

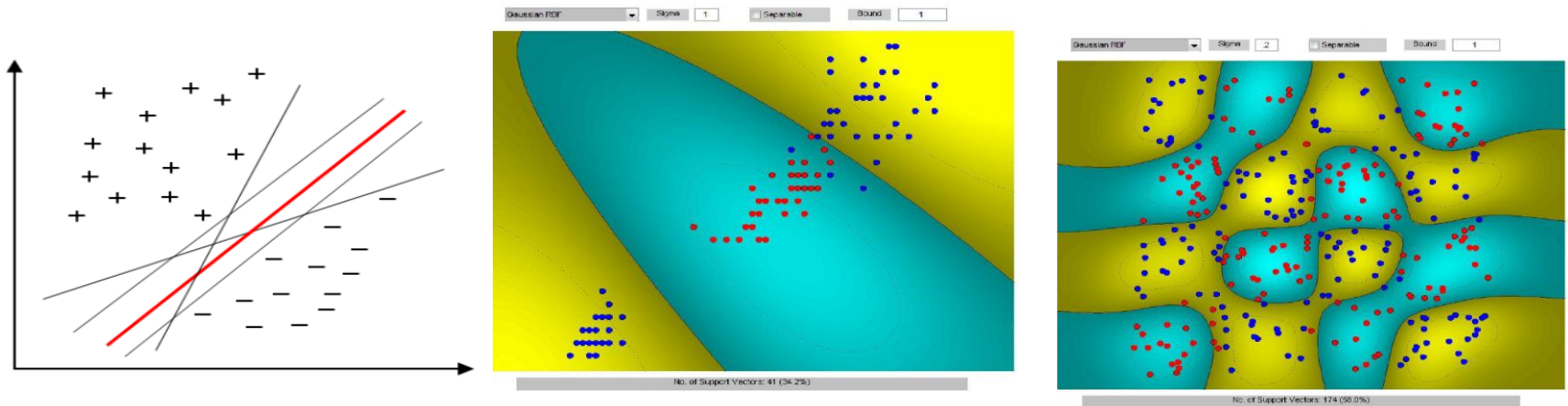
ISE 589/OR 591

LECTURE 5 – CLASSIFICATION

1. Bi-classification
2. Basic Linear Support Vector Machine Models
3. Kernel-based Support Vector Machines
4. Multi-classification
5. Support Vector Regression Models

Classification

- Classification is a process designed to assign objects to some pre-existing categories.
- It is a supervised machine learning algorithm that sets the boundaries of pre-known classes and criteria to assign an object to the class it belongs to.



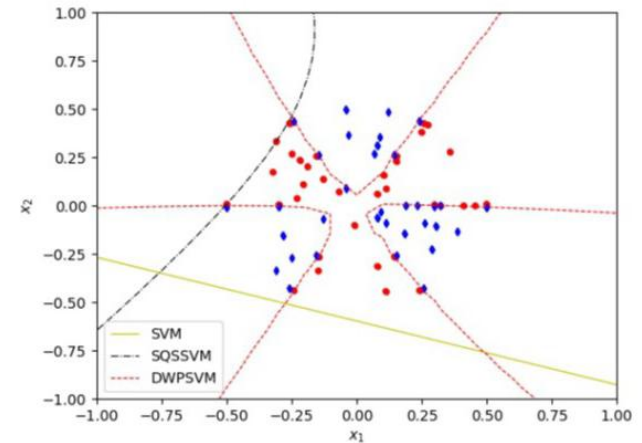
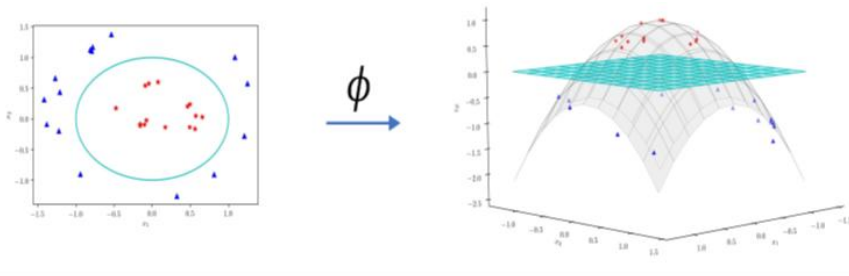
Classification

- How useful is classification?
 - Situation identification
 - Control and management
- What's challenging?
 - Identify the boundary of a pre-existing class
 - Set decision rules for assignment
 - Conduct bi-classification vs. multi-classification

Classification

An effective approach: Support vector machines (SVM)

- Linear SVM
- Kernel-based SVM
- Nonlinear SVM



Support vector machines (SVM)

- Support vector machines are mainly for pattern recognition in supervised machine learning.
 - SVM is commonly used for classification (recognition, diagnosis, preference, prediction, etc.)
 - SVR means support vector regression
 - SVC means support vector clustering (unsupervised learning)

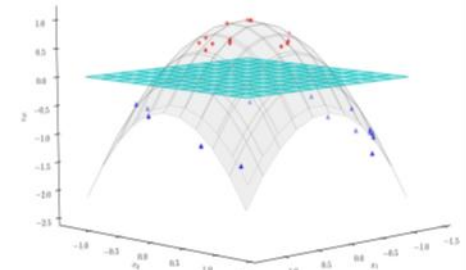
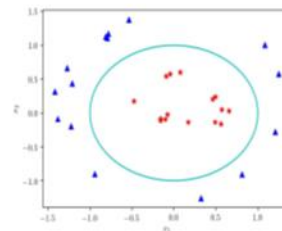
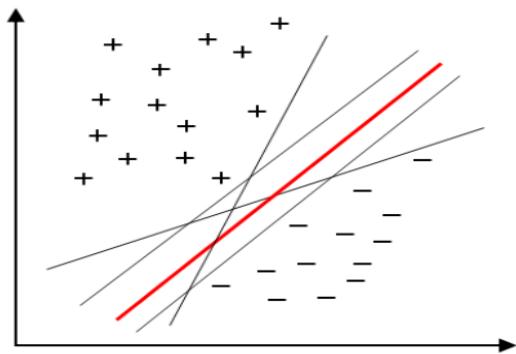
Bi-classification

- Problem facing:

We have a set of N data points $\{x^1, x^2, \dots, x^N\}$, $x^i \in \mathbb{R}^n$, in two different classes labeled by $y_i \in \{-1, 1\}$, $i = 1, \dots, N$.

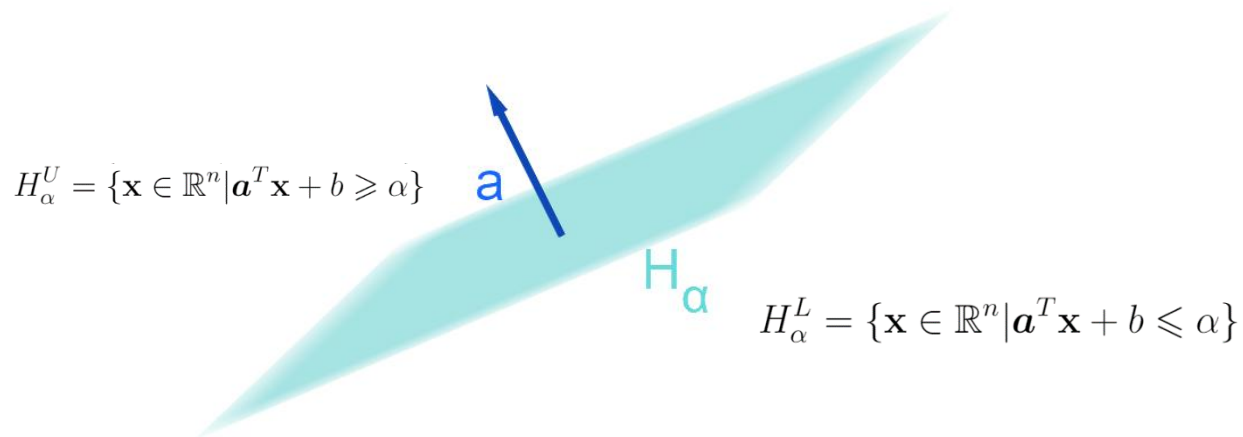
Given a new data point $\bar{x} \in \mathbb{R}^n$, should we label it with $\bar{y} = 1$ or $\bar{y} = -1$?

- Decision making: How? and Why?



Contours of affine (linear) function

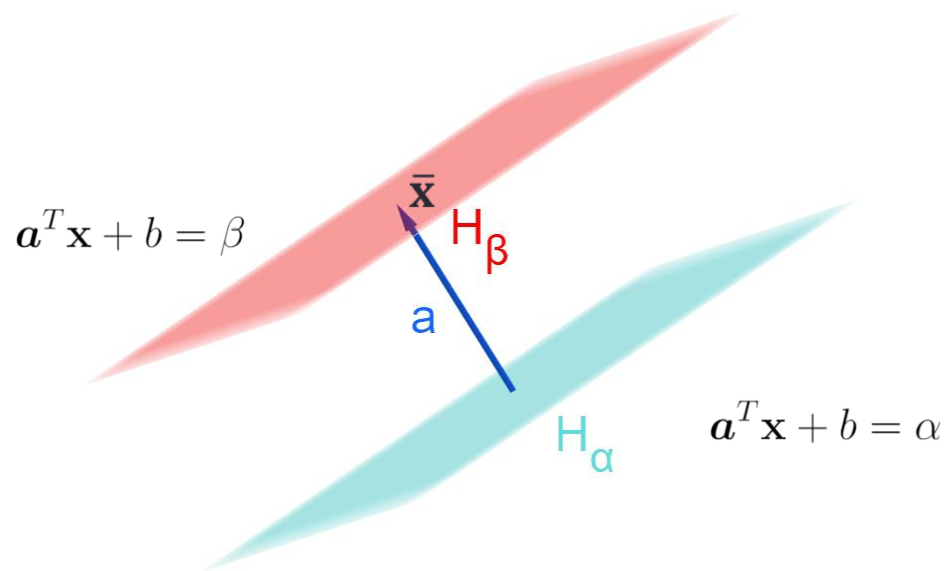
- Define $H_\alpha = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^T \mathbf{x} + b = \alpha\}$



- A **hyperplane** in \mathbb{R}^n with \mathbf{a} being its normal vector.
- Moving along \mathbf{a} will increase $f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + b$, $x \rightarrow H_\alpha^U$

Contours of affine function

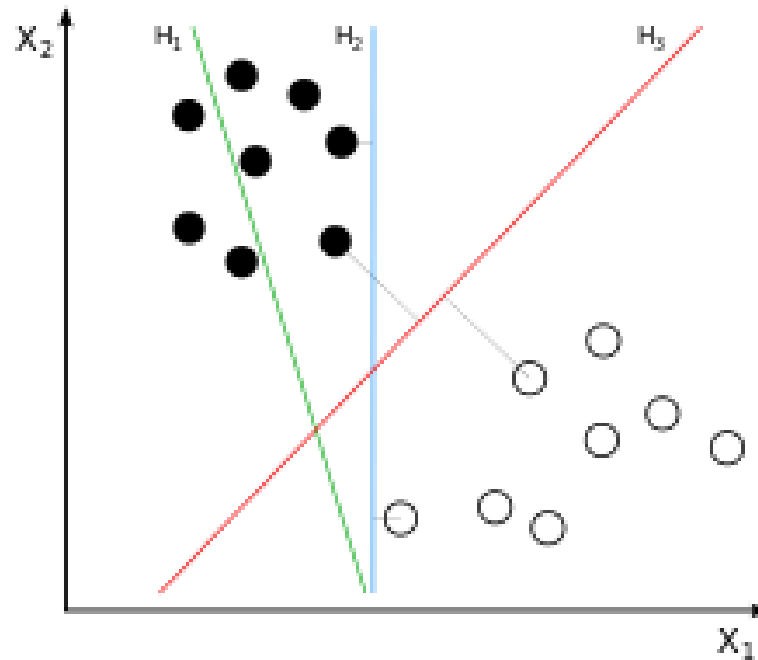
- Given $\bar{\mathbf{x}} \in \mathbb{R}^n$ and H_α , distance $(\bar{\mathbf{x}}, H_\alpha) = ?$



- Distance between $\bar{\mathbf{x}}$ and H_α is $d(\bar{\mathbf{x}}, H_\alpha) = \frac{|\alpha - \beta|}{\|\mathbf{a}\|_2}$

