# Introduction to Support Vector Machine

Yuh-Jye Lee

National Taiwan University of Science and Technology

September 23, 2009

Binary Classification Problem

# 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

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

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

Main Goal:

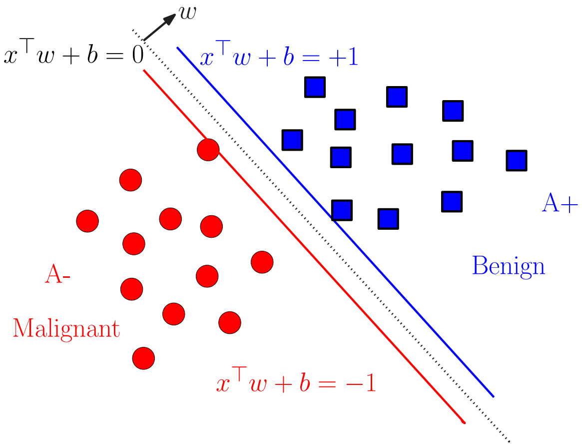
$$\boxed{\text{Predict the unseen class label for new data}}$$

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

$$f(x) \geq 0 \Rightarrow x \in A_+ \text{ and } f(x) < 0 \Rightarrow x \in A_-$$

The simplest function is linear: $f(x) = w^\top x + b$

# Binary Classification Problem
# Linearly Separable Case



# Perceptron Algorithm (Primal Form)
# Rosenblatt, 1956

- An on-line and mistake-driven procedure Repeat:

  *for* $i = 1$ *to* $\ell$

  *if* $y_i(\langle w^k \cdot x^i\rangle + b_k) \leq 0$ *then*

  $w^{k+1} \leftarrow x^k + \eta y_i x^i$

  $b_{k+1} \leftarrow b_k + \eta y_i R^2$

  $$\boxed{R = \max_{1\leq i\leq \ell} \|x^i\|}$$

  $k \leftarrow k + 1$

  *end if*

  until no mistakes made within the for loop return: $k, (w^k, b_k)$.

  What is $k$ ?

## $y_i(\langle w^{k+1} \cdot x^i \rangle + b_{k+1}) > y_i(\langle w^k \cdot x^i \rangle) + b_k$ ?
## $w^{k+1} \longleftarrow w^k + \eta y_i x^i$ and $b_{k+1} \longleftarrow b_k + \eta y_i R^2$

$$
\begin{aligned}
y_i(\langle w^{k+1} \cdot x^i \rangle + b_{k+1}) &= y_i(\langle (w^k + \eta y_i x^i) \cdot x^i \rangle + b_k + \eta y_i R^2) \\
&= y_i(\langle w^k \cdot x^i \rangle + b_k) + y_i(\eta y_i(\langle x^i \cdot x^i \rangle + R^2)) \\
&= y_i(\langle w^k \cdot x^i \rangle + b_k) + \eta(\langle x^i \cdot x^i \rangle + R^2)
\end{aligned}
$$

$$\boxed{R = \max_{1 \leq i \leq \ell} \|x^i\|}$$

# Perceptron Algorithm Stop in Finite Steps

Theorem(Novikoff)
Let $S$ be a non-trivial training set, and let

$$R = \max_{1 \leq i \leq \ell} \|x^i\|$$

Suppose that there exists a vector $w_{opt}$ *such that* $\|w_{opt}\| = 1$ and

$$y_i(\langle w_{opt} \cdot x^i \rangle + b_{opt}) \text{ for } 1 \leq i \leq \ell.$$

Then the number of mistakes made by the on-line perceptron algorithm on $S$ is almost $(\frac{2R}{r})^2$.

## Perceptron Algorithm (Dual Form)

$$w = \sum_{i=1}^{\ell} \alpha_i y_i x^i$$

Given a linearly separable training set $S$ and $\alpha = 0$ , $\alpha \in \mathbb{R}^\ell$ , $b = 0$ , $R = \max_{1 \le i \le \ell} \|x_i\|$.

Repeat: *for* $i = 1$ *to* $\ell$

$$\textit{if} \;\; y_i(\sum_{j=1}^{\ell})\alpha_i y_i \langle x^j \cdot x^i \rangle + b) \le 0 \;\; \textit{then}$$

$$\alpha_i \leftarrow \alpha_i + 1 \; ; \; b \leftarrow b + y_i R^2$$

*end if*

*end for*

Until no mistakes made within the for loop return: $(\alpha, b)$

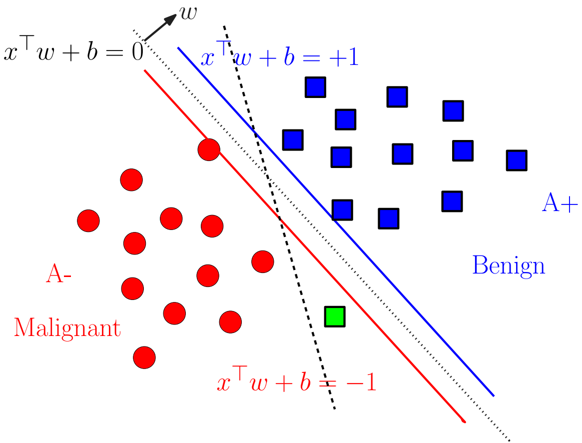
# What We Got in the DualForm PerceptronAlgorithm?

