Lecture 6

Support Vector Machines

Outline

- Margin and Support Vector
- Dual Problem
- Soft Margin and Regularization
- Kernel Function
- Support Vector Regression
- Kernel Methods

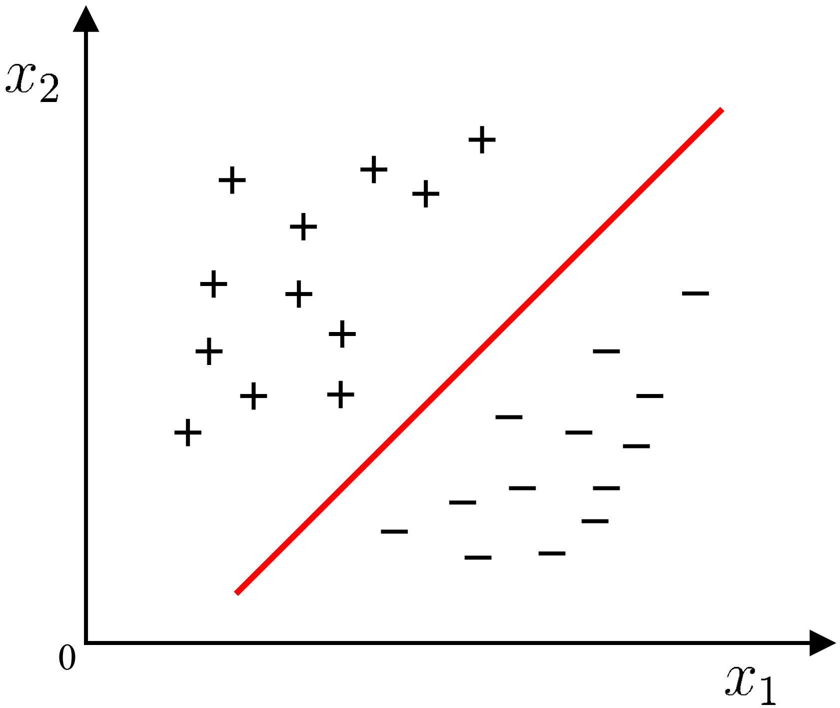
Support Vector Machine

- Vladimir Vapnik
  - Born in the Soviet Union
    - PhD in statistics, 1964
    - Co-invented the VC dimension
      - Vapnik-Chervonenkis Theory, 1974
  - Moved to the U.S. in 1990
    - Jointed AT&T
    - Developed SVM algorithm in the 90's

Vladimir N. Vapnik
1936-Present

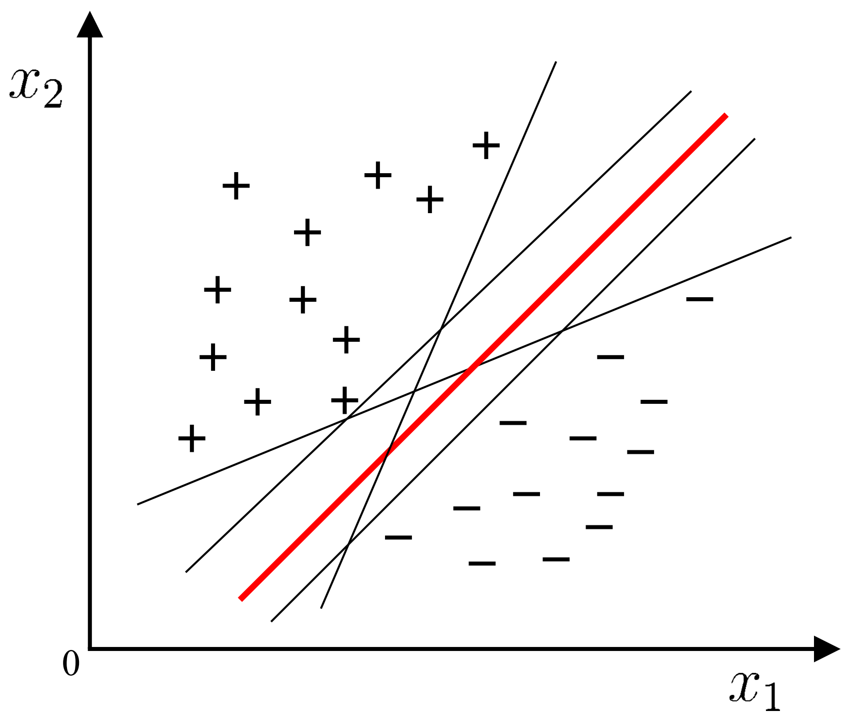
Introduction

Linear model: find a separating hyperplane in the sample space that can separate samples of different classes.



