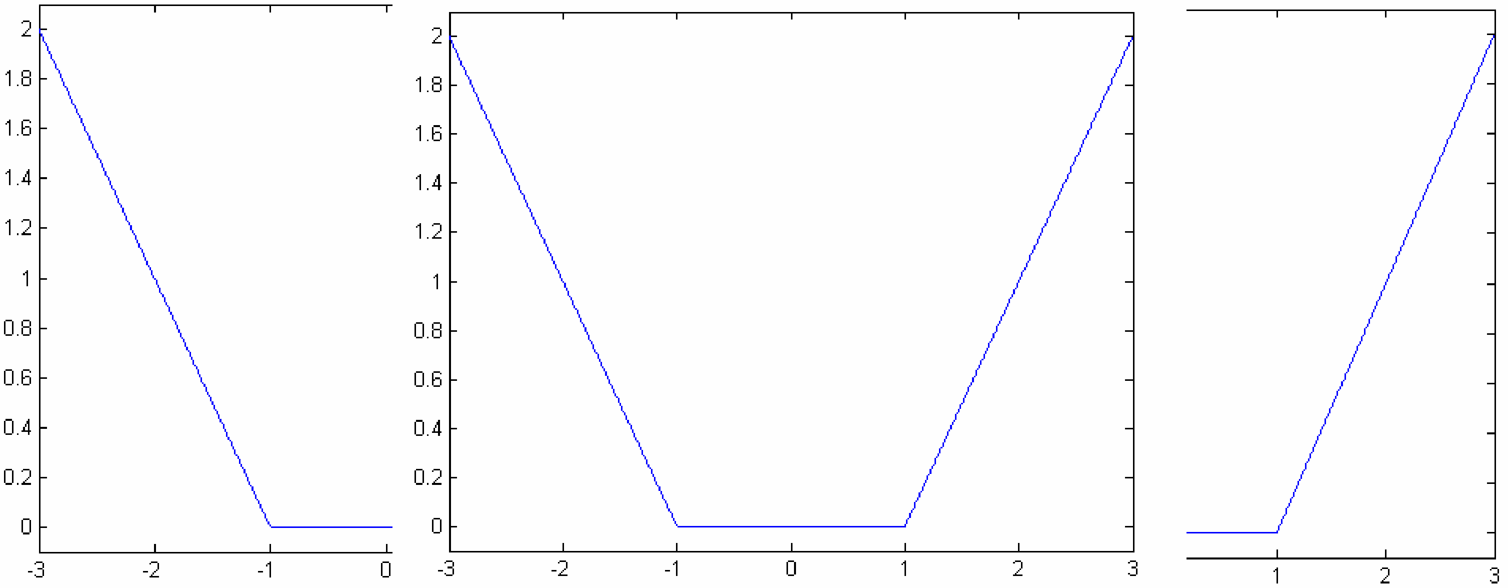
SV Regression by Minimizing Quadratic ϵ-Insensitive Loss

$$\min_{(w,b,\xi)\in R^{n+1+m}} \quad \frac{1}{2}(\|w\|_2^2 + b^2) + \frac{C}{2}\|(|\xi|_\epsilon)\|_2^2$$

where $(|\xi|_\epsilon)_i = |y_i - (w'x^i + b)|_\epsilon$

- We are going to "smooth" $\|(|\xi|)_\epsilon\|_2^2$ and solve the unconstrained problem directly.
- The objective function is strongly convex

ϵ-insensitive Loss Function



$(-x-\epsilon)_+$ $\quad |x|_\epsilon = (x-\epsilon)_+ + (-x-\epsilon)_+$ $\quad (x-\epsilon)_+$

Quadratic ϵ-insensitive Loss Function

$$|x|_\epsilon^2 = ((x - \epsilon)_+ + (-x - \epsilon)_+)^2$$

$$= (x - \epsilon)_+^2 + (-x - \epsilon)_+^2$$

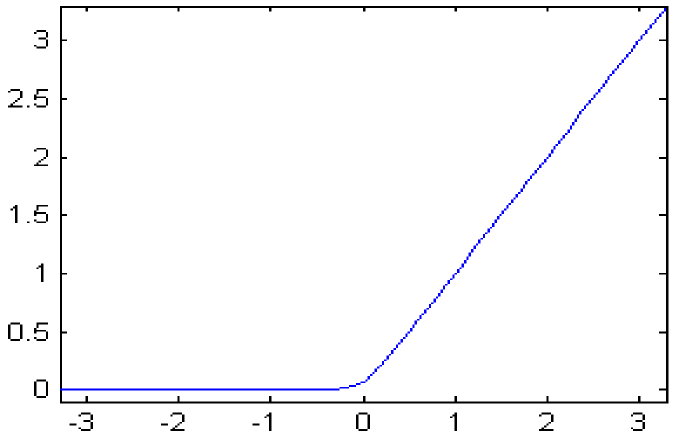
$$(x - \epsilon)_+ \cdot (-x - \epsilon)_+ = 0$$