- The number of updates equals: $\sum_{i=1}^{\ell} \alpha_i = \|\alpha\|_1 \leq (\frac{2R}{r})^2$
- $\alpha_i > 0$ implies that the training point $(x_i, y_i)$ has been misclassified in the training process at least once.
- $\alpha_i = 0$ implies that removing the training point $(x_i, y_i)$ will not affect the final results.
- The training data only appear in the algorithm through the entries of the Gram matrix, $G \in \mathbb{R}^{\ell \times \ell}$ which is defined below:

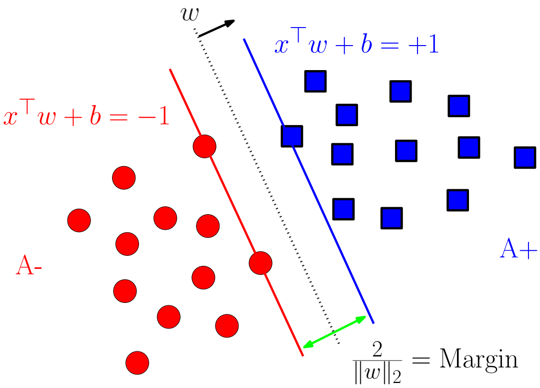
$$G_{ij} = \langle x_i, x_j \rangle$$

Support Vector Machine

# 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

# Why We Maximize the Margin? (Based on Statistical Learning Theory)

- The Structural Risk Minimization (SRM):
  - The expected risk will be less than or equal to empirical risk (training error)+ VC (error) bound
- $\|w\|_2 \propto$ *VC bound*
- $\min$ *VC bound* $\Leftrightarrow \min \frac{1}{2}\|w\|_2^2 \Leftrightarrow \max$ *Margin*

# Summary the Notations

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

$$A = \begin{bmatrix} (x^1)^\top \\ (x^2)^\top \\ \vdots \\ (x^\ell)^\top \end{bmatrix} \in \mathbb{R}^{\ell \times n}, D = \begin{bmatrix} y_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & y_\ell \end{bmatrix} \in \mathbb{R}^{\ell \times \ell}$$

$A_i w + b \geq +1, \ \ for\ D_{ii} = +1$
$A_i w + b \leq -1, \ \ for\ D_{ii} = -1$ , equivalent to $D(Aw + \mathbf{1}b) \geq \mathbf{1}$ , where $\mathbf{1} = [1, 1, \ldots, 1]^\top \in \mathbb{R}^\ell$

# Support Vector Classification (Linearly Separable Case, Primal)

The hyperplane $(w, b)$ is determined by solving the minimization problem:

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

$$D(Aw + \mathbf{1}b) \geq \mathbf{1},$$

It realizes the maximal margin hyperplane with geometric margin

$$\gamma = \frac{1}{\|w\|_2}$$

# Support Vector Classification (Linearly Separable Case, Dual Form)

The dual problem of previous MP:

$$\max_{\alpha \in R^{\ell}} \quad \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top DAA^\top D\alpha$$

subject to

$$\mathbf{1}^\top D\alpha = 0, \alpha \geq \mathbf{0}$$

Applying the KKT optimality conditions, we have $A^\top D\alpha$. But where is $b$ ?
Don't forget

$$\mathbf{0} \leq \alpha \perp D(Aw + \mathbf{1}b) - \mathbf{1} \geq \mathbf{0}$$

# Dual Representation of SVM
# (Key of Kernel Methods: $w = A^\top D\alpha^* = \sum_{i=1}^{\ell} y_i\alpha_i^* A_i^\top$)

The hypothesis is determined by $(\alpha^*, b^*)$

$$
\begin{aligned}
h(x) &= sgn(\langle x \cdot A^\top D\alpha^* \rangle + b^*) \\
&= sgn(\sum_{i=1}^{\ell} y_i\alpha_i^* \langle x^i \cdot x \rangle + b^*) \\
&= sgn(\sum_{\alpha_i^* > 0} y_i\alpha_i^* \langle x^i \cdot x \rangle + b^*)
\end{aligned}
$$

*Remember :* $A_i^\top = x_i$

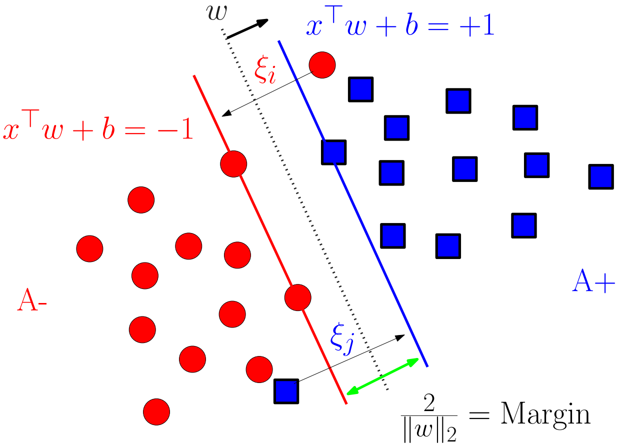
# Soft Margin SVM (Nonseparable Case)

- If data are not linearly separable
  - Primal problem is infeasible
  - Dual problem is unbounded above
- Introduce the slack variable for each training point

$$y_i(w^\top x^i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i$$

- The inequality system is always feasible e.g.