Support vector machines – basic ideas

- Linearly separable



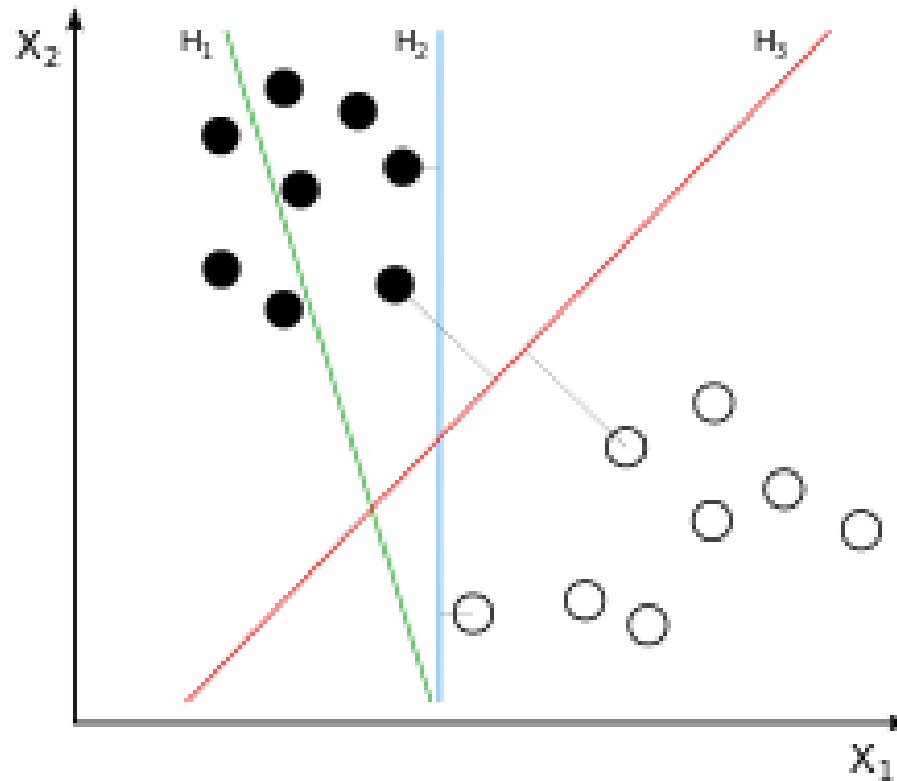
- Given a set of points $\{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ with binary labels $y_i \in \{-1, 1\}$
- Find a hyperplane that strictly separates the two classes.

$$\begin{aligned} \mathbf{a}^T \mathbf{x}^i + b &> 0 & \text{if } y_i = 1 \\ \mathbf{a}^T \mathbf{x}^i + b &< 0 & \text{if } y_i = -1 \end{aligned}$$

$$y_i(\mathbf{a}^T \mathbf{x}^i + b) \geq 0, \quad i = 1, \dots, N.$$

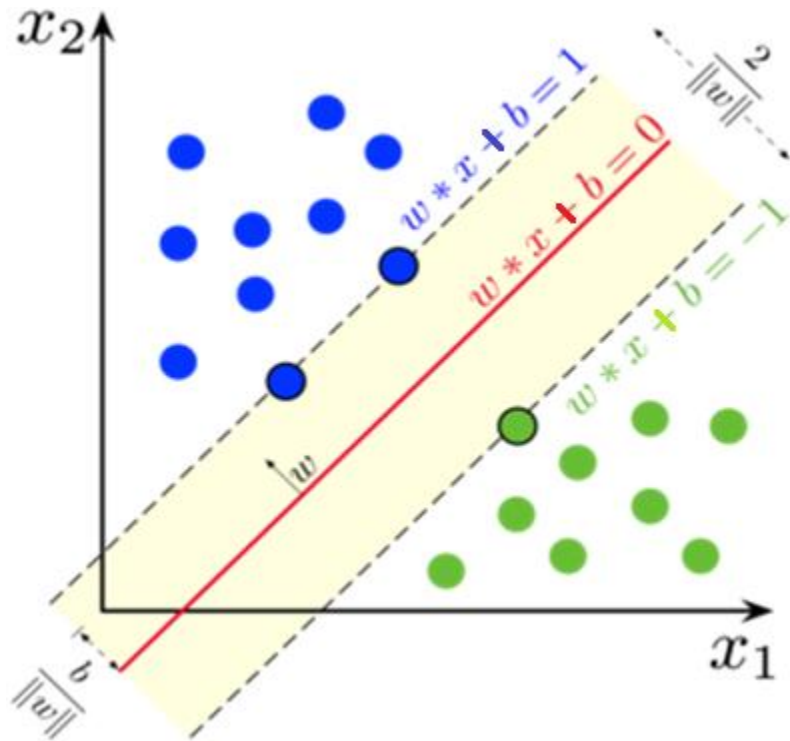
Support vector machines – basic ideas

- Which one to choose? (generalizability)



Linear support vector machine (LSVM) – basic model

- Linear separation with **maximum margin** (distance)



$$\begin{aligned} \max \quad & \frac{2}{\|\mathbf{w}\|_2} \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \mathbf{x}^i + b) \geq 1 \\ & \forall i = 1, \dots, N. \\ & \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}. \end{aligned}$$

equivalently,

$$\begin{aligned} \min \quad & \frac{\|\mathbf{w}\|_2}{2} \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \mathbf{x}^i + b) \geq 1 \\ & \forall i = 1, \dots, N. \\ & \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}. \end{aligned}$$

Linear SVM (hard margin) – LSVM model

- Primal LSVM

$$\min \quad \frac{1}{2} \|\mathbf{w}\|_2^2$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T \mathbf{x}^i + b) \geq 1, \quad i = 1, 2, \dots, N \quad (\text{LSVM})$$

$$\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}$$

- It is a **linearly constrained convex quadratic program** with $n + 1$ variables and N inequality constraints.
- Implications?

LSVM Classifier

- LSVM provides $(\bar{\mathbf{w}}, \bar{b})$ to form a classifier for bi-classification:

- Given an input data point $\mathbf{x} \in \mathbb{R}^n$

$$\mathit{class}_{LSVM}(\mathbf{x}) = \mathit{sign}(\bar{\mathbf{w}}^T \mathbf{x} + \bar{b})$$

where

$$\mathit{sign}(y) = \begin{cases} +1, & \text{if } y > 0 \\ -1, & \text{if } y < 0 \end{cases}$$

Linear SVM (hard margin) – LSVM model

- What else can be say about LSVM?
 - Dual LSVM
 - Optimality conditions
 - Solution methods

Lagrangian dual approach

- Primal LSVM

$$\min \frac{1}{2} \|\mathbf{w}\|_2^2$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}^i + b) \geq 1, \quad i = 1, 2, \dots, N \quad (\text{LSVM})$$

$$\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}$$

- *Lagrangian multiplier method:*

- associating the i^{th} constraint, assign a multiplier $\alpha_i \geq 0$ to construct the Lagrangian function

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|_2^2 + \sum_{i=1}^N \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}^i + b))$$

- * α_i indicates the **influence** of the data point (\mathbf{x}^i, y_i)

Lagrangian dual approach

- Stationary point of the Lagrangian function

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|_2^2 + \sum_{i=1}^N \alpha_i (1 - y_i (\mathbf{w}^T \mathbf{x}^i + b))$$

Lagrangian dual function

$$h(\boldsymbol{\alpha}) \triangleq \min_{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}} L(\mathbf{w}, b, \boldsymbol{\alpha})$$

- Optimality conditions:

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \implies \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}^i$$

$$\nabla_b L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \implies \sum_{i=1}^N \alpha_i y_i = 0$$

\implies dual objective function

$$h(\boldsymbol{\alpha}) = -\frac{1}{2} \left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}^i \right)^T \sum_{i=1}^N \alpha_i y_i \mathbf{x}^i + \sum_{i=1}^N \alpha_i$$

Lagrangian dual approach

KKT conditions for LSVM:

- Stationarity

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}^i \quad \text{and} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

- Primal feasibility

$$y_i (\mathbf{w}^T \mathbf{x}^i + b) \geq 1, \quad i = 1, 2, \dots, N$$

- Dual feasibility

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, N$$

- Complementary slackness

$$\alpha_i (1 - y_i (\mathbf{w}^T \mathbf{x}^i + b)) = 0$$

Dual linear SVM (DLSVM)

- Lagrangian dual model

$$\begin{aligned} \max \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i y_i (\mathbf{x}^i)^T \mathbf{x}^j y_j \alpha_j + \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \quad (\text{DLSVM}) \\ & \alpha_i \geq 0, i = 1, \dots, N \end{aligned}$$

- The Hessian of the dual objective function

$$h(\boldsymbol{\alpha}) = -\frac{1}{2} \boldsymbol{\alpha}^T H \boldsymbol{\alpha} + \sum_{i=1}^N \alpha_i \text{ is}$$

$$H = \text{Diag}(\mathbf{y}) X^T X \text{Diag}(\mathbf{y}) \succeq 0$$

- DLSVM is a convex quadratic program with N nonnegative variables and 1 linear equality constraint.

LSVM or DLSVM ?

- Which one to solve? Why?
 - LSVM or DLSM?
 - how about $n \gg N$ and $N \gg n$?
- How are they related?
 - *primal* – *dual* relation

Relations of LSVM and DLSVM

- Key relations:

1. Convex QP pair means there is no duality gap!

2. Complementary slackness says that

$$\alpha_i (y_i (\mathbf{w}^T \mathbf{x}^i + b) - 1) = 0, \forall i = 1, 2, \dots, N$$

- (a) $\alpha_i = 0$ holds for data point \mathbf{x}^i not on separation hyperplane

(inactive constraint means \mathbf{x}^i plays no role)

- (b) $\alpha_i > 0$ means the point \mathbf{x}^i lies on separation hyperplane

(active constraint means \mathbf{x}^i is a supporting vector)

3. Dual to primal conversion says that

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

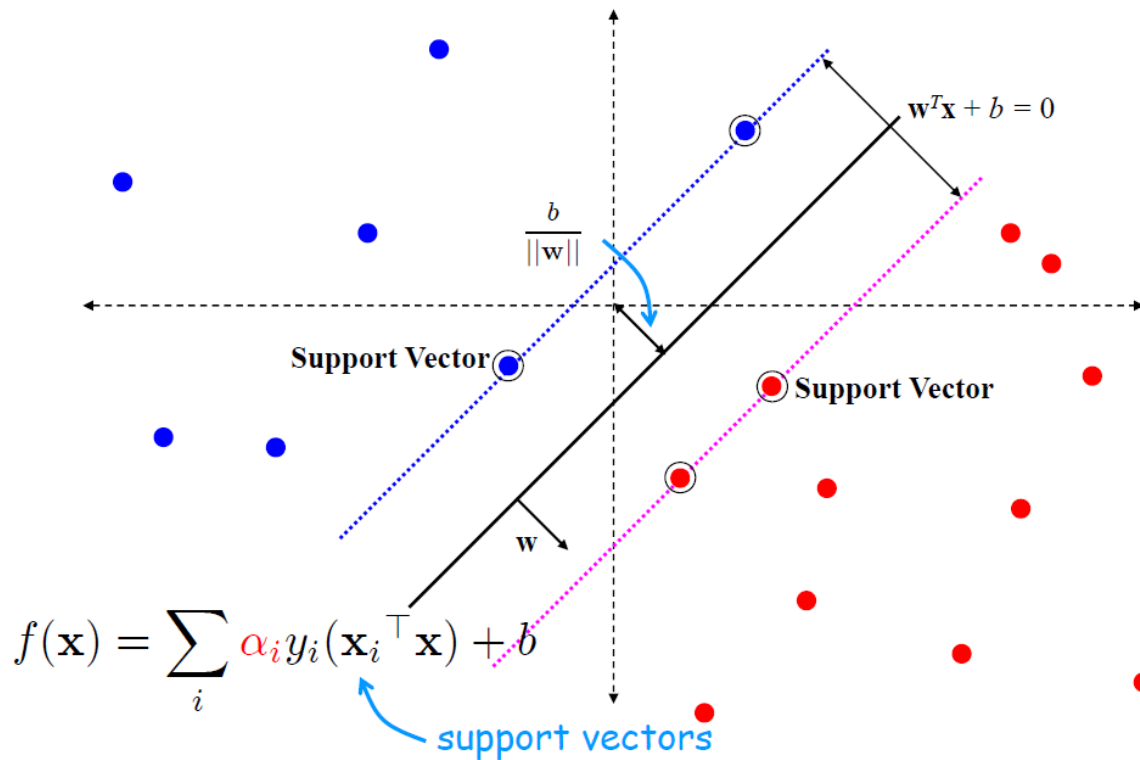
For a point \mathbf{x}^i on the hyperplane, since $y_i^2 = 1$,

$$y_i (\mathbf{w}^T \mathbf{x}^i + b) = 1 \Leftrightarrow \mathbf{w}^T \mathbf{x}^i + b = y_i$$

$$\Leftrightarrow b = y_i - \mathbf{w}^T \mathbf{x}^i$$

Supporting vectors

- Picture from “C19 Machine Learning Hilary 2015 A. Zisserman”



Dual LSVM Classifier

- DLSVM provides $\bar{\alpha} \in \mathbb{R}_+^N$ to form a classifier of bi-classification by taking $S = \{i \mid \bar{\alpha}_i > 0, i = 1, \dots, N\}$ and $\bar{b} = y_k - (\sum_{i \in S} \bar{\alpha}_i y_i \mathbf{x}^i)^T \mathbf{x}^k$ for any particular $k \in S$.