Use p_ϵ^2-function replace Quadratic ϵ -insensitive Function

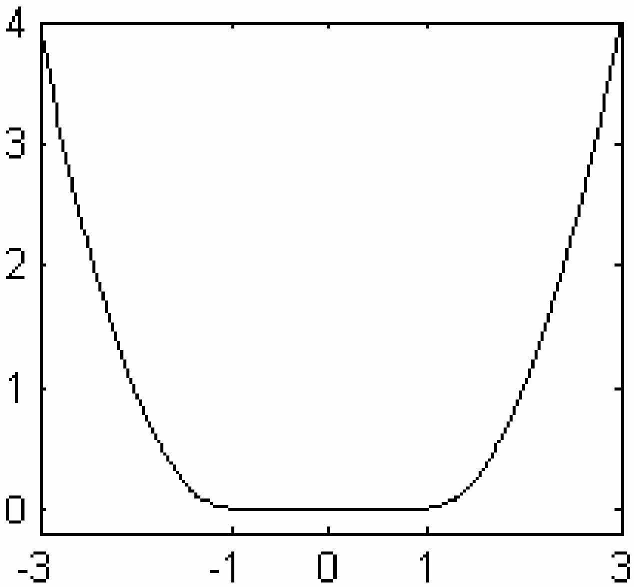
$$p_\epsilon^2(x,\beta) = (p(x-\epsilon,\beta))^2 + (p(-x-\epsilon,\beta))^2$$

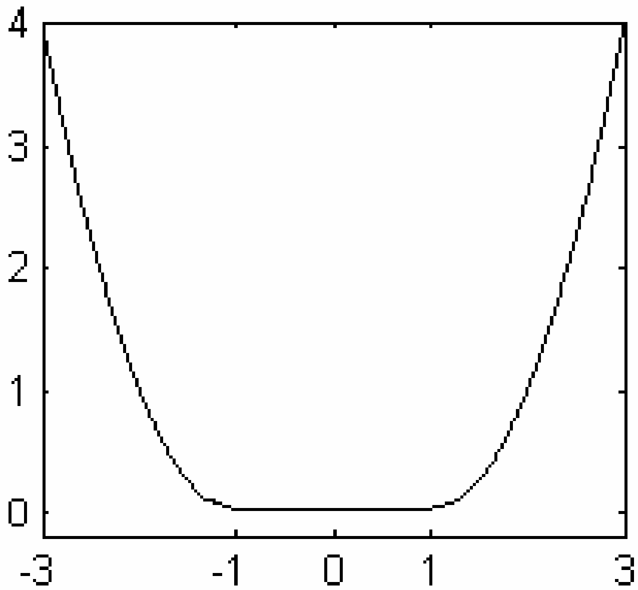
where $p(x,\beta)$ is defined by

$$p(x,\beta) = x + \frac{1}{\beta}\log(1+\exp^{-\beta x})$$



p -function with
$\beta = 10,\ p(x,10),\ x \in [-3,3]$





$|\,x\,|_{\epsilon}^{2}$

$p_{\epsilon}^{2}(x,\beta),\ \ \epsilon = 1,\ \ \beta = 5$

ϵ -insensitive Smooth Support Vector Regression

$$\min_{(w,b)\in R^{n+1}} \quad \Phi_{\epsilon,\alpha}(w,b) :=$$

$$\min_{(w,b)\in R^{n+1}} \quad \frac{1}{2}(w'w+b^2)+\frac{C}{2}\sum_{i=1}^{m}|A_i w+b-y_i|_{\epsilon}^{2}$$

$$\frac{C}{2}\sum_{i=1}^{m}p_{\epsilon}^{2}(A_i w+b-y_i,\beta)$$

This problem is a strongly convex minimization problem without any constrain

The object function is twice differentiable thus we can use a fast Newton-Armijo method to solve this problem

Nonlinear Smooth Support Vector ϵ-insensitive Regression

$$\min_{(\alpha,b)\in R^{m+1}} \quad \frac{1}{2}(\alpha'\alpha + b^2)$$

$$+ \frac{C}{2}\sum_{i=1}^{m} |K(A_i, A')\alpha + b - y_i|_{\epsilon}^{2}$$

$$+ \frac{C}{2}\sum_{i=1}^{m} p_{\epsilon}^{2}(K(A_i, A')\alpha + b - y_i, \beta)$$

- Nonlinear regression function depends on the data points with nonzero coefficients :

$$K(x', A')D\alpha + b = \sum_{\alpha_j \neq 0} \alpha_j K(A_j, x) + b = 0$$

Nonlinear SVM: A Full Model

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x, A_i) + b$$

- Nonlinear SVM uses a full representation for a classifier or regression function:
  - As many parameters α_i as the data points
- Nonlinear SVM function is a linear combination of basis functions, $\mathcal{B} = \{1\} \cup \{k(\cdot\,, x^i)\}_{i=1}^{m}$
  - $\mathcal{B}$ is an overcomplete dictionary of functions when m is large or approaching infinity
- Fitting data to an overcomplete full model may
  - Increase computational difficulties & model complexity
  - Need more CPU time and memory space
  - Be in danger of overfitting