$$w = \mathbf{0}, \quad b = 0, \quad \xi = \mathbf{1}$$


$w$
$x^\top w + b = +1$
$\xi_i$
$x^\top w + b = -1$
A-
A+
$\xi_j$
$\frac{2}{\|w\|_2} = \text{Margin}$

## Robust Linear Programming
## Preliminary Approach to SVM

$$\begin{aligned}\min_{w,b,\xi} \quad & \mathbf{1}^\top \xi \\ \text{s.t.} \quad & D(Aw + \mathbf{1}b) + \xi \geq \mathbf{1} \quad (LP) \\ & \xi \geq \mathbf{0}\end{aligned}$$

where $\xi$ is nonnegative slack(*error*) vector

- The term $\mathbf{1}^\top\xi$, 1-norm measure of *error* vector, is called the *training error*
- For the linearly separable case, at solution of(LP): $\xi = \mathbf{0}$

# Support Vector Machine Formulations (Two Different Measures of Training Error)

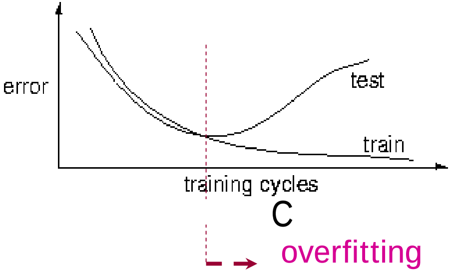
2-Norm Soft Margin:

$$\min_{(w,b,\xi)\in\mathbb{R}^{n+1+\ell}} \quad \frac{1}{2}\|w\|_2^2 + \frac{C}{2}\|\xi\|_2^2$$
$$D(Aw + \mathbf{1}b) + \xi \geq \mathbf{1}$$

1-Norm Soft Margin (Conventional SVM)

$$\min_{(w,b,\xi)\in\mathbb{R}^{n+1+\ell}} \quad \frac{1}{2}\|w\|_2^2 + C\mathbf{1}^\top\xi$$
$$D(Aw + \mathbf{1}b) + \xi \geq \mathbf{1}$$
$$\xi \geq \mathbf{0}$$

# Tuning Procedure
# How to determine C ?




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

# Lagrangian Dual Problem

$$
\begin{aligned}
\max_{\alpha,\beta}\min_{x\in\Omega} \quad & L(x,\alpha,\beta) \\
\text{subject to} \quad & \alpha \geq \mathbf{0}
\end{aligned}
$$

$$\Updownarrow$$

$$
\begin{aligned}
\max_{\alpha,\beta} \quad & \theta(\alpha,\beta) \\
\text{subject to} \quad & \alpha \geq \mathbf{0}
\end{aligned}
$$

where$\theta(\alpha,\beta) = \inf_{x\in\Omega} L(x,\alpha,\beta)$

# 1-Norm Soft Margin SVM
# Dual Formalation

The Lagrangian for 1-norm soft margin:

$$\mathcal{L}(w, b, \xi, \alpha, \gamma) = \frac{1}{2}w^\top w + C\mathbf{1}^\top \xi + \alpha^\top[\mathbf{1} - D(Aw + \mathbf{1}b) - \xi] - \gamma^\top \xi$$

where $\alpha \geq \mathbf{0}$ & $\gamma \geq \mathbf{0}$.
The partial derivatives with respect to primal variables equal zeros:

$$\frac{\partial \mathcal{L}(w, b, \xi, \alpha)}{\partial w} = w - A^\top D\alpha = \mathbf{0},$$

$$\frac{\partial \mathcal{L}(w, b, \xi, \alpha)}{\partial b} = \mathbf{1}^\top D\alpha = 0, \qquad \frac{\partial \mathcal{L}(w, b, \xi, \alpha)}{\partial \xi} = C\mathbf{1} - \alpha - \gamma = \mathbf{0}.$$

## Substitute: $w = A^\top D\alpha$, $C\mathbf{1}^\top\xi = (\alpha+\gamma)^\top\xi$ $\mathbf{1}^\top D\alpha = 0$, in $L(w, b, \xi, \alpha, \gamma)$

$$\begin{aligned}\mathcal{L}(w,b,\xi,\alpha,\gamma) &= \frac{1}{2}w^\top w + C\mathbf{1}^\top\xi + \\ &\quad \alpha^\top[\mathbf{1} - D(Aw + \mathbf{1}b) - \xi] - \gamma^\top\xi\end{aligned}$$

where $\alpha \geq \mathbf{0}$ & $\gamma \geq \mathbf{0}$

$$\begin{aligned}\theta(\alpha,\gamma) &= \frac{1}{2}\alpha^\top DAA^\top D\alpha + \mathbf{1}^\top\alpha - \alpha^\top DA(A^\top D\alpha) \\ &= -\frac{1}{2}\alpha^\top DAA^\top D\alpha + \mathbf{1}^\top\alpha\end{aligned}$$

s.t. $\mathbf{1}^\top D\alpha = 0$, $\alpha - \gamma = C\mathbf{1}$ and $\alpha \geq \mathbf{0}$ & $\gamma \geq \mathbf{0}$