- Given an input data point $\mathbf{x} \in \mathbb{R}^n$

$$class_{DLSVM}(\mathbf{x}) = sign(\sum_{i \in S} \bar{\alpha}_i y_i (\mathbf{x}^i)^T \mathbf{x} + \bar{b})$$

where

$$sign(y) = \begin{cases} +1, & \text{if } y > 0 \\ -1, & \text{if } y < 0 \end{cases}$$

Primal LSVM vs. Dual LSVM

- SVM classifier

$$class_{SVM}(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$$

- Primal version (LSVM)

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \quad : \text{learning from data the normal vector and intercept}$$

- Dual version (DLSVM)

$$f(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i (\mathbf{x}^i)^T \mathbf{x} + \bar{b}$$

: learning from data the role of each data point

Primal LSVM vs. Dual LSVM

- Primal version (LSVM)

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

- Dual version (DLSVM)

$$f(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i (\mathbf{x}^i)^T \mathbf{x} + \bar{b}$$

Potentials of DLSVM:

1. Its **dimensionality** is **fixed** !

-- N variables and one linear equality constraint

-- solely **determined by** the number of data points N

-- **independent of** the size of each data point n .

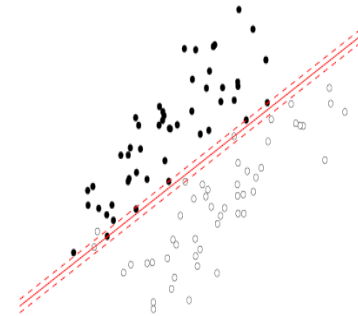
2. The **set** $S = \{\alpha_i \mid \alpha_i > 0\}$ is in general very **sparse**!

-- **easy to store and update**

Approximate LSVM considering generalizability

- **Basic Idea:** Open the margin to **allow violation with penalized tolerance.**
- **Original model**

$$\min \sum_{i=1}^N \max\{0, 1 - y_i(\mathbf{a}^T \mathbf{x}^i + b)\}$$



- **New model**

$$\min \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \max\{0, 1 - y_i(\mathbf{w}^T \mathbf{x}^i + b)\}$$

where $C > 0$ is a given parameter.

** C is an indicator emphasizing possible violations.

When $C \rightarrow +\infty$, new model returns to the original model.

Linear SVM with soft margin

- Reformulate the new model

$$\min \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \max\{0, 1 - y_i(\mathbf{w}^T \mathbf{x}^i + b)\}$$

by allowing violations $y_i(\mathbf{w}^T \mathbf{x}^i + b) < 1$ (a soft margin)

- Linear soft SVM

$$\min \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}^i + b) \geq 1 - \xi_i, i = 1, \dots, N \quad (\text{LSSVM})$$

$$\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}_+^N$$

where $C > 0$ is a given parameter.

** When $C \rightarrow +\infty$, $\xi \rightarrow \mathbf{0}$ and LSSVM becomes LSVM, but it may fail.

Linear soft SVM (LSSVM)

- Geometric meaning and complexity

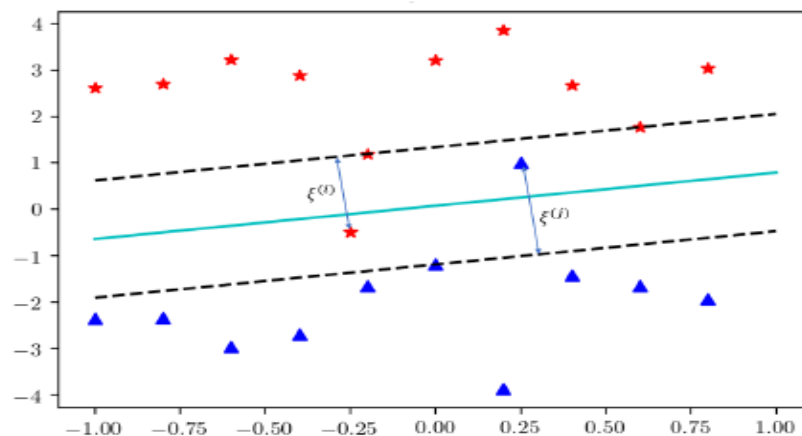
$$\min \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}^i + b) \geq 1 - \xi_i, i = 1, \dots, N \quad (\text{LSSVM})$$

$$\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}_+^N$$

where $C > 0$ is a given parameter.

- Linearly constrained convex quadratic program with $n + 1 + N$ variables and N inequality constraints.



LSVM vs. LSSVM

- **LSVM** works only for those **linearly separable** datasets.
 - Why?
- **LSSVM is always feasible** even a dataset is not linearly separable.
 - Why?
- For a **linearly separable** dataset, will LSVM and LSSVM produce the **same separation hyperplane**?
 - Why?
- LSSVM has **N more nonnegative variables** than LSVM.
What can we expect to meet for the dual LSSVM?
 - **N more constraints?**

Lagrangian dual approach

- Stationary point of the Lagrangian function

$$L(\mathbf{w}, b, \xi, \alpha, \theta) = \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i + \sum_{i=1}^N \alpha_i (1 - \xi_i - y_i(\mathbf{w}^T \mathbf{x}^i + b)) - \sum_{i=1}^N \theta_i \xi_i$$

where $\alpha_i \geq 0$ and $\theta_i \geq 0$.

Lagrangian dual function

$$h(\alpha, \theta) \triangleq \min_{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}_+^N} L(\mathbf{w}, b, \xi, \alpha, \theta)$$

Optimality conditions:

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \xi, \alpha, \theta) = 0 \implies \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}^i$$

$$\nabla_b L(\mathbf{w}, b, \xi, \alpha, \theta) = 0 \implies \sum_{i=1}^N \alpha_i y_i = 0$$

$$\nabla_{\xi} L(\mathbf{w}, b, \xi, \alpha, \theta) = 0 \implies C - \alpha_i = \theta_i \geq 0$$

$$\iff \alpha_i \leq C$$

\implies dual objective function

$$h(\alpha) = -\frac{1}{2} \left(\sum_{i=1}^N \alpha_i y_i \mathbf{x}^i \right)^T \sum_{j=1}^N \alpha_j y_j \mathbf{x}^j + \sum_{i=1}^N \alpha_i$$

Dual linear soft SVM (DLSSVM)

- Lagrangian dual model

$$\begin{aligned} \max \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i y_i ((\mathbf{x}^i)^T \mathbf{x}^j) y_j \alpha_j + \sum_i^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \quad (\text{DLSSVM}) \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \end{aligned}$$

- The Hessian of the objective function in α is

$$H = \text{Diag}(y) X^T X \text{Diag}(y) \succeq 0$$

- DLSSM is convex quadratic program with N bounded variables and 1 linear equality constraint.
- The quadratic term is determined by an $N \times N$ (kernel) matrix (in terms of the # of data points)

$$K = X^T X \text{ with } K_{ij} = (\mathbf{x}^i)^T \mathbf{x}^j \text{ (regardless the dimensionality of each data point } \mathbf{x}^i).$$

Relations of LSSVM and DLSSVM

- Key relations:

1. Convex QP pair means there is no duality gap!

2. Complementary slackness says that

$$\alpha_i(y_i(\mathbf{w}^T \mathbf{x}^i + b) - 1 + \xi_i) = 0, \forall i = 1, 2, \dots, N$$

- (a) $\alpha_i = 0$ holds for data point \mathbf{x}^i with $y_i(\mathbf{w}^T \mathbf{x}^i + b) > 1 - \xi_i$
(inactive constraint means such \mathbf{x}^i plays no role)

- (b) $C > \alpha_i > 0$ means the point \mathbf{x}^i with $y_i(\mathbf{w}^T \mathbf{x}^i + b) = 1 - \xi_i \leq 1$
(active constraint means \mathbf{x}^i is a supporting vector)

- (c) support vectors are \mathbf{x}^i s with $y_i(\mathbf{w}^T \mathbf{x}^i + b) \leq 1$ including those corresponding to $C > \alpha_i > 0$.

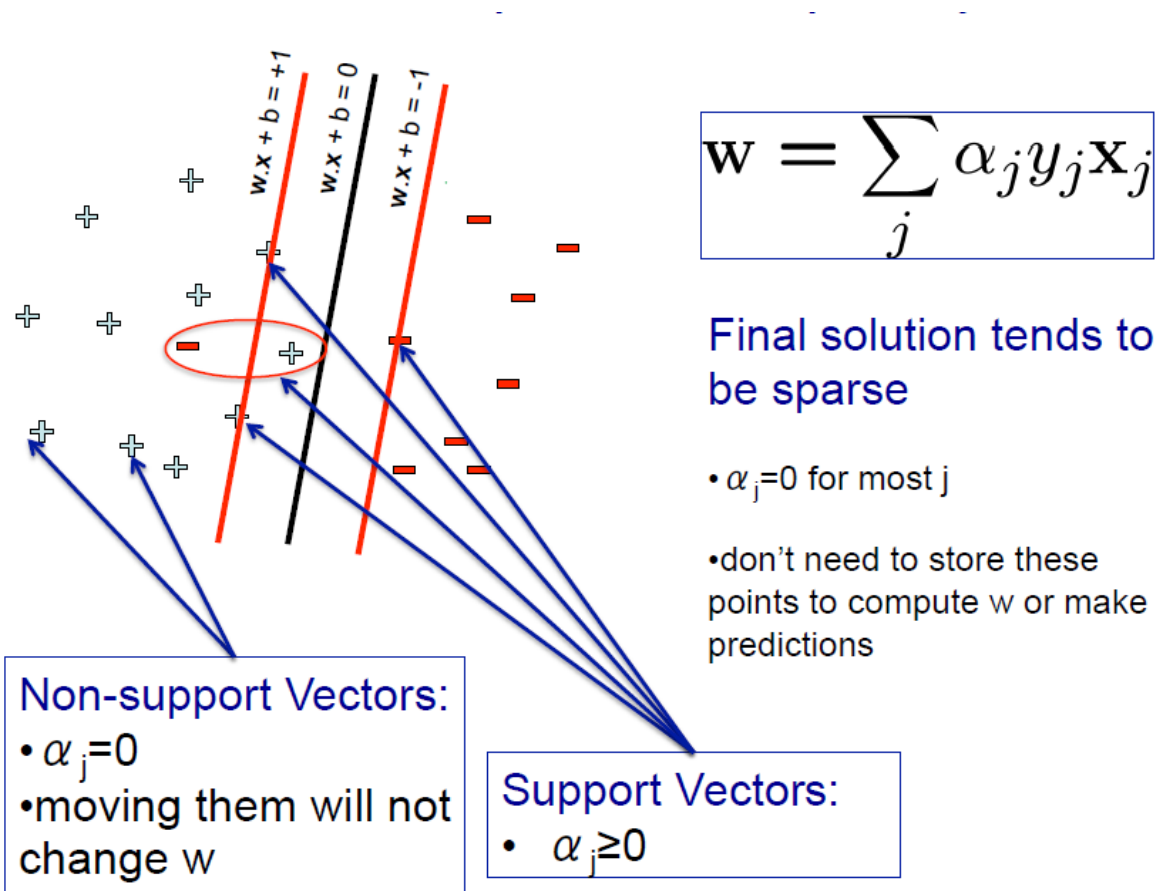
3. Dual to primal conversion says that

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

For a point \mathbf{x}^i on the hyperplane H_1 or H_{-1} , since $y_i^2 = 1$,
 $y_i(\mathbf{w}^T \mathbf{x}^i + b) = 1 \Leftrightarrow \mathbf{w}^T \mathbf{x}^i + b = y_i \Leftrightarrow b = y_i - \mathbf{w}^T \mathbf{x}^i$

Dual LSSVM

- Picture taken from David Sontag, SVM & Kernels Lecture 6.