Reduced SVM: A Compressed Model

It's desirable to cut down the model complexity

- Reduced SVM randomly selects a small subset $\bar{S}$ to generate the basis functions $\overline{\mathcal{B}}$:

$$\bar{S} = \{(\bar{x}^i, \bar{y}_i) \big| i = 1, \ldots, \bar{m}\} \subseteq S, \quad \overline{\mathcal{B}} = \{1\} \cup \{k(\cdot\,, \overline{x}^i)\}_{i=1}^{\overline{m}}$$

- RSVM classifier is in the form $f(x) = \sum_{i=1}^{\overline{m}} \overline{u}_i k(x, \overline{x}^i) + b$
- The parameters are determined by fitting entire data

$$\min_{\overline{u}, b, \xi \geqslant 0} \quad C \sum_{j=1}^{m} \xi_j + \frac{1}{2} \left\| \overline{u} \right\|_2^2$$

s.t. $\quad y_j \left(\sum_{i=1}^{\overline{m}} \overline{u}_i k(x^j, \overline{x}^i) + b \right) + \xi_j \geqslant 1, \forall j = 1, \ldots, m$

Nonlinear SVM *vs.* RSVM

$$K(A, A') \in R^{m \times m} \quad \textit{vs.} \quad K(A, \bar{A}') \in R^{m \times \bar{m}}$$

Nonlinear SVM

$$\min_{\alpha, b, \xi \geqslant 0} \quad C \sum_{j=1}^{m} \xi_j + \frac{1}{2} \left\| \alpha \right\|_2^2$$

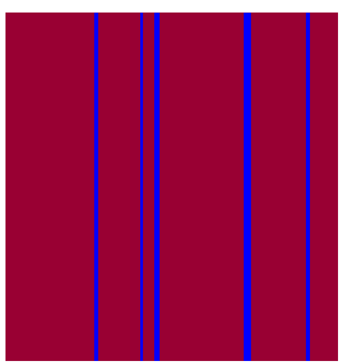
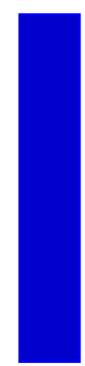
$$D(K(A, A')\alpha + \mathbf{1}b) + \xi \geqslant \mathbf{1}$$

RSVM

$$\min_{\bar{u}, b, \xi \geqslant 0} \quad C \sum_{j=1}^{m} \xi_j + \frac{1}{2} \left\| \bar{u} \right\|_2^2$$

$$D(K(A, \bar{A}')\bar{u} + \mathbf{1}b) + \xi \geqslant \mathbf{1}$$

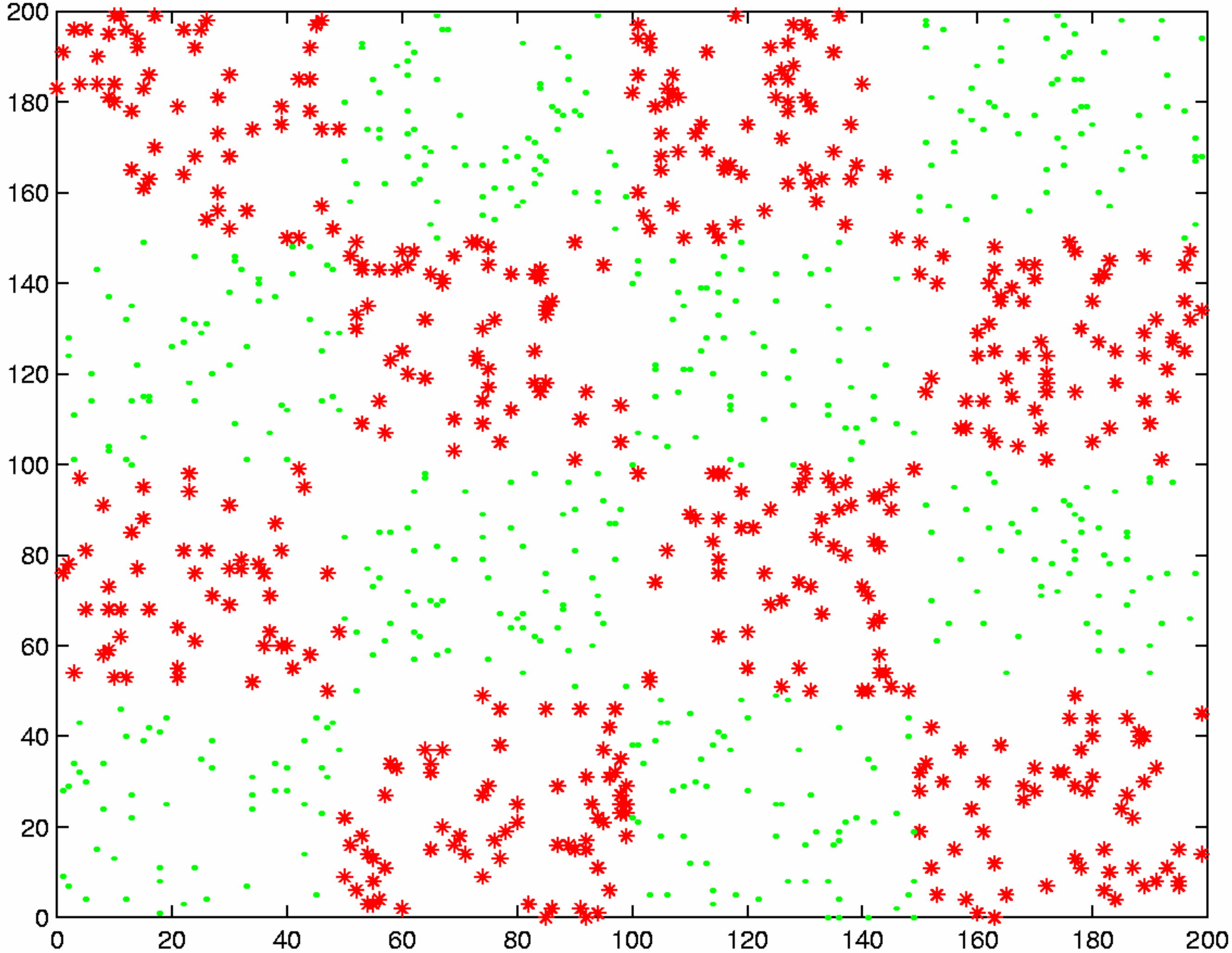
where $K(A, A')_{ij} = k(x^i, x^j)$ and $K(A, \bar{A}')_{ij} = k(x^i, \bar{x}^j)$