# Dual Maximization Problem for 1-Norm Soft Margin

Dual:

$$\max_{\alpha \in \mathbb{R}^{\ell}} \quad \mathbf{1}^{\top}\alpha - \frac{1}{2}\alpha^{\top}DAA^{\top}D\alpha$$
$$\mathbf{1}^{\top}D\alpha = 0$$
$$\mathbf{0} \leq \alpha \leq C\mathbf{1}$$

- The corresponding KKT complementarity

$$\mathbf{0} \leq \alpha \perp D(Aw + \mathbf{1}b) + \xi - \mathbf{1} \geq \mathbf{0}$$
$$\mathbf{0} \leq \xi \perp \alpha - C\mathbf{1} \leq \mathbf{0}$$

# Slack Variables for 1-Norm Soft Margin SVM $f(x) = \sum_{\alpha_i^*>0} y_i\alpha_i^*\langle x^i, x\rangle + b^*$

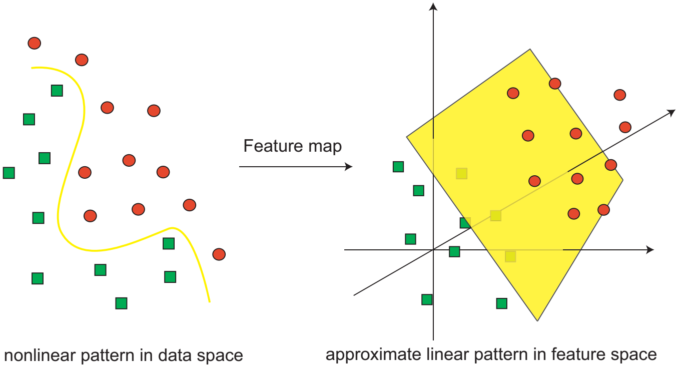
- Non-zero slack can only occur when $\alpha_i^* = C$
  - The contribution of outlier in the decision rule will be at most $C$
  - The trade-off between accuracy and regularization directly controls by $C$
- The points for which $0 < \alpha_i^* < C$ lie at the bounding planes
  - This will help us to find $b^*$

# Two-spiral Dataset (94 white Dots & 94 Red Dots)

# Learning in Feature Space (Could Simplify the Classification Task)

- Learning in a high dimensional space could degrade generalization performance
  - This phenomenon is called *curse of dimensionality*
- By using a *kernel function*, that represents the inner product of training example in feature space, we never need to explicitly know the nonlinear map
  - Even do not know the dimensionality of feature space
- There is no free lunch
  - Deal with a huge and dense kernel matrix
    - Reduced kernel can avoid this difficulty

$$X \overset{\Phi}{- - - \longrightarrow} F$$



# Linear Machine in Feature Space

Let $\phi : X \longrightarrow F$ be a nonlinear map from the input space to some feature space

The classifier will be in the form(*primal*):

$$f(x) = (\sum_{j=1}^{?} w_j \phi_j(x)) + b$$

Make it in the *dual* form:

$$f(x) = (\sum_{i=1}^{\ell} \alpha_i y_i \langle \phi(x^i) \cdot \phi(x) \rangle) + b$$

# Kernel:Represent Inner Product in Feature Space

Definition: A kernel is a function $K : X \times X \longrightarrow \mathbb{R}$ such that *for all* $x, z \in X$

$$K(x,z) \quad = \quad \langle \phi(x) \cdot \phi(z) \rangle$$

where $\phi : X \longrightarrow F$
The classifier will become:

$$f(x) \quad = \quad (\sum_{i=1}^{\ell} \alpha_i y_i K(x^i, x)) + b$$

# A Simple Example of Kernel
## Polynomial Kernel of Degree 2: K(x,z)=$\langle x, z \rangle^2$

Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^2$ and the nonlinear map

$\phi : \mathbb{R}^2 \longmapsto \mathbb{R}^3$ defined by $\phi(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{bmatrix}$.

Then $\langle \phi(x), \phi(z) \rangle = \langle x, z \rangle^2 = K(x, z)$

- There are many other nonlinear maps, $\psi(x)$, that satisfy the relation: $\langle \psi(x), \psi(z) \rangle = \langle x, z \rangle^2 = K(x, z)$

## Power of the Kernel Technique

Consider a nonlinear map $\phi : \mathbb{R}^n \longmapsto \mathbb{R}^p$ that consists of distinct features of all the *monomials* of degree $d$.

Then $p = \binom{n+d-1}{d}$.

$x_1^3 x_2^1 x_3^4 x_4^4 \Longrightarrow$ x o o o x o x o o o o x o o o o

For example: n=11, d=10, p=92378

- Is it necessary? We only need to know $\langle \phi(x), \phi(z) \rangle$!
- This can be achieved $K(x,z) = \langle x, z \rangle^d$

# 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 function 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

# Example of Kernel $K(A, B) : \mathbb{R}^{\ell \times n} \times \mathbb{R}^{n \times \tilde{\ell}} \longmapsto R^{\ell \times \tilde{\ell}}$

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

- Polynomial Kernel:
  - $(AA^\top + \mu aa^\top)^d_\bullet$ (Linear Kernel$AA^\top$ : $\mu = 0, d = 1$)
- Gaussian (Radial Basis) Kernel:
  - $K(A, A^\top)_{ij} = e^{-\mu \|A_i - A_j\|_2^2}$, $i, j = 1, ..., m$