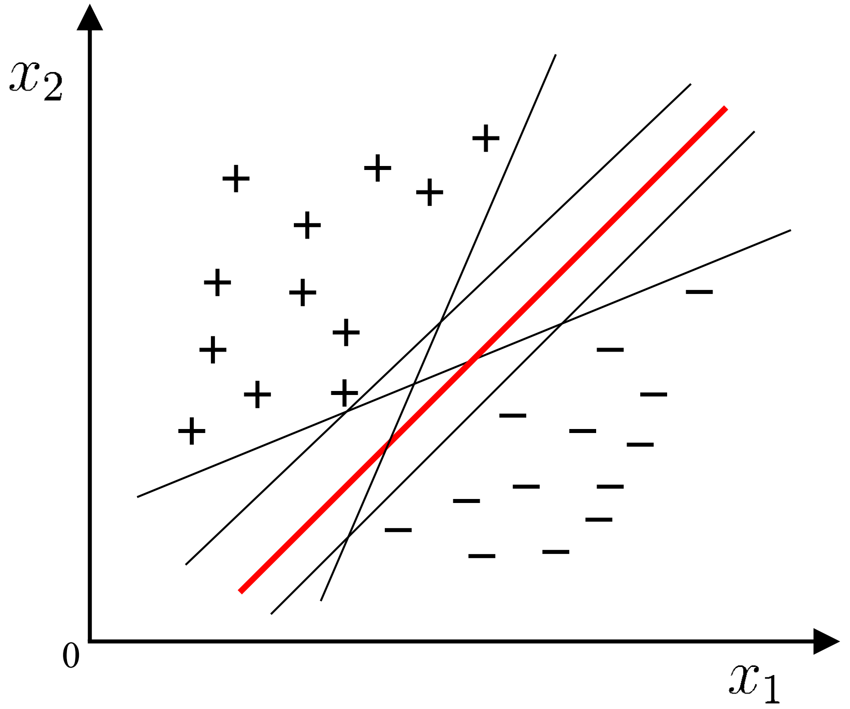
Introduction

-Q: There could be multiple qualified separating hyperplanes, which one should be chosen?



Introduction

-Q: There could be multiple qualified separating hyperplanes, which one should be chosen?

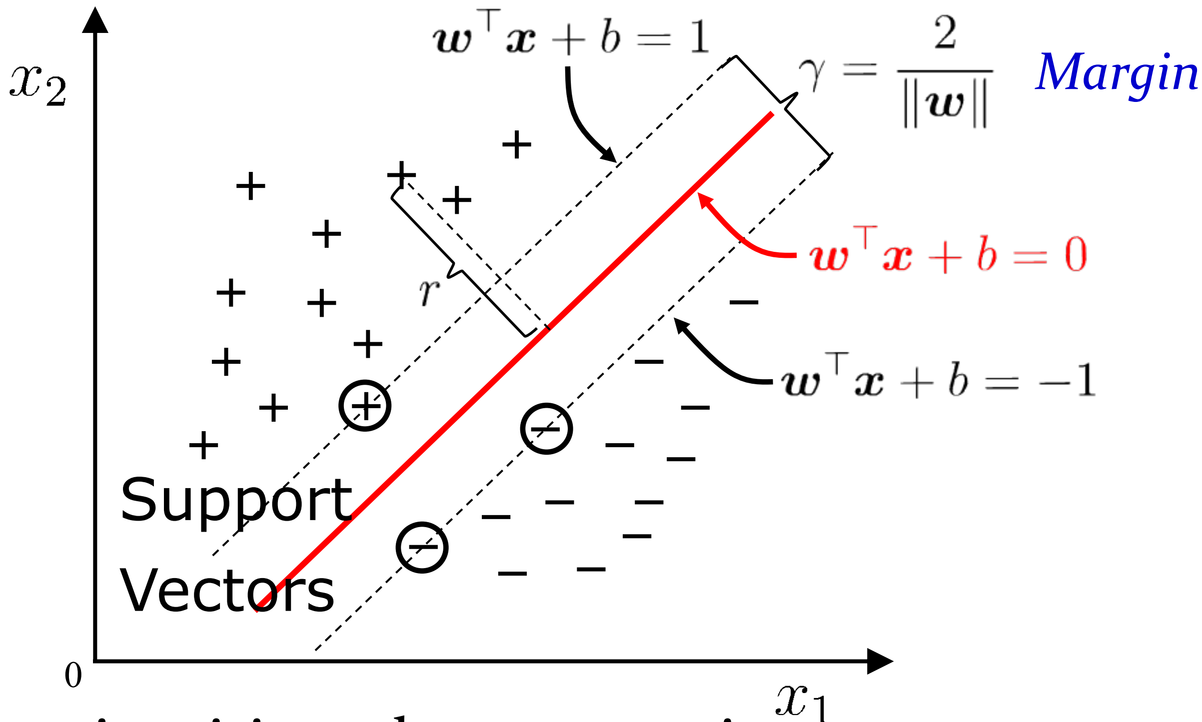


-A: The one right in the middle of two classes. It has the best tolerance to local data perturbation, the strongest generalization ability and the most robust classification results.

Margin and Support Vector

SVMs (Vapnik, 1990's) choose the linear separator with the largest margin.