$K(A, A') :$  $\longrightarrow$ $K(A, \bar{A}') :$ 

A Nonlinear Kernel Application

Checkerboard Training Set: 1000 Points in R^2

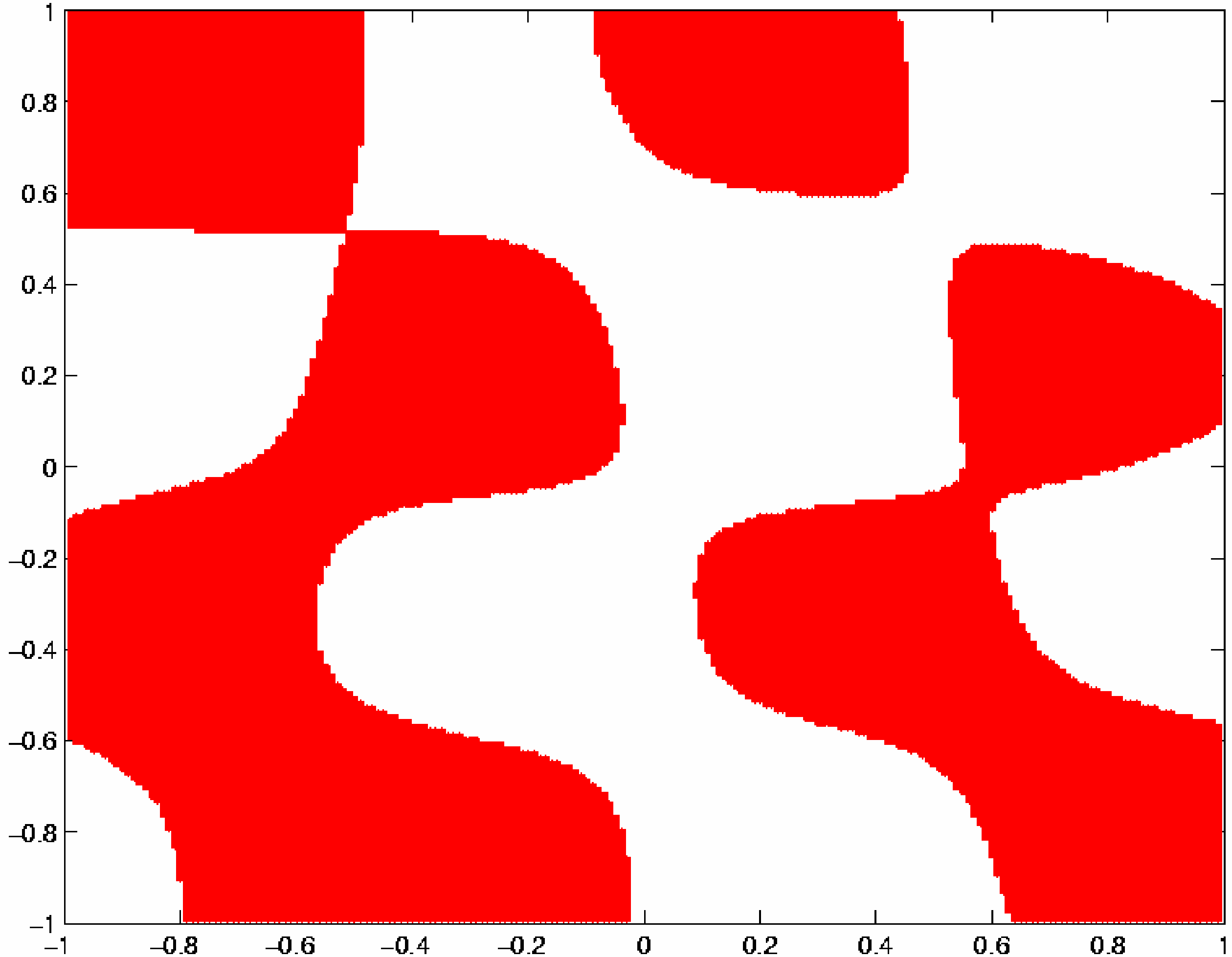
Separate 486 Asterisks from 514 Dots