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

# Nonlinear Support Vector Machine (Applying the Kernel Trick)

1-Norm Soft Margin Linear SVM:

$$\max_{\alpha \in \mathbb{R}^\ell} \ \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top D A A^\top D \alpha \ \ s.t. \ \ \mathbf{1}^\top D\alpha = 0, \ \ \mathbf{0} \le \alpha \le C\mathbf{1}$$

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

1-Norm Soft Margin Nonlinear SVM:

$$\begin{aligned} &\max_{\alpha \in \mathbb{R}^\ell} \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top D K(A, A^\top) D\alpha \\ &s.t. \ \ \mathbf{1}^\top D\alpha = 0, \ \ \mathbf{0} \le \alpha \le C\mathbf{1} \end{aligned}$$

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

# 1-Norm SVM
# (Different Measure of Margin)

1-Norm SVN:

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

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

$$\xi \geq \mathbf{0}$$

Equivalent to:

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

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

$$-s \leq w \leq s$$

$$\xi \geq \mathbf{0}$$

Good for feature selection and similar to the LASSO

Smooth Support Vector Machine

# Support Vector Machine Formulations

## Two Different Measures of Training Error

2-Norm Soft Margin (Primal form):

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

$$D\left(Aw + \mathbf{1}b\right) + \xi \geq \mathbf{1}$$

1-Norm Soft Margin (Primal form):

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

$$D\left(Aw + \mathbf{1}b\right) + \xi \geq \mathbf{1}, \quad \xi \geq \mathbf{0}$$

- Margin is maximized by minimizing reciprocal of margin.

# SVM as an Unconstrained Minimization Problem

$$\begin{array}{ll} \min\limits_{w,b} & \frac{C}{2}\|\xi\|_2^2 + \frac{1}{2}\left(\|w\|_2^2 + b^2\right) \\ \textbf{s.t.} & D\left(Aw + 1b\right) + \xi \geq \mathbf{1} \end{array} \quad (QP)$$

At the solution of (QP) : $\xi = (\mathbf{1} - D(Aw + \mathbf{1}b))_+$ where $(\cdot)_+ = max\left\{\cdot, 0\right\}$.

Hence (QP) is equivalent to the nonsmooth SVM:

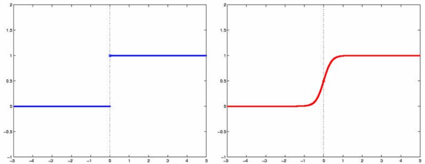
$$\min_{w,b} \frac{C}{2}\,\|(\mathbf{1} - D(Aw + \mathbf{1}b))_+\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2)$$

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

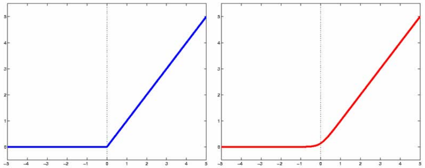
# Smooth the Plus Function: Integrate $\left(\frac{1}{1+\epsilon^{-\beta x}}\right)$

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

**The Step Function $(x)_*$ and the Sigmoid-Function $\frac{1}{1+\varepsilon^{-\alpha x}}$**



**The Plus Function $(x)_+$ and the 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:

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

- The solution of SSVM converges to the solution of nonsmooth SVM as $\beta$ goes to infinity.

## 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 $(w^0, b_0) \in \mathbb{R}^{n+1}$ . Having $(w^i, b_i)$, stop if $\nabla\Phi_\beta(w^i, b_i) = 0$, else :

1. Newton Direction :

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

2. Armijo Stepsize :

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

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

such that Armijos rule is satisfied

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

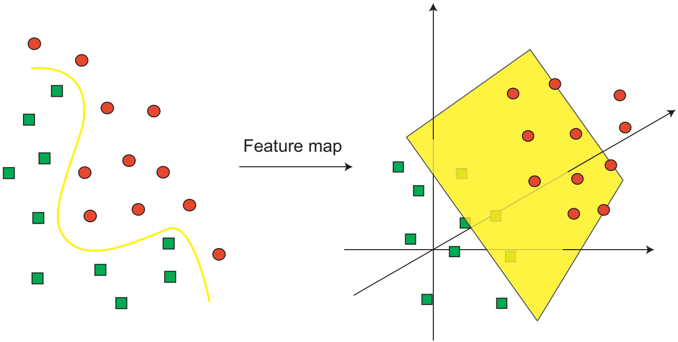
# Newton-Armijo Method: Quadratic Approximation of SSVM

- 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

Nonlinear Smooth Support Vector Machine

# The Illustration of Nonlinear SVM

$$X \overset{\Phi}{- - - \longrightarrow} F$$




nonlinear pattern in data space

approximate linear pattern in feature space

## Nonlinear SSVM Motivation

- Linear SVM: (Linear separator: $x^\top w + b = 0$ )

$$\begin{array}{cc} \min\limits_{\xi \geq 0, w, b} & \frac{C}{2}\|\xi\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2) \\ \textbf{s.t.} & D(Aw + \mathbf{1}b) + \xi \geqslant \mathbf{1} \end{array} \quad (QP)$$

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

$$\begin{array}{cc} \min\limits_{\xi \geq 0, \alpha, b} & \frac{C}{2}\|\xi\|_2^2 + \frac{1}{2}(\|\alpha\|_2^2 + b^2) \\ \textbf{s.t.} & D(AA^\top D\alpha + \mathbf{1}b) + \xi \geqslant \mathbf{1} \end{array}$$