Separation Hyperplane

- How to find the separation hyperplane?

$$(\mathbf{w}, b) = ?$$

- Supporting vectors:

$$S = \{\mathbf{x}^i \mid C_i > \alpha_i > 0\},$$

- Normal vector

$$\mathbf{w} = \sum_{i \in I_S} \alpha_i y_i \mathbf{x}_i, \text{ where } I_S = \text{the index set of } S$$

- Choose

$$I_1 = \text{the index set of all } \mathbf{x}^i \in S \text{ with } y_i = 1$$

- Find

$$i^* = \arg \max_{i \in I_1} \{\mathbf{w}^T \mathbf{x}^i\}$$

- Intercept

$$b = y_{i^*} - \mathbf{w}^T \mathbf{x}^{i^*}$$

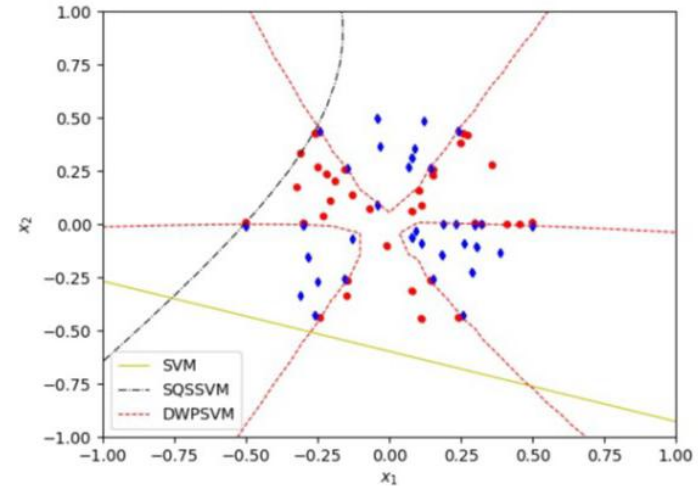
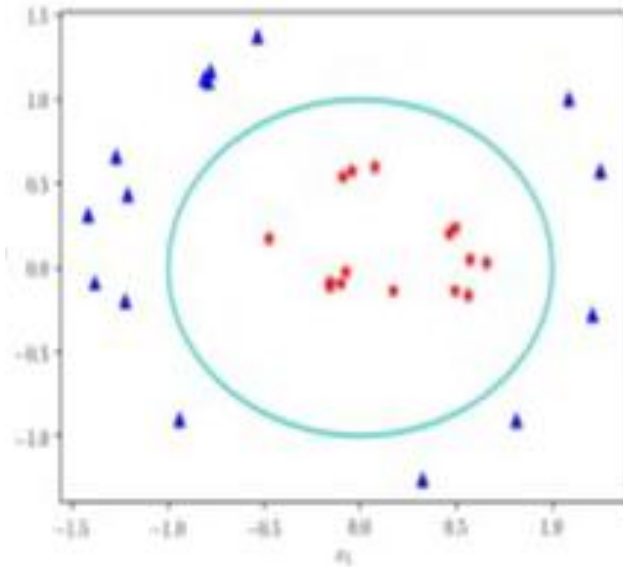
LSSVM vs. DLSSVM ?

- Which one to solve? Why?
 - LSSVM or DLSSM?
 - how about $n \gg N$ and $N \gg n$?
- What's the effect of choosing different parameter value of C ?
- Classifier?
 - $class_{LSSVM}(\mathbf{x}) = ?$
 - $class_{DLSSVM}(\mathbf{x}) = ?$

Comparisons and discussions

- LSVM vs. Approximate LSVM
 - applicability?
 - equivalency?
 - complexity?
- LSVM vs. LSSVM
- LSSVM vs. Approximate LSVM

SVM for not linearly separable data sets



- Will LSVM, Approximate LSVM, LSSVM work ?
- How well can they be?
- Any better SVM classifier?

SVM for not linearly separable data sets

- Basic ideas:

1. Reformulate the problem in a higher dimensional space for linear separability

(Kernel Method): LSVM with kernel functions

2. Adopt nonlinear surface to separate data points apart in the original space

- Quadratic surface SVM

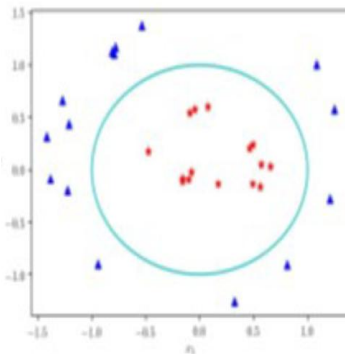
- Double-well potential function based SVM

Idea of kernel based SVM

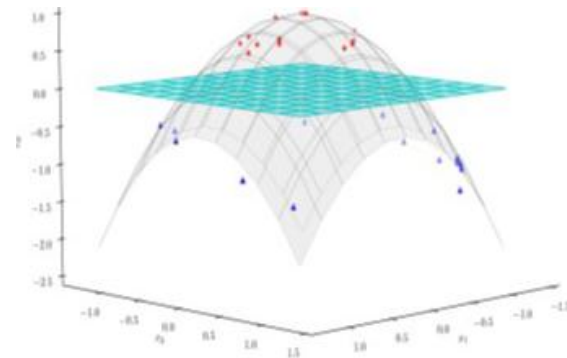
- **Feature map:** a function $\phi(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^l$, with $l \geq n$, that maps all data points to a higher dimensional space for linear separation.

- Example 1: $\|\mathbf{x}\|_2^2 < 1$, $\|\mathbf{x}\|_2^2 > 1$,

$$\phi_1(\mathbf{x}): \mathbb{R}^2 \rightarrow \mathbb{R}^3, \phi_1(\mathbf{x}) = \phi_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ 1 - x_1^2 - x_2^2 \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ 1 - \|\mathbf{x}\|^2 \end{pmatrix}$$



$\phi \rightarrow$



Kernel-based soft SVM - KSSVM

- Using a *feature map* $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^l$ ($l \geq n$) to transform the problem to a higher dimensional space for linear separability.
- **Build upon LSSVM**
- Primal model

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i (\mathbf{w}^T \phi(\mathbf{x}^i) + b) \geq 1 - \xi_i, i = 1, \dots, N \quad (\text{KSSVM}) \\ & \mathbf{w} \in \mathbb{R}^l, b \in \mathbb{R}, \xi \in \mathbb{R}_+^N \\ & \text{where } C > 0 \text{ is a given parameter.} \end{aligned}$$

** More variables involved than using LSSVM.

How difficult to solve DKSSVM?

- Lagrangian dual model

$$\begin{aligned} \max & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i y_i K_{ij} y_j \alpha_j + \sum_i^N \alpha_i \\ \text{s.t.} & \sum_{i=1}^N \alpha_i y_i = 0 && \text{(DKSSVM)} \\ & 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N \end{aligned}$$

where $K_{ij} = K(\mathbf{x}^i, \mathbf{x}^j) \triangleq \phi(\mathbf{x}^i)^T \phi(\mathbf{x}^j)$

- Given any feature map ϕ , corresponding K is *psd* and DKSSVM becomes a **convex quadratic program** with N **bounded variables** and **only one linear equality constraint**.
- In practice, we may **use a kernel matrix** $K = (K_{ij})$ **without knowing the feature map** $\phi(x)$.

Separation Hyperplane

- How to find the separation hyperplane in the feature space?

$$(\mathbf{w}, b) = ?$$

- Supporting vectors: $S = \{\boldsymbol{\phi}(\mathbf{x}^i) \mid C_i > \alpha_i > 0\}$,

- Normal vector

$$\mathbf{w} = \sum_{i \in I_S} \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i), \text{ where } I_S = \text{the index set of } S$$

- Choose

$$I_1 = \text{the index set of all } \boldsymbol{\phi}(\mathbf{x}^i) \in S \text{ with } y_i = 1$$

$$i^* = \arg \max_{i \in I_1} \{\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}^i)\}$$

- Intercept $b = y_{i^*} - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}^{i^*})$

- Map the hyperplane back to the original space via ϕ^{-1} ? If possible.

Kernel-based soft SVM - DKSSVM

- SVM classifier

$$\text{class}_{SVM}(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$$

Dual version DKSSVM

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}^i)^T \phi(\mathbf{x}) + b(\alpha_i) \\ &= \sum_{i \in S} \alpha_i y_i K(\mathbf{x}^i, \mathbf{x}) + \bar{b} \end{aligned}$$

Kernel matrix

- To make sure that $K_{ij} = K(\mathbf{x}^i, \mathbf{x}^j)$ is the inner product of $\phi(\mathbf{x}^i)$ and $\phi(\mathbf{x}^j)$ in the feature space, such that
 - (1) DKSSVM is an easily solved convex QP,
 - (2) there is a chance to solve KSSVM,
we **need K to be symmetric and positive semidefinite**
(Mercer's condition).

- **Commonly used kernels:**

1. **Polynomial kernel** of degree $d = 1, 2, \dots$

$$K(\mathbf{x}^i, \mathbf{x}^j) = ((\mathbf{x}^i)^T \mathbf{x}^j + r)^d \quad \begin{array}{l} \text{(homogeneous, if } r = 0) \\ \text{(inhomogeneous, if } r > 0) \end{array}$$

** popular in image processing*

Polynomial kernels

- Example 1: (inhomogeneous degree 2)

For $\mathbf{x} \in \mathbb{R}^1$, $K(x^i, x^j) = (x^i x^j + 1)^2$ for $r = 1, d = 2$,

we have $\phi(x)^T = (1, \sqrt{2}x, x^2) \in \mathbb{R}^3$ such that

$$\phi(x^i)^T \phi(x^j) = 1 + 2x^i x^j + (x^i)^2 (x^j)^2 = (x^i x^j + 1)^2$$

Example 2: (homogeneous degree 2)

For $\mathbf{x} \in \mathbb{R}^2$, $K(\mathbf{x}^i, \mathbf{x}^j) = ((\mathbf{x}^i)^T \mathbf{x}^j)^2$ for $r = 0, d = 2$,

we have $\phi(\mathbf{x})^T = (x_1^2, \sqrt{2}x_1 x_2, x_2^2) \in \mathbb{R}^3$ such that

$$\phi(\mathbf{x}^i)^T \phi(\mathbf{x}^j) = (x_1^i)^2 (x_1^j)^2 + (x_2^i)^2 (x_2^j)^2 + 2(x_1^i x_2^i x_1^j x_2^j) = ((\mathbf{x}^i)^T \mathbf{x}^j)^2$$

**General form $\phi(x)$: contains all polynomial terms up to degree d .

Kernel matrix

Commonly used kernels:

2. Gaussian kernel with $\sigma \in \mathbb{R} \setminus \{0\}$

$$K(\mathbf{x}^i, \mathbf{x}^j) = \exp \left(- \frac{\|\mathbf{x}^i - \mathbf{x}^j\|_2^2}{2\sigma^2} \right)$$

* no prior information, general purpose

** *General form $\phi(x)$ in infinite dimensional feature space.*

3. Gaussian Radial basis function (RBF) kernel with $\gamma > 0$

$$K(\mathbf{x}^i, \mathbf{x}^j) = \exp \left(-\gamma \|\mathbf{x}^i - \mathbf{x}^j\|_2^2 \right)$$

* no prior information, general purpose

** *General form $\phi(x)$: see https://en.wikipedia.org/wiki/Radial_basis_function_kernel*

Kernel matrix

Commonly used kernels:

4. Laplace RBF kernel with $\sigma > 0$

$$K(\mathbf{x}^i, \mathbf{x}^j) = \exp \left(-1/\sigma \|\mathbf{x}^i - \mathbf{x}^j\|_2 \right)$$

* no prior information, general purpose

5. Sigmoid kernel with $\beta > 0$, $\theta \in \mathbb{R}$

$$K(\mathbf{x}^i, \mathbf{x}^j) = \tanh \left(\beta (\mathbf{x}^i)^T \mathbf{x}^j + \theta \right)$$