Conventional SVM Result on Checkerboard

Using 50 Randomly Selected Points Out of 1000

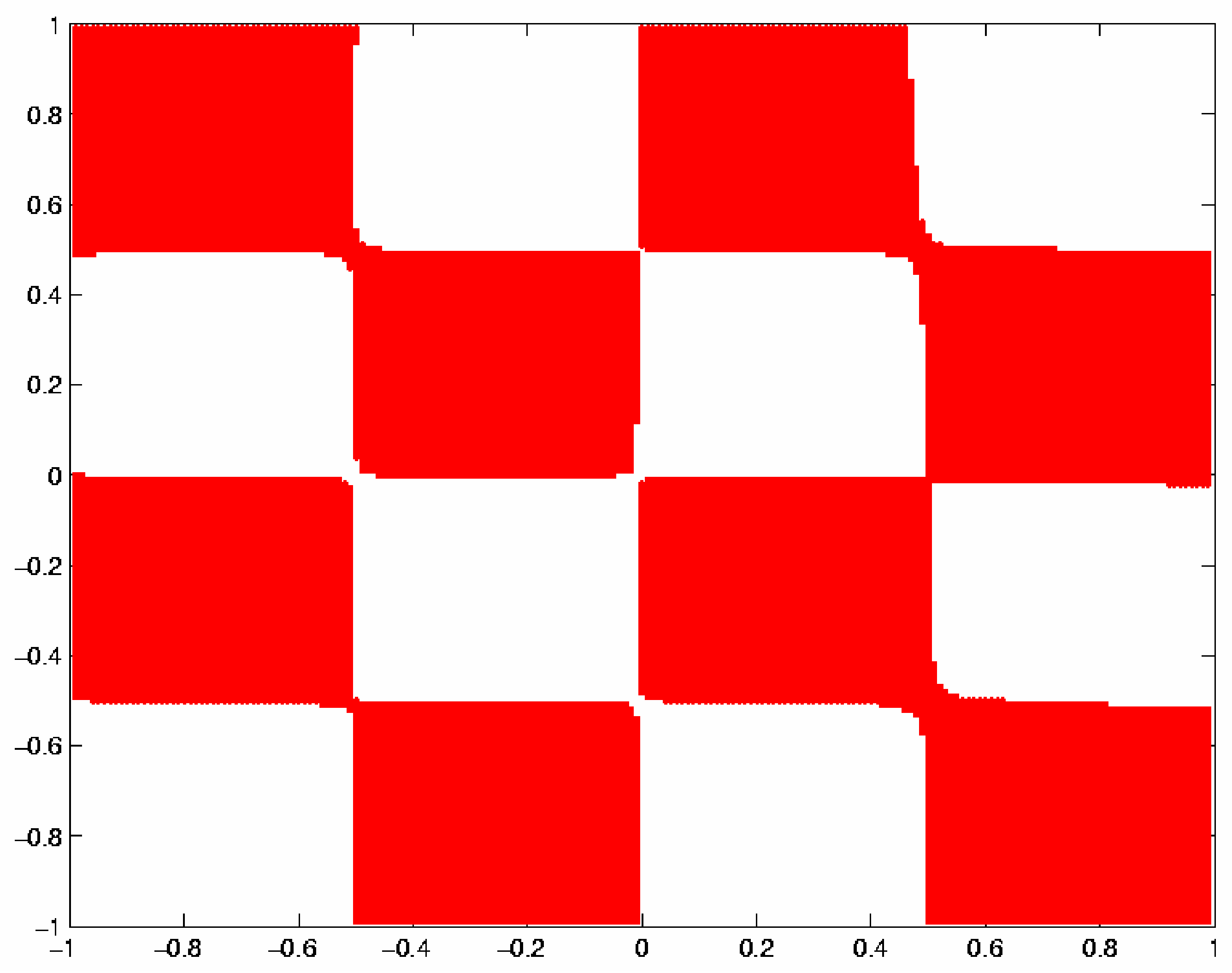
$$K(\overline{A}, \overline{A}') \in R^{50 \times 50}$$