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

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

# Kernel Trick

- We can use the value of kernel function to represent the inner product of two training points in feature space as follows:

$$K(\mathbf{x}, \mathbf{z}) = < \phi(\mathbf{x}), \phi(\mathbf{z}) > .$$

- The most popular kernel function is the Gaussian kernel

$$K(\mathbf{x}, \mathbf{z}) = e^{-\gamma ||\mathbf{x}-\mathbf{z}||_2^2}.$$

- The kernel matrix $K(A, A^\top)_{n\times n}$ represents the inner product of all points in the feature space where $K(A, A^\top)_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.
- Replace $AA^\top$ by a nonlinear kernel $K(A, A^\top)$ without defining a explicit feature map $\phi$

# Nonlinear Smooth SVM
## Nonlinear Classifier: $K(x^\top, A^\top)D\alpha + b = 0$

- Replace $AA^\top$ by a nonlinear kernel $K(A, A^\top)$ :

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

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

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

Reduced Support Vector Machine

# Nonlinear SVM: A Full Model $f(x) = \sum_{i=1}^{\ell} \alpha_i k(x, A_i) + b$

- Nonlinear SVM uses a full representation for a classifier or regression function:
  - As many parameters $\alpha_i$ as the data points
- Nonlinear SVM function is a linear combination of basis functions,$\beta = \{1\} \cup \left\{k(\cdot, x^i)\right\}_{i=1}^{\ell}$
  - $\beta$ is an overcomplete dictionary of functions when 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 $\overline{\mathcal{B}}$:
  $\bar{S} = \{(\bar{x}^i, \bar{y}_i) \big| i = 1, \ldots, \bar{\ell}\} \subseteq S, \overline{\mathcal{B}} = \{1\} \cup \left\{k(\cdot, \bar{x}^i)\right\}_{i=1}^{\bar{\ell}}$
- RSVM classifier is in the form $f(x) = \sum_{i=1}^{\bar{\ell}} \bar{u}_i k(x, \bar{x}^i) + b$
- The parameters are determined by fitting entire data

$$\begin{array}{ll} \min\limits_{\bar{u}, b, \xi \geqslant 0} & C \sum_{j=1}^{\ell} \xi_j + \frac{1}{2}(\|\bar{u}\|_2^2 + b^2) \\ \text{s.t.} & D(K(A, \bar{A}^\top)\bar{u} + \mathbf{1}b) + \xi \geqslant \mathbf{1} \end{array}$$

# Nonlinear SVM vs. RSVM
# $K(A, A^\top) \in \mathbb{R}^{\ell \times \ell}$ vs. $K(A, \bar{A}^\top) \in \mathbb{R}^{\ell \times \bar{\ell}}$

*Nonlinear SVM*

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

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

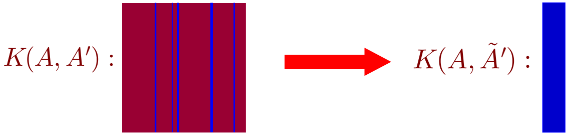
where $K(A, A^\top)_{ij} = k(x^i, x^j)$

*RSVM*

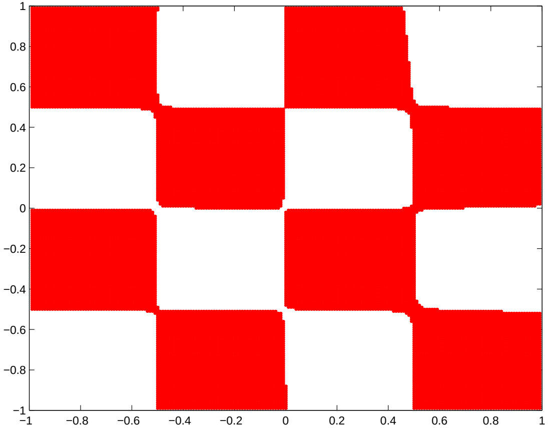
$$\min_{\bar{u}, b, \xi \geqslant 0} \quad C \sum_{j=1}^{\ell} \xi_j + \frac{1}{2}(\|\bar{u}\|_2^2 + b^2)$$

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

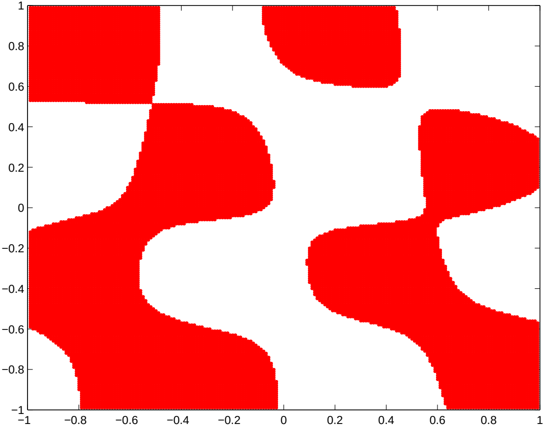
where $K(A, \bar{A}^\top)_{ij} = k(x^i, \bar{x}^j)$