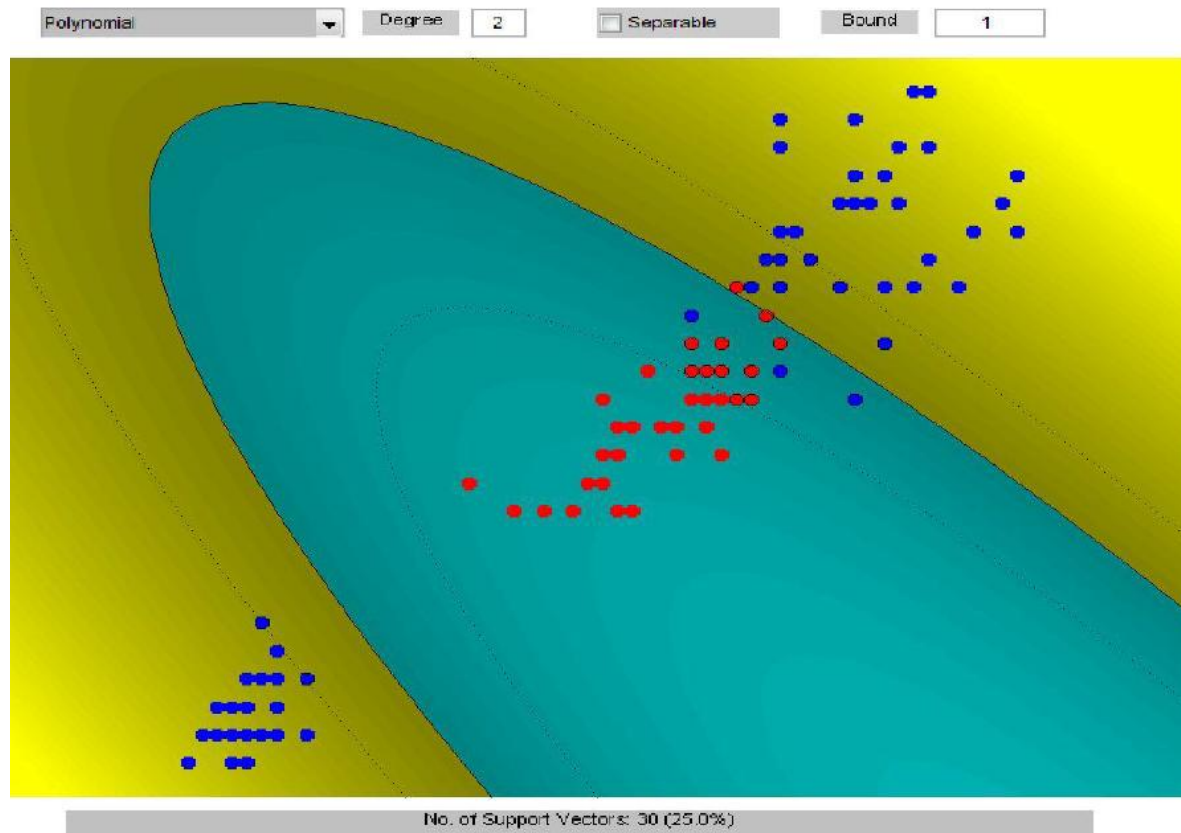
* proxy for neural networks

Quality of kernel-based SVM

- Two major factors:
 1. Like LSSVM, the **parameter C** plays a role.
 2. The choice of an appropriate **kernel matrix** (and **its parameters**) is important.

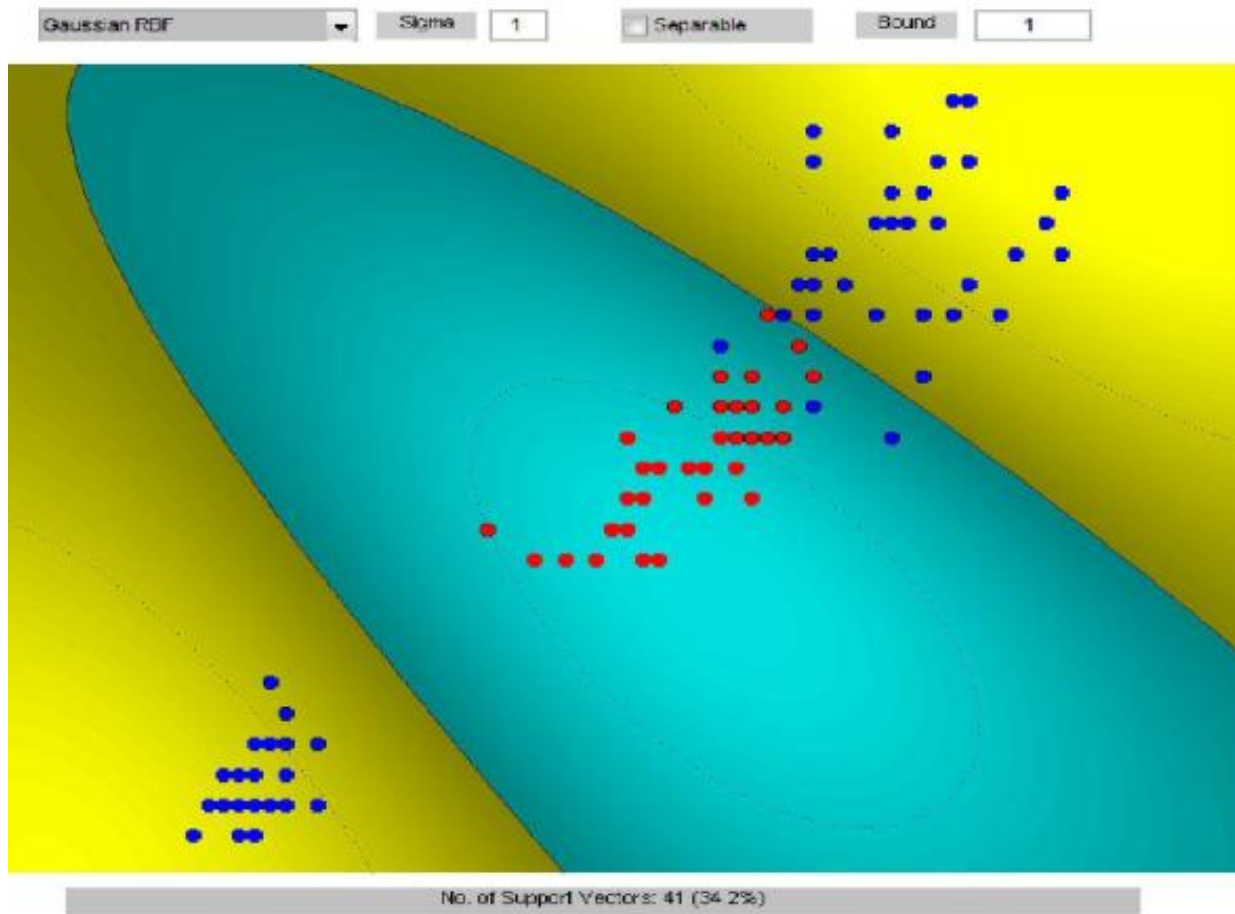
Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020, CMU
Iris dataset, 1 vs 23, Polynomial Kernel degree 2 ($C = 1$)



Effect of kernel matrix

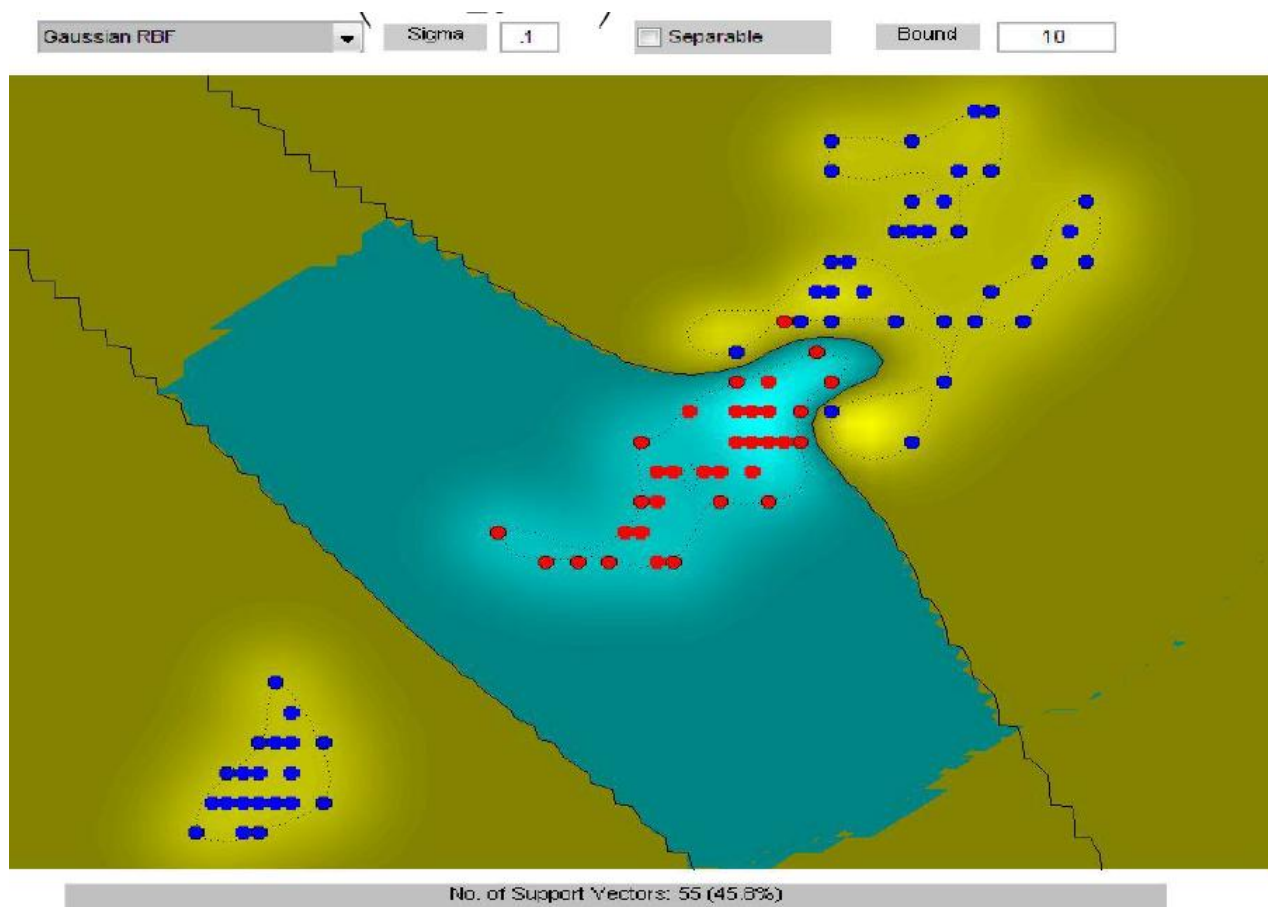
- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020, CMU
Iris dataset, 1 vs 23, Gaussian RBF kernel ($C = 1, \sigma = 1$)



Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020, CMU

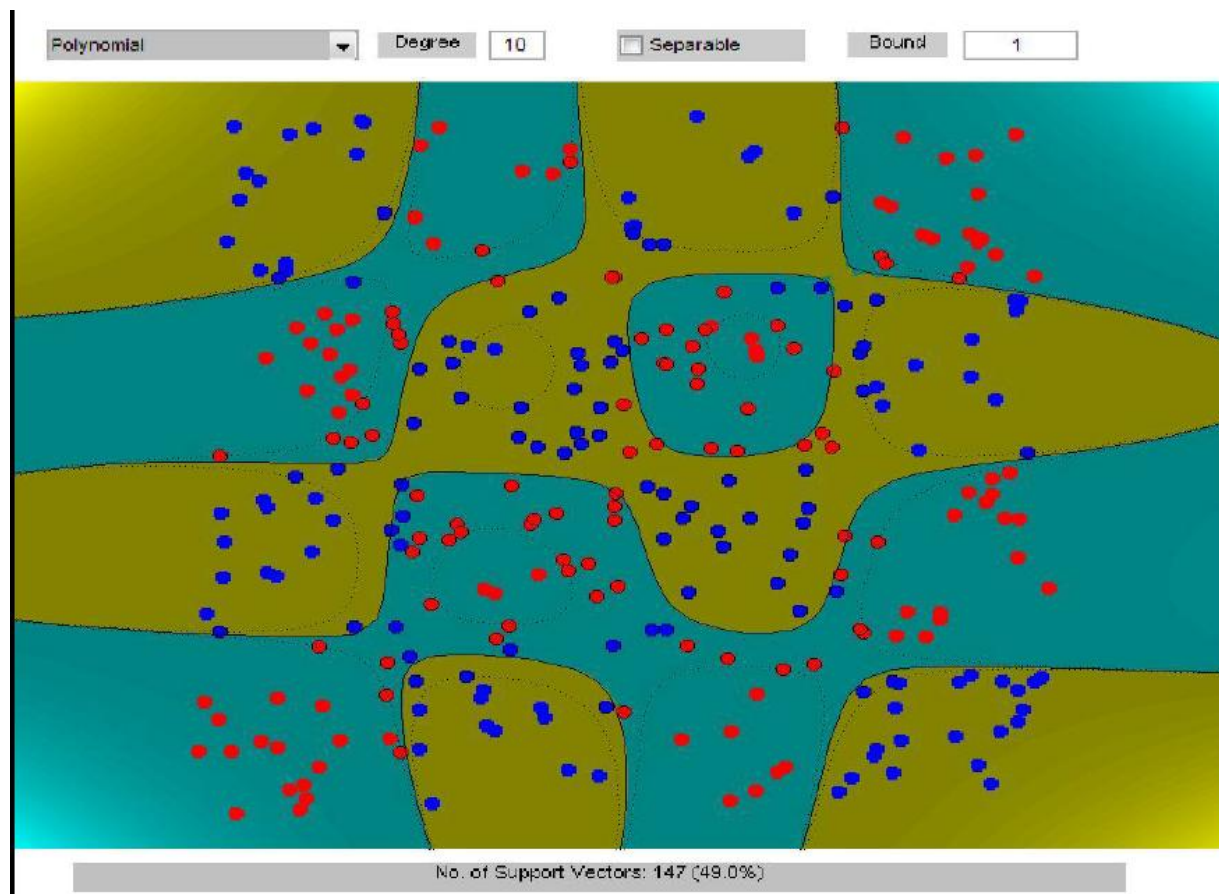
Iris dataset, 1 vs 23, Gaussian RBF kernel ($C = 10, \sigma = 1$)



Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020, CMU

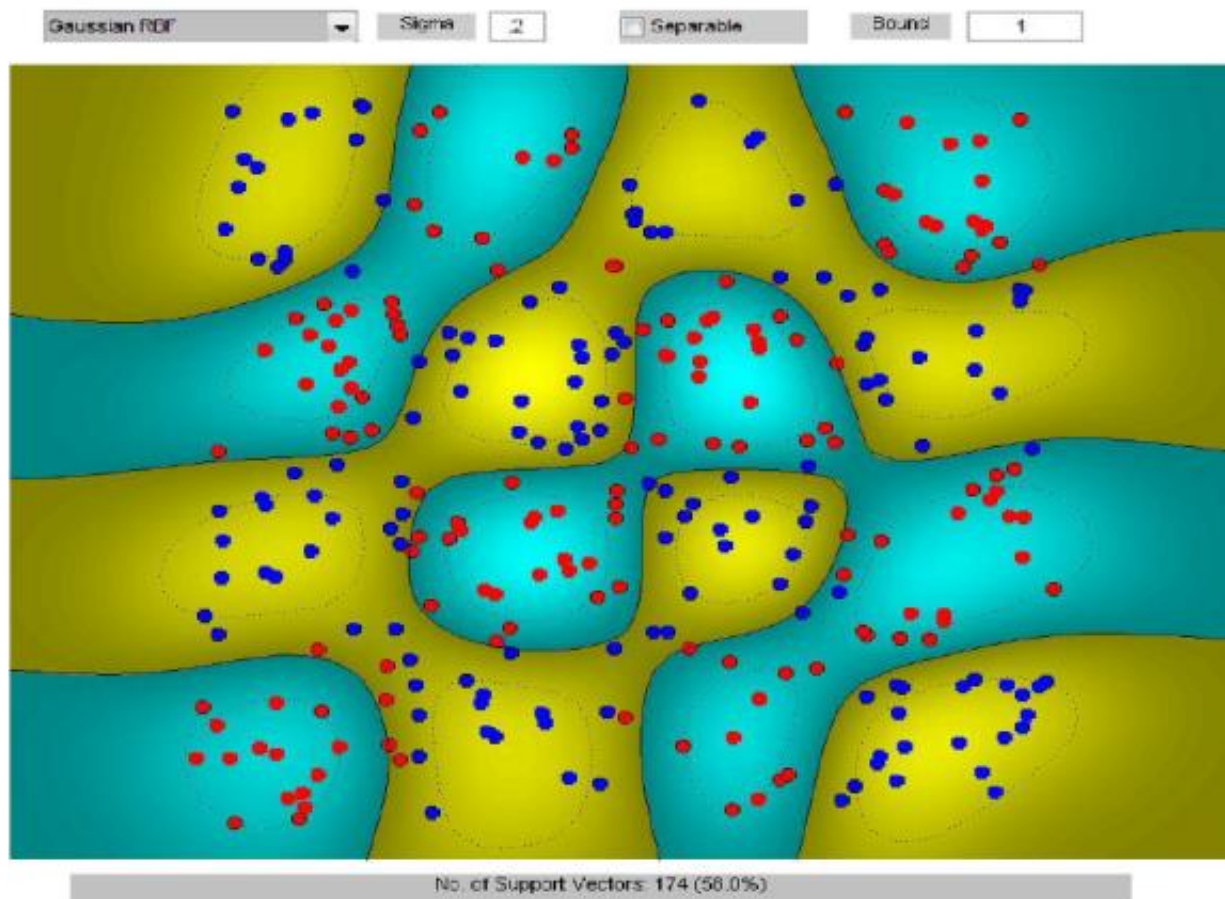
Chessboard dataset, Polynomial kernel ($d = 10, C = 1$)



Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020, CMU

Chessboard dataset, Gaussian RBF kernel ($C = 1, \sigma = 2$)



Quality of kernel-based SVM

- Two major factors:
 1. Like LSSVM, the **parameter C** plays a role.
 2. The choice of an appropriate **kernel matrix** (and **its parameters**) is important.

Question: How to choose/design right ones?

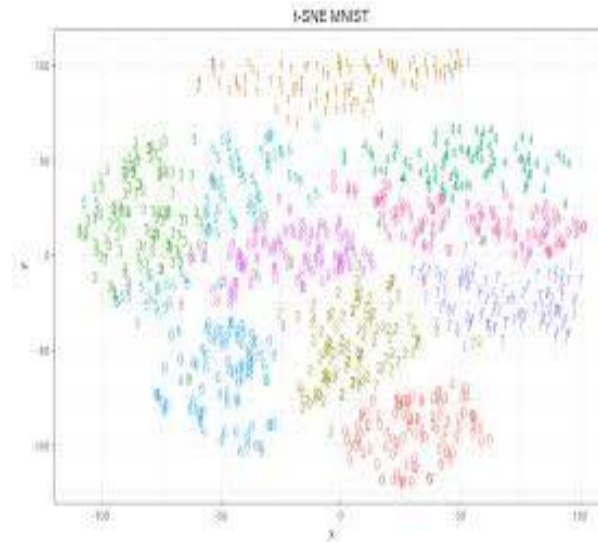
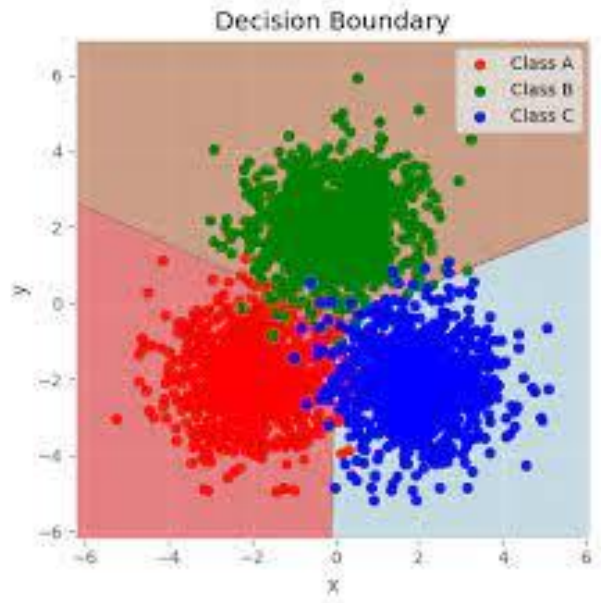
- theoretical analysis?
- **computational experiments !**

Ideas of choosing parameters

- **Example:** choosing parameter C
 1. Define an **error or score measure**:
for example, MSE (mean squares error),
MAPE (mean absolute percentage error),
 $1/\|\mathbf{w}\|_2^2$, or $\sum_{i=1}^N y_i (\mathbf{w}^T \mathbf{x}^i + b)$, ...
 2. Conduct **computational experiments** with different value of C :
 - **statistically meaningful**
 3. Plot resulting error measures against C .
 4. Find the **elbow/ turning point** value of C .
- ** check many other “cross-validation” methods.

Multi-class classification

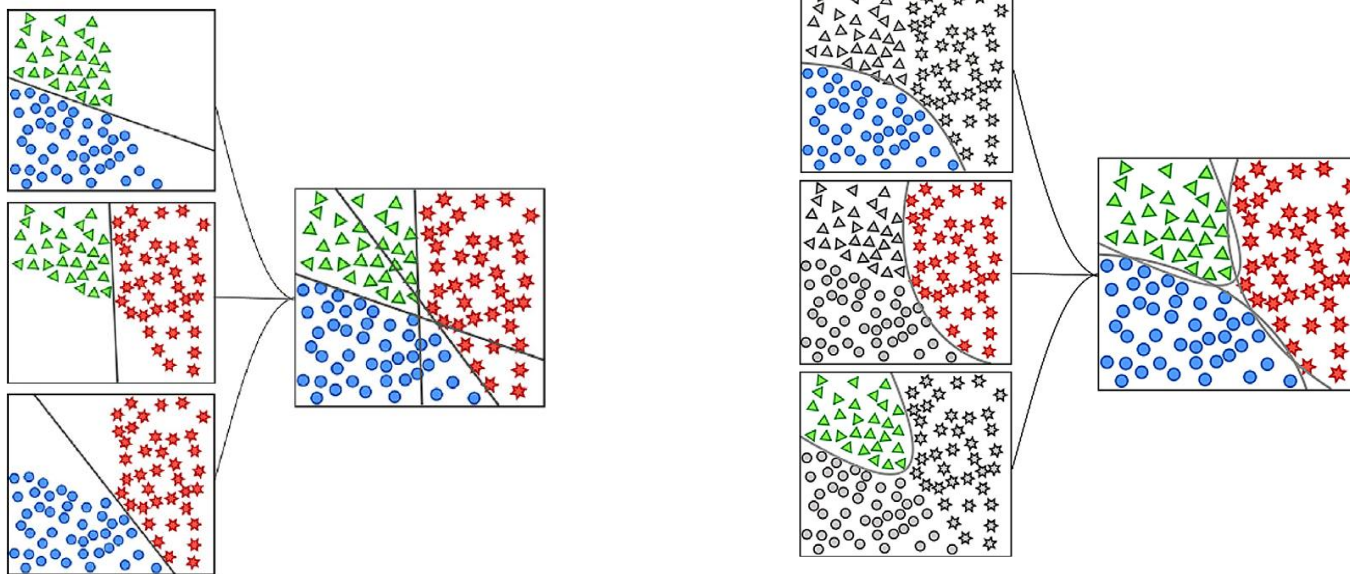
- **Multi-class classification** (multi-classification) is a problem of classifying instances into one of three or more classes.



- Pictures taken from Wikipedia

Basic ideas of multiclass classification

- If we are given a dataset in C classes and a new data-point $x \in \mathbb{R}^n$, we may consider two commonly adopted approaches to determine which class x belongs to:
 1. OVO (one vs. one) approach
 2. OVA (one vs. all) approach



Basic ideas of multiclass classification

- 1. OVO (one vs. one) approach

- Take each pair of different classes (C_i, C_j) .
- Label all data-points in C_i with a label $+1$;
Label all data-points in C_j with a label -1 .
- Apply a bi-class SVM classifier to find (\mathbf{w}, b)
and determine $\mathbf{x} \in C_i$ or $\mathbf{x} \in C_j$.
- Assign a voting score to each class of the pair

$$Score(C_i) = \begin{cases} 1, & \text{if } \mathbf{x} \in C_i \\ 0, & \text{if } \mathbf{x} \in C_j \end{cases}$$

- Sum up scores/votes over all pairs involving C_i .

- **Decision:** \mathbf{x} belongs to the class with the highest total score.

OVO related issues

- How many SVM involved?

of all possible pairs = $c(c-1)/2$

- Tie breaker?

Secondary score?

- Better scoring measure?

Basic ideas of multiclass classification

- 2. OVA (one vs. all of the rest) approach
 - Take each class C_i .
 - Label all data-points **in** C_i with a label +1;
Label all data-points **not in** C_i with a label - 1.
 - Apply a bi-class SVM classifier to find (\mathbf{w}^i, b^i)
and determine $\mathbf{x} \in C_i$ or $\mathbf{x} \notin C_i$.
 - Assign a score to each class
$$\text{Score}(C_i) = (\mathbf{w}^i)^T \mathbf{x} + b^i$$
 - **Decision:** \mathbf{x} belongs to the class with highest score.

OVA related issues

- How many SVM involved?
of all possible pairs = c ($\ll c(c-1)/2$)
- Imbalanced datasets
quality of results?
- Better scoring ?