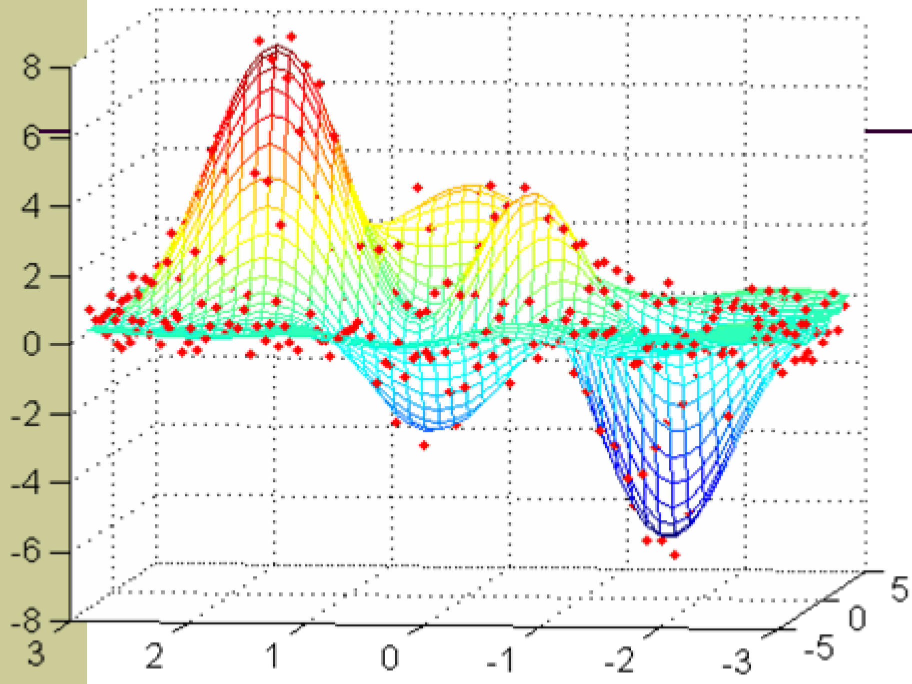
RSVM Result on Checkerboard

Using SAME 50 Random Points Out of 1000

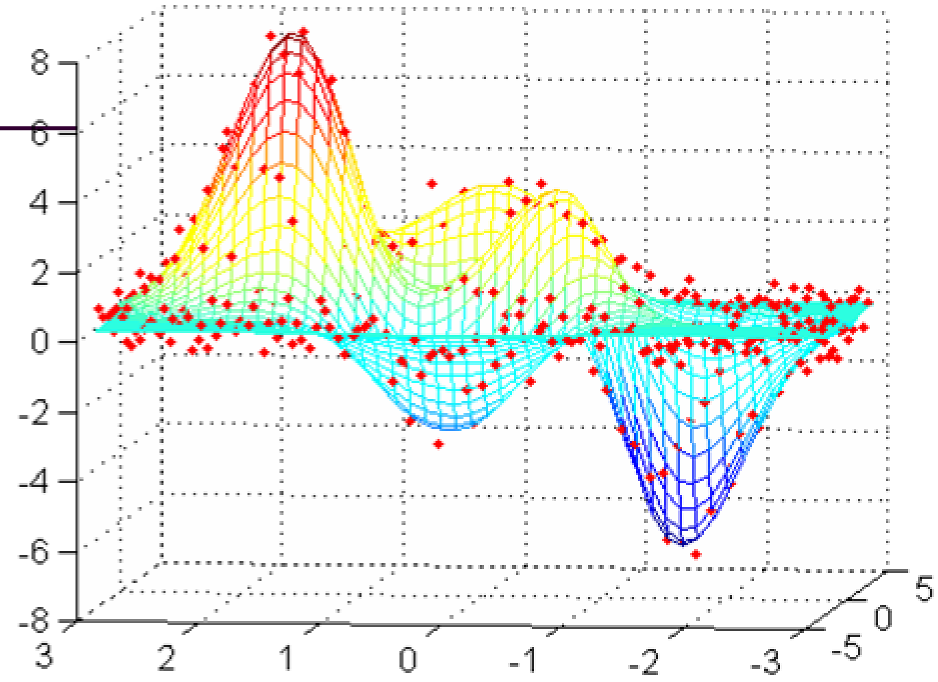
$$K(A, \overline{A}') \in R^{1000 \times 50}$$