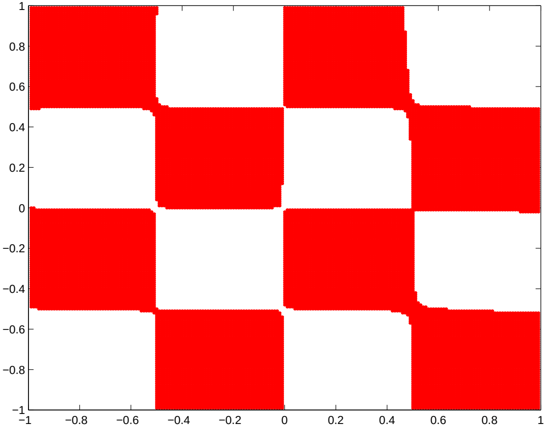
# A Nonlinear Kernel Application
## Checkerboard Training Set: 1000 Points in Separate 486 Asterisks from 514 Dots



# Conventional SVM Result on Checkerboard Using 50 Randomly Selected Points Out of 1000 $K(\overline{A}, \overline{A}^\top) \in \mathbb{R}^{50\times 50}$



# RSVM Result on Checkerboard Using SAME 50 Random Points Out of 1000 $K(A, \overline{A}^{\top}) \in \mathbb{R}^{1000 \times 50}$



## Merits of RSVM
## Compressed Model vs. Full Model

- Computation point of view:
  - Memory usage: Nonlinear SVM $\sim O(\ell^2)$
    Reduced SVM $\sim O(\ell \times \bar{\ell})$
  - Time complexity: Nonlinear SVM $\sim O(\ell^3)$
    Reduced SVM $\sim O(\bar{\ell}^3)$
- 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

Automatic Model Selection via Uniform Design

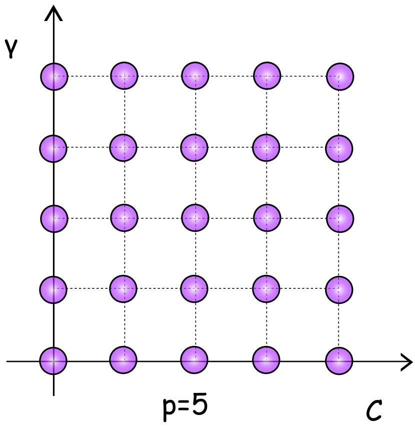
# Model Selection for SVMs

- Choosing a good parameter setting for a better generalization performance of SVMs is the so called model selection problem
- It will be desirable to have an effective and automatic model selection scheme to make SVMs practical for real applications
  - In particular for the people who are not familiar with parameters tuning procedure in SVMs
- Focus on selecting the combinations of regularization parameter C and width parameter $\gamma$ in the Gaussian kernel

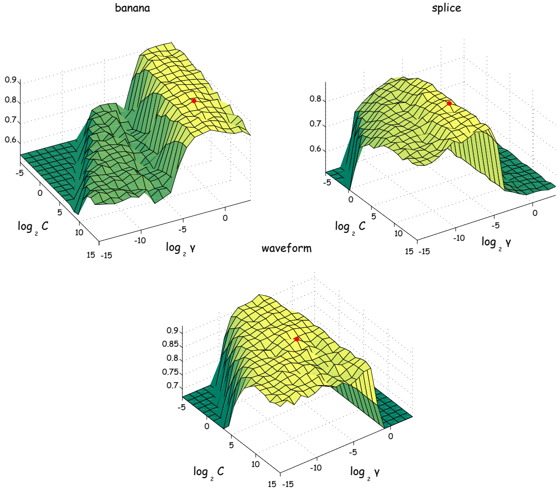
## Searching the Optimal Combination of Two Parameters

- Model selection can be treated as an optimization problem:
  - The objective function is only vaguely specified
  - It has many local maxima and minima
  - Evaluating the objective function value is very expensive task which includes:
    - Training a SVM with a particular parameter setting
    - Testing the SVM resulting model on a validation set

## Grid Search



# Validation Set Accuracy Surface



# Where Are our Tuning Parameters

- Gaussian kernel: $K(A, A^\top)_{ij} = e^{-\gamma\|A_i - A_j\|_2^2}$
- Conventional nonlinear SVM:

$$\begin{aligned} \max_{\alpha\in\mathbb{R}^\ell} \quad & \mathbf{1}^\top\alpha - \tfrac{1}{2}\alpha^\top DK(A, A^\top)D\alpha \\ & e^\top D\alpha = 0 \\ & \mathbf{0} \leq \alpha \leq C\mathbf{1} \end{aligned}$$

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

# Heuristic for Determining Parameters Search Range

- The parameter in Gaussian kernel is more sensitive than parameter C in objective function
- The range of $\gamma$ is determined by the closest pair of data points in the training set such that

$$0.15 \leq e^{-r\|u-v\|_2^2} \leq 0.999$$

- For massive dataset, you may try other heuristics e.g., sampling or the shortest distance to centriod
- We want to scale the distance factor in the Gaussian kernel automatically

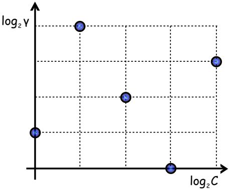
# Heuristic for Determining Parameters Search Range(cont.)