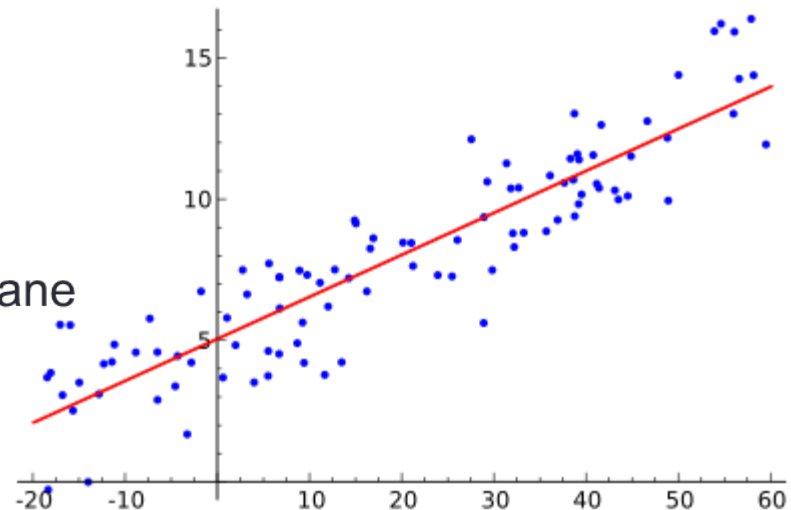
- Any approaches other than OVO and OVA ?
- tournament ?
- Which SVM model to use?

Regression

- Linear regression:
 - a model that assumes a linear relationship between the input variables (\mathbf{x}) and the single output variable (y) such that y can be calculated from a linear combination of the input variables (\mathbf{x}).

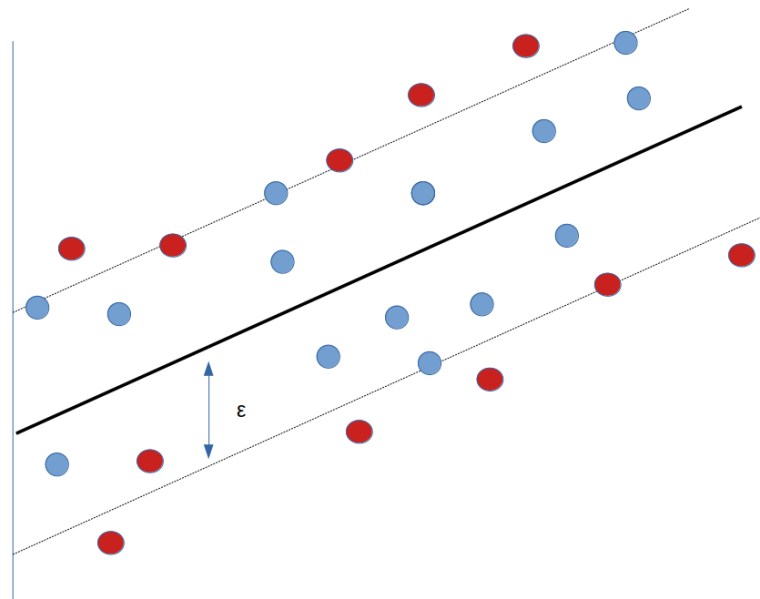
$$y = \mathbf{w}^T \mathbf{x} + b + \varepsilon$$

Fit the data to a supporting hyperplane with the minimum mean squared error.



Support vector regression (SVR)

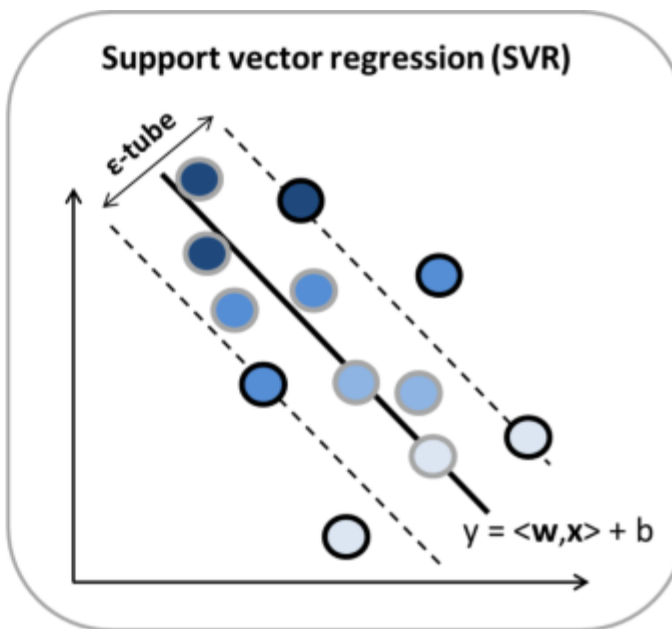
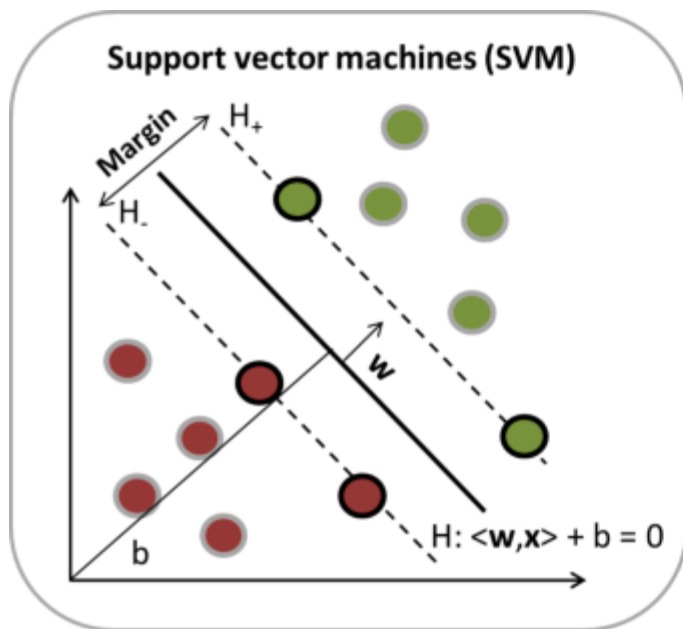
- Basic idea:
 - Find a hyperplane centering around the data by boxing as many data points as possible in a given tube around the hyperplane.



Support vector machines: classification vs. regression

- SVM: data-points in \mathbb{R}^n
with class label y
separate data-points
apart

- SVR: data-points in \mathbb{R}^{n+1}
with data value y
box data-points in a tube



Linear support vector regression

- Problem settings:

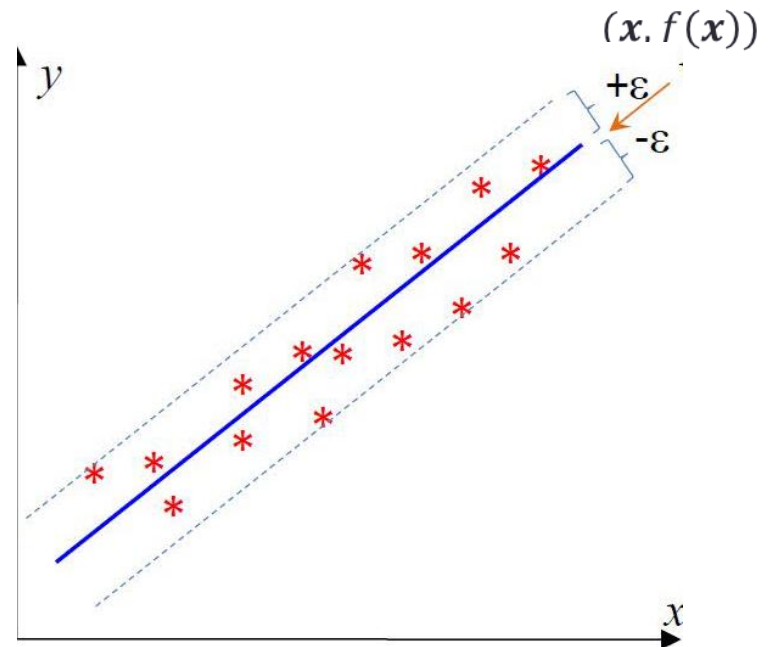
- Dataset $\{ (\mathbf{x}^i, y_i) \in \mathbb{R}^n \times \mathbb{R} \mid i = 1, 2, \dots, N \}$ of N data points
- tube tolerance $\varepsilon > 0$

- Aim: to find

affine map $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$

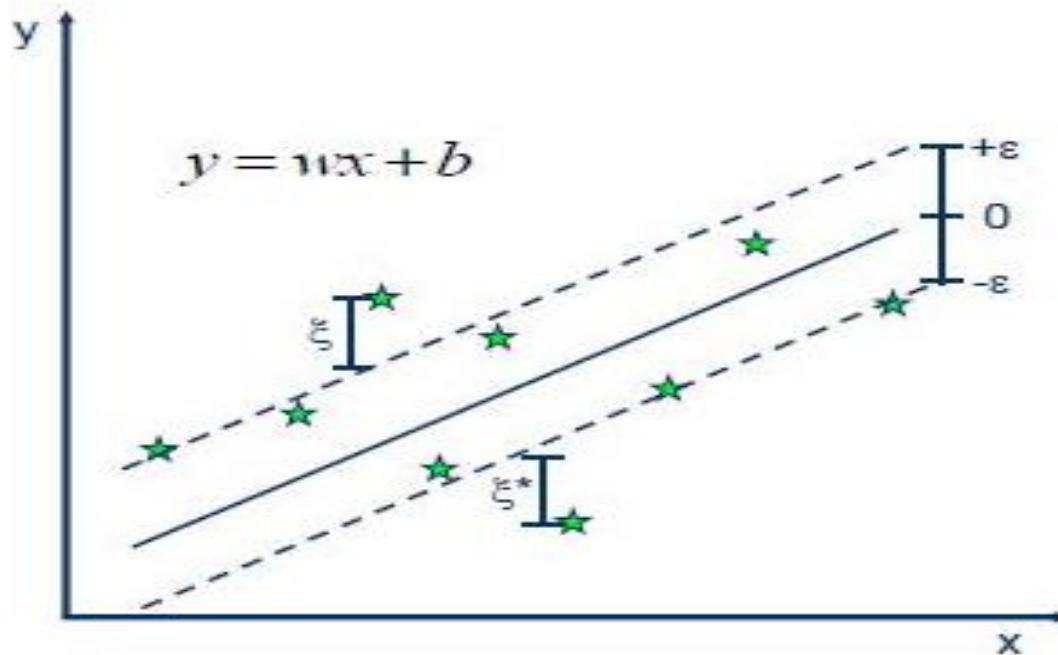
with **wide margin** such that

$$|y_i - f(\mathbf{x}^i)| < \varepsilon, i = 1, \dots, N$$



Observation

- **Question:** How big the box tolerance ε should be?
 - When $\varepsilon (> 0)$ is too small, we may not be able to box all data-points in the tube.