481 Data Points in $R^2 \times R$



Estimated Function



Original Function

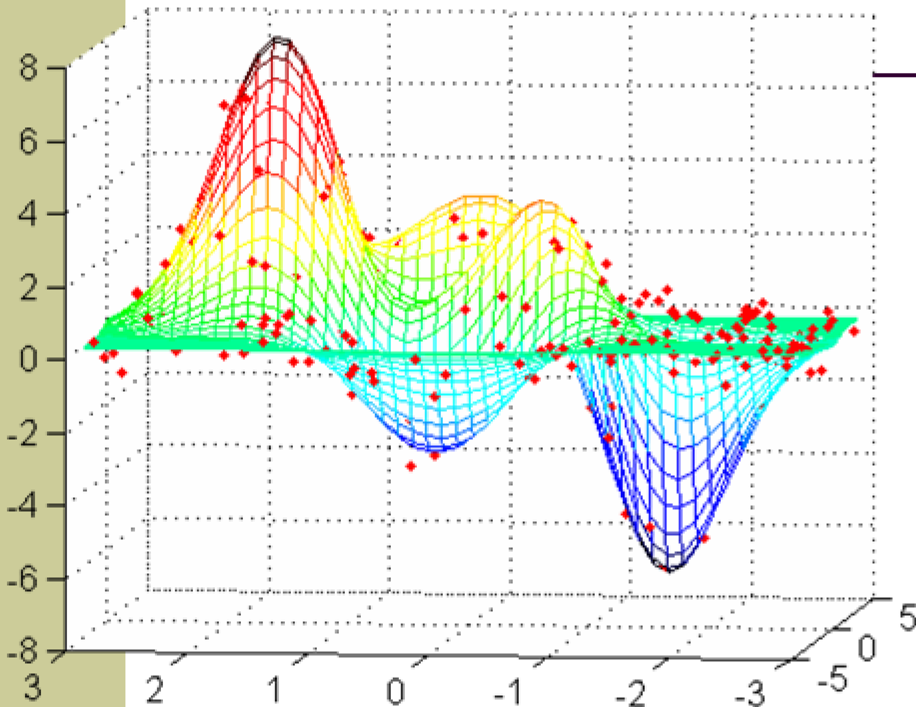
Noise : mean=0 , $\sigma = 0.4$

Parameter : $C = 50, \ \ \gamma = 1, \ \ \varepsilon = 0.5$

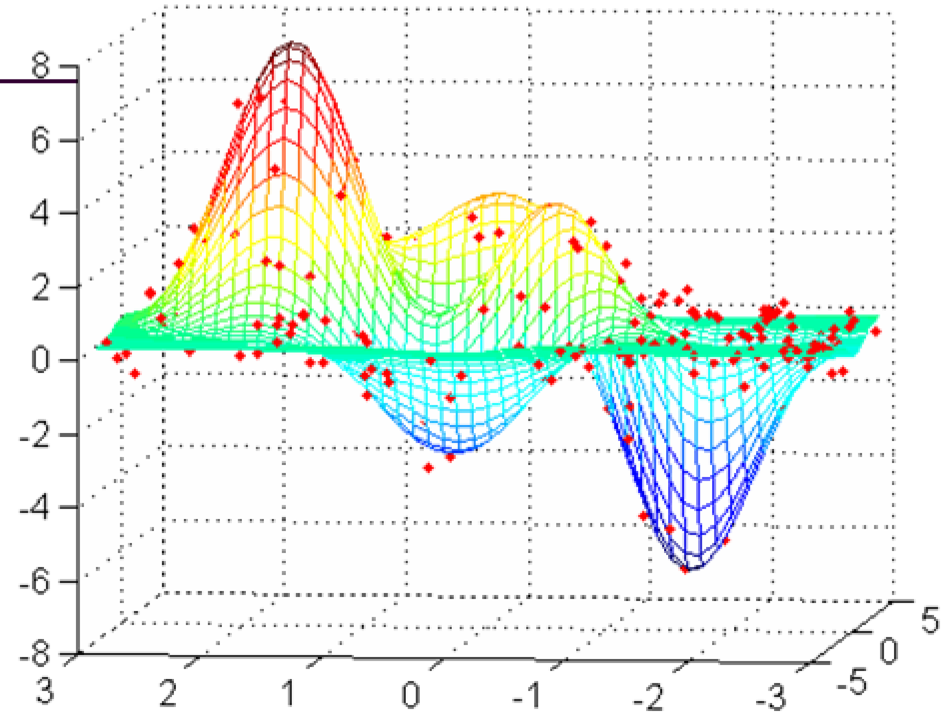
Mean Absolute Error (MAE) of 49x49 mesh points : 0.1761

Training time : 9.61 sec.

Using Reduced Kernel: $K(A, \overline{A}') \in R^{28900 \times 300}$



Estimated Function



Original Function

Noise : mean=0 , $\sigma = 0.4$

Parameter $C = 10000, \ \ \gamma = 1, \ \ \epsilon = 0.2$

MAE of 49x49 mesh points : 0.0513

Training time : 22.58 sec.

Merits of RSVM

Compressed Model *vs*. Full Model

- Computation point of view:
  - Memory usage: Nonlinear SVM $\sim O(m^2)$
    Reduced SVM $\sim O(m \times \overline{m})$
  - Time complexity: Nonlinear SVM $\sim O(m^3)$
    Reduced SVM $\sim O(\overline{m}^3)$
- Model complexity point of view:
  - Compressed model is much *simpler* than full one
  - This may reduced the risk of overfitting
- Successfully applied to other kernel based algorithms
  - SVR, KFDA and Kernel canonical correction analysis

Why RSVM Works so Well?