- Reduced kernel always has a larger C than full kernel since the reduced model has been simplified
  - Full kernel:C_Range=[1e-2, 1e+4]
  - Reduced kernel:C_Range=[1e0, 1e+6]

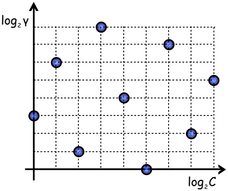
# Uniform Experimental Design

- The uniform design (UD) is one kind of space filling designs that seeks its design points to be uniformly scattered on the experimental domain
- UD can be used for industrial experiments when the underlying model is unknown or only vaguely specified
  - Our SVM model selection problem is in this case
- Once the search domain and number of levels for each parameter are determined the candidate set of parameter combinations can be found by a UD table
  Available at: http://www.math.hkbu.edu.hk/UniformDesign

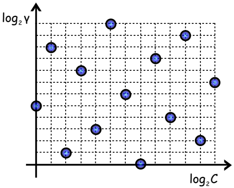
# UD Sampling Patterns




The 5 runs UD sampling pattern



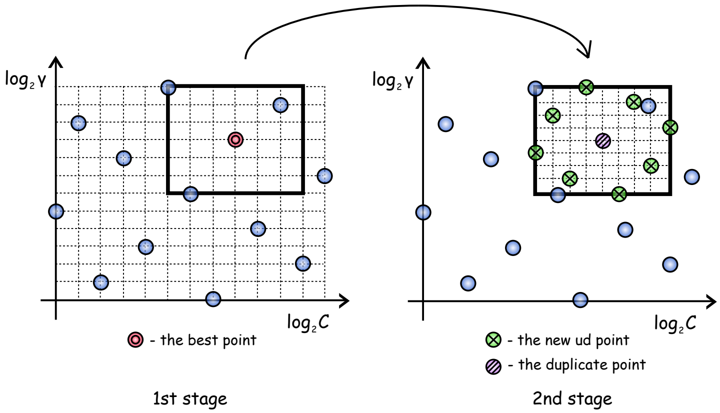

The 9 runs UD sampling pattern



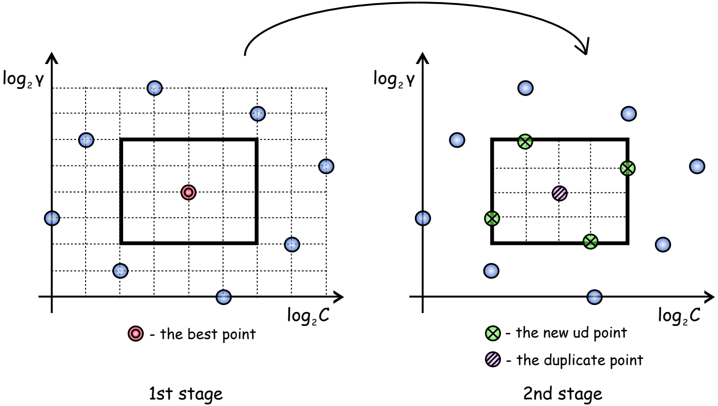

The 13 runs UD sampling pattern

UD: Uniform Design

# Nested UD-based Method(1/2)



# Nested UD-based Method(2/2)



# Experimental Results(1/2)

| Problem | SSVM | | |
|---|---|---|---|
| | DOE | UD1 | UD2 |
| banana | 0.1207±0.0071 | 0.1219±0.0070 | 0.1185±0.0070 |
| image | 0.0289±0.0058 | 0.0307±0.0040 | 0.0279±0.0061 |
| splice | 0.1015±0.0030 | 0.1005±0.0019 | 0.1003±0.0030 |
| waveform | 0.1048±0.0046 | 0.1055±0.0035 | 0.1087±0.0053 |
| tree | 0.1183±0.0023 | 0.1171±0.0026 | 0.1189±0.0029 |
| adult | 0.1604±0.0011 | 0.1605±0.0020 | 0.1611±0.0021 |
| web | 0.0232±0.0007 | 0.0236±0.0014 | 0.0229±0.0020 |

# Experimental Results(2/2)

| Problem | RSVM | | |
|---|---|---|---|
| | DOE | UD1 | UD2 |
| banana | 0.1203±0.0038 | 0.1229±0.0077 | 0.1239±0.0053 |
| image | 0.0461±0.0082 | 0.0437±0.0082 | 0.0429±0.0081 |
| splice | 0.1342±0.0069 | 0.1346±0.0041 | 0.1360±0.0053 |
| waveform | 0.1117±0.0044 | 0.1138±0.0040 | 0.1121±0.0039 |
| tree | 0.1186±0.0033 | 0.1193±0.0054 | 0.1178±0.0040 |
| adult | 0.1621±0.0017 | 0.1614±0.0019 | 0.1625±0.0016 |
| web | 0.0266±0.0039 | 0.0248±0.0014 | 0.0258±0.0020 |

# 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.
- Applied uniform design to SVMs model selection that can be done automatically

# Reference


- Yuh-Jye Lee and O. L. Mangasarian. "SSVM: A Smooth Support Vector Machine for Classification", Computational Optimization and Applications, 20, (2001) 5-22.
- Yuh-Jye Lee and Su-Yun Huang. "Reduced Support Vector Machines: A Statistical Theory", IEEE Transactions on Neural Networks, Vol. 18, No. 1(2007), 1-13.
- Chien-Ming Huang, Yuh-Jye Lee, Dennis K. J. Lin and Su-Yun Huang. "Model Selection for Support Vector Machines via Uniform Design", A special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis. Vol. 52, (2007), 335-346.