Linear soft support vector regression

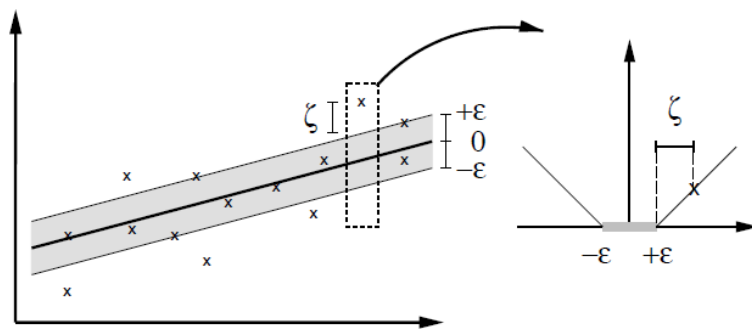
- Primal model: (For a given $C > 0$)

$$\text{Min} \quad \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t.} \quad y_i - \mathbf{w}^T \mathbf{x}^i - b \leq \varepsilon + \xi_i, \quad i = 1, \dots, N \quad (\text{LSSVR})$$

$$y_i - \mathbf{w}^T \mathbf{x}^i - b \geq -\varepsilon - \xi_i, \quad i = 1, \dots, N$$

$$\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}_+^N$$



soft margin with ε – *insensitive loss* function

Linear soft SVR - LSSVR

1. (LSSVR) is a **convex quadratic program** with $n + 1$ free variables, N non-negative variables, and $2N$ linear inequality constraints.
2. (LSSVR) is **always feasible**.
3. **Who are supporting vectors?**
4. **Any dual information?**

Dual LSSVR - DLSSVR

- Lagrangian

$$\begin{aligned} L(\mathbf{w}, b, \xi, \alpha, \alpha^*, \eta) &= \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i \\ &\quad - \sum_{i=1}^N \eta_i \xi_i - \sum_{i=1}^N \alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w}^T \mathbf{x}^i + b) \\ &\quad - \sum_{i=1}^N \alpha_i^* (\varepsilon + \xi_i + y_i - \mathbf{w}^T \mathbf{x}^i - b) \end{aligned}$$

- KKT conditions

- Primal & dual feasibility

(i) $\alpha_i, \alpha_i^*, \eta_i \geq 0, i = 1, \dots, N;$

(ii) $\varepsilon + \xi_i - y_i + \mathbf{w}^T \mathbf{x}^i + b \geq 0; \varepsilon + \xi_i + y_i - \mathbf{w}^T \mathbf{x}^i - b \geq 0;$

Dual LSSVR - DLSSVR

- Lagrangian

$$\begin{aligned} L(\mathbf{w}, b, \xi, \alpha, \alpha^*, \eta) &= \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i \\ &\quad - \sum_{i=1}^N \eta_i \xi_i - \sum_{i=1}^N \alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w}^T \mathbf{x}^i + b) \\ &\quad - \sum_{i=1}^N \alpha_i^* (\varepsilon + \xi_i + y_i - \mathbf{w}^T \mathbf{x}^i - b) \end{aligned}$$

- KKT conditions

Stationarity

(iii) $\nabla_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^N (\alpha_i - \alpha_i^*) \mathbf{x}^i = 0;$

(iv) $\nabla_b L = \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0;$

(v) $\nabla_{\xi_i} L = C - \eta_i - (\alpha_i + \alpha_i^*) = 0;$

$$\Rightarrow \eta_i = C - (\alpha_i + \alpha_i^*) \geq 0 \text{ and } 0 \leq \alpha_i + \alpha_i^* \leq C$$

Dual soft support vector regression -DLSSVR

- Dual model:

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) \langle \mathbf{x}^i, \mathbf{x}^j \rangle + (\alpha_j - \alpha_j^*) \\ & -\varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad (\text{DLSSVR}) \\ & 0 \leq \alpha_i + \alpha_i^* \leq C, \alpha_i \geq 0, \alpha_i^* \geq 0, i = 1, \dots, N \end{aligned}$$

Depending on $y_i > \mathbf{w}^T \mathbf{x}^i + b$, or $y_i < \mathbf{w}^T \mathbf{x}^i + b$, at least one of α_i or $\alpha_i^ = 0$. So we have

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) \langle \mathbf{x}^i, \mathbf{x}^j \rangle + (\alpha_j - \alpha_j^*) \\ & -\varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad (\text{DLSSVR}) \\ & 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C, i = 1, \dots, N \end{aligned}$$

Dual soft support vector regression -DLSSVR

- Observations:

1. (DLSSVR) is a **convex quadratic program** with $2N$ bounded variables and **1 linear equality constraint**.

2. (DLSSVR) is **independent of the size of n** , which is absolved in the inner product of

$$(\mathbf{x}^i)^T \mathbf{x}^j = \langle \mathbf{x}^i, \mathbf{x}^j \rangle.$$

DLSSVR

- Dual-to-primal conversion:
- KKT (iii) say that

$$\nabla_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^N (\alpha_i - \alpha_i^*) \mathbf{x}^i = 0.$$

Hence,

$$\mathbf{w} = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \mathbf{x}^i \text{ and}$$

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \langle \mathbf{x}^i, \mathbf{x} \rangle + b$$

* This is called a “support vector expansion” of $f(\mathbf{x})$.

* What is b ?

DLSSVR

- KKT conditions: Complementary slackness:

$$(vi) \alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w}^T \mathbf{x}^i + b) = 0$$

$$(vii) \alpha_i^* (\varepsilon + \xi_i + y_i - \mathbf{w}^T \mathbf{x}^i - b) = 0$$

$$(viii) \eta_i \xi_i = (C - (\alpha_i + \alpha_i^*)) \xi_i = 0$$

Observations:

1. Depend on $y_i > \mathbf{w}^T \mathbf{x}^i + b$, or $y_i < \mathbf{w}^T \mathbf{x}^i + b$,
at least **one of α_i or $\alpha_i^* = 0$** .

2. When data-point (\mathbf{x}^i, y_i) is **in the tube**

$$|y_i - (\mathbf{w}^T \mathbf{x}^i + b)| < \varepsilon \Rightarrow \alpha_i = 0 \text{ and } \alpha_i^* = 0.$$

DLSSVR

- KKT conditions: Complementary slackness:

$$(vi) \alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w}^T \mathbf{x}^i + b) = 0$$

$$(vii) \alpha_i^* (\varepsilon + \xi_i + y_i - \mathbf{w}^T \mathbf{x}^i - b) = 0$$

$$(viii) \eta_i \xi_i = (C - (\alpha_i + \alpha_i^*)) \xi_i = 0$$

Observations:

3. When data-point (\mathbf{x}^i, y_i) is **outside of the tube**,

$$|y_i - (\mathbf{w}^T \mathbf{x}^i + b)| > \varepsilon \Rightarrow \xi_i > 0 \Rightarrow \alpha_i = C \text{ or } \alpha_i^* = C.$$

4. $\alpha_i \in (0, C)$ or $\alpha_i^* \in (0, C)$ happens only when (\mathbf{x}^i, y_i) lies **on the tube**

$$|y_i - (\mathbf{w}^T \mathbf{x}^i + b)| = \varepsilon$$

$$\Rightarrow \text{either } y_i - (\mathbf{w}^T \mathbf{x}^i + b) = \varepsilon \Rightarrow b = \varepsilon - y_i + \mathbf{w}^T \mathbf{x}^i, \text{ when } \alpha_i \in (0, C)$$

$$\text{or } y_i - (\mathbf{w}^T \mathbf{x}^i + b) = -\varepsilon \Rightarrow b = -\varepsilon - y_i + \mathbf{w}^T \mathbf{x}^i, \text{ when } \alpha_i^* \in (0, C)$$

5. **Supporting vectors are indeed sparse!**

DLSSVR

- Dual-to-primal conversion:
- KKT (iii) say that

$$\nabla_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^N (\alpha_i - \alpha_i^*) \mathbf{x}^i = 0.$$

Hence,

$$\mathbf{w} = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \mathbf{x}^i$$

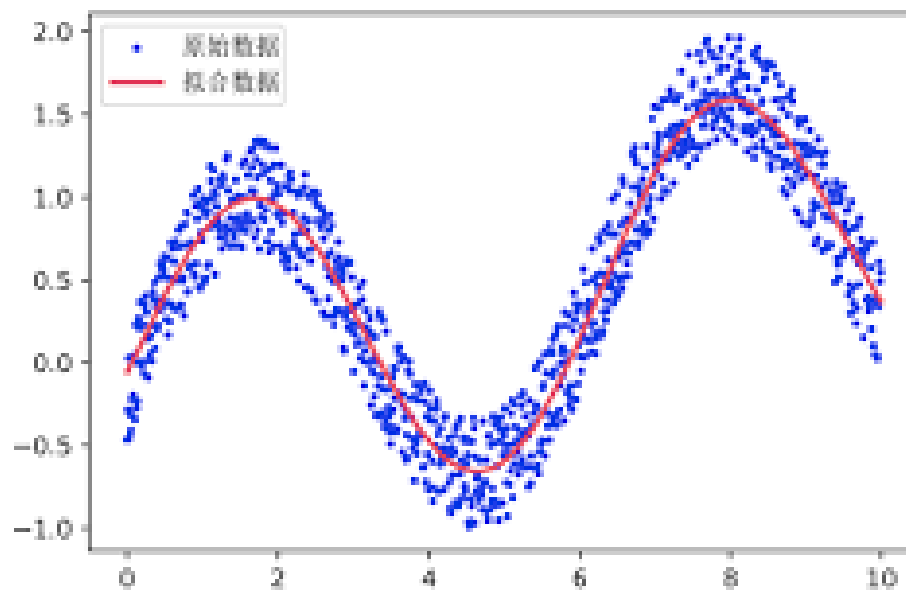
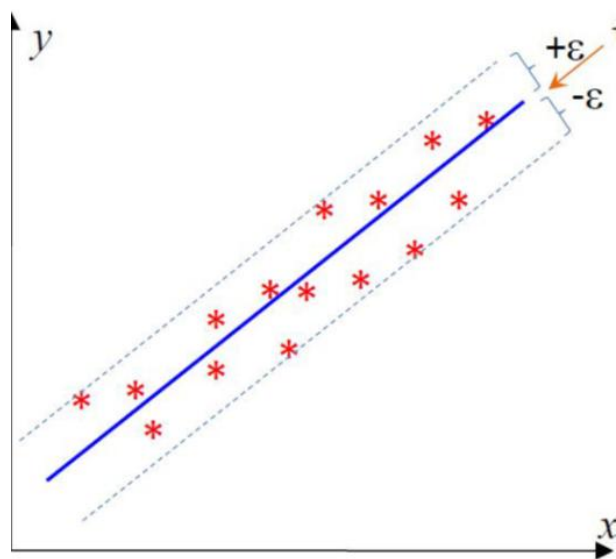
$$\mathbf{b} = \begin{pmatrix} \varepsilon - y_i + \mathbf{w}^T \mathbf{x}^i, & \text{if } \alpha_i \in (0, C) \\ -\varepsilon - y_i + \mathbf{w}^T \mathbf{x}^i, & \text{if } \alpha_i^* \in (0, C) \end{pmatrix}$$

and DLSSVR prediction is

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \langle \mathbf{x}^i, \mathbf{x} \rangle + b$$

SVM-based nonlinear regression

- From linear to nonlinear regression



Kernel-based linear soft SVR

- Use a *feature map* $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^l$ ($l \geq n$) to transform the problem to a higher dimensional space for linear separability.
- **Primal model:** (For a given $C > 0$)

$$\text{Min} \quad \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t.} \quad y_i - \mathbf{w}^T \phi(\mathbf{x}^i) - b \leq \varepsilon + \xi_i, \quad i = 1, \dots, N \quad (\text{KLSSVR})$$

$$y_i - \mathbf{w}^T \phi(\mathbf{x}^i) - b \geq -\varepsilon - \xi_i, \quad i = 1, \dots, N$$

$$\mathbf{w} \in \mathbb{R}^l, b \in \mathbb{R}, \xi \in \mathbb{R}_+^N$$

* Dimensionality changes from n to l .

Dual kernel-based linear soft support vector regression

- Dual model:

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) < \phi(\mathbf{x}^i), \phi(\mathbf{x}^j) > (\alpha_j - \alpha_j^*) \\ & -\varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad \text{(DKLSSVR)} \\ & 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C, i = 1, \dots, N \end{aligned}$$

*(DKLSSVR) is a convex quadratic program with $2N$ bounded variables and 1 linear equality constraint.

*(DKLSSVR) is independent of the size of n , which is absolved in the inner product of

$$\phi(\mathbf{x}^i)^T \phi(\mathbf{x}^j) = < \phi(\mathbf{x}^i), \phi(\mathbf{x}^j) >.$$

Kernel-based linear soft SVR

- Knowing an admissible kernel (Mercer's condition)

$K = (k(x, x'))$ with $k(x, x') = \phi(x)^T \phi(x')$ rather than the feature mapping $\phi(x)$ explicitly, we have a **kernel-based LSSVR for nonlinear regression**:

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) k(\mathbf{x}^i, \mathbf{x}^j) (\alpha_j - \alpha_j^*) \\ & - \varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad \text{(DKLSSVR)} \\ & 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C, i = 1, \dots, N \end{aligned}$$

DLSSVR vs. DKLSSVR

- Same structure, same complexity:

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) k(\mathbf{x}^i, \mathbf{x}^j) (\alpha_j - \alpha_j^*) \\ & -\varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad (\text{DKLSSVR}) \\ & 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C, i = 1, \dots, N \end{aligned}$$

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) \langle \mathbf{x}^i, \mathbf{x}^j \rangle (\alpha_j - \alpha_j^*) \\ & -\varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad (\text{DLSSVR}) \\ & 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C, i = 1, \dots, N \end{aligned}$$

Support vector expansion of KLSSVR

- For KLSSVR

$$\mathbf{w} = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \phi(\mathbf{x}^i)$$

$$b = \begin{cases} \varepsilon - y_i + \mathbf{w}^T \mathbf{x}^i, & \text{if } \alpha_i \in (0, C) \\ -\varepsilon - y_i + \mathbf{w}^T \mathbf{x}^i, & \text{if } \alpha_i^* \in (0, C) \end{cases}$$

KLSSVR Prediction:

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \phi(\mathbf{x}^i)^T \phi(\mathbf{x}) + b$$

or

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) k(\mathbf{x}^i, \mathbf{x}) + b$$