An Algebraic Explanation

- The full kernel can be approximated by a low-rank approximation which is known as the Nyström approximation. That is,

$$K(A, A') \approx K(A, \overline{A}')K(\overline{A}, \overline{A}')^{-1}K(\overline{A}', \overline{A})$$

- For a vector $u \in R^m$

$$K(A, A')u \approx K(A, \overline{A}')\boxed{K(\overline{A}, \overline{A}')^{-1}K(\overline{A}', \overline{A})u} = \overline{u}$$

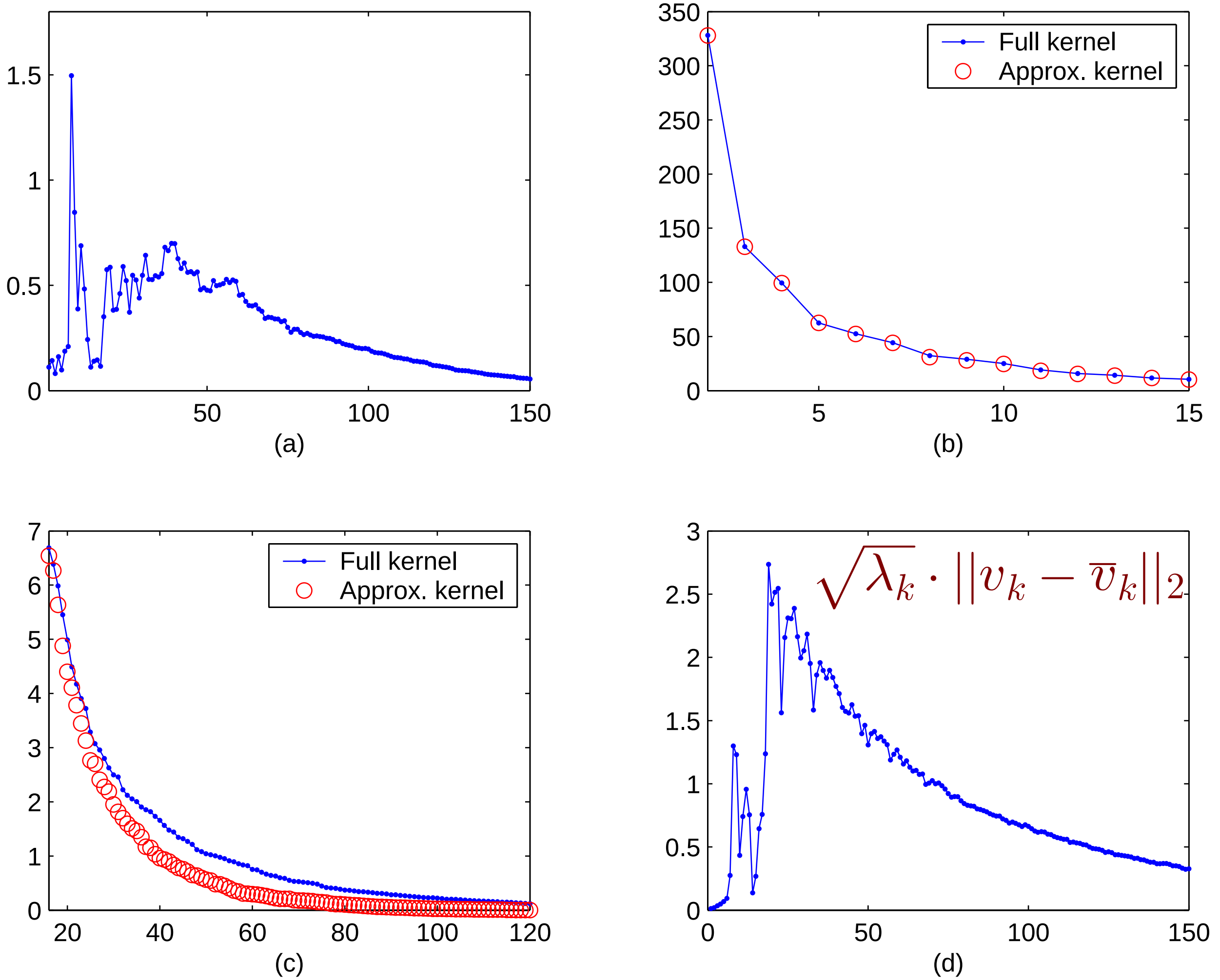
$$= K(A, \overline{A}')\overline{u}$$

- In RSVM, $\overline{u}$ is directly determined by *fitting the entire dataset*

Spectral Analysis

$$K(A,A') \;\; vs. \;\; K(A,\overline{A}')K(\overline{A},\overline{A}')^{-1}K(\overline{A}',\overline{A})$$

Image(2310, 116): Max-diff: 1.496, Rel-diff of Traces: 0.021



Statistical Optimality

Random selection is an optimal robust scheme

- Uniform random selection of reduced set to form the compressed model is an optimal robust scheme in terms of the following criteria:
  - Optimal sampling design for bases selection
    - It minimizes the model variance
  - (MinMax): Minimizes the maximal bias measure between the compressed and full models

Conclusions

- SSVM: A new formulation of support vector machines as a smooth unconstrained minimization problem
  - Can be solved by a fast Newton-Armijo algorithm
  - No optimization (LP, QP) package is needed
- RSVM: A new nonlinear method for massive datasets
  - Overcomes two main difficulties of nonlinear SVMs
    - Reduces the memory storage & computational time
- Rectangular kernel: novel idea for kernel-based Algs.

Thank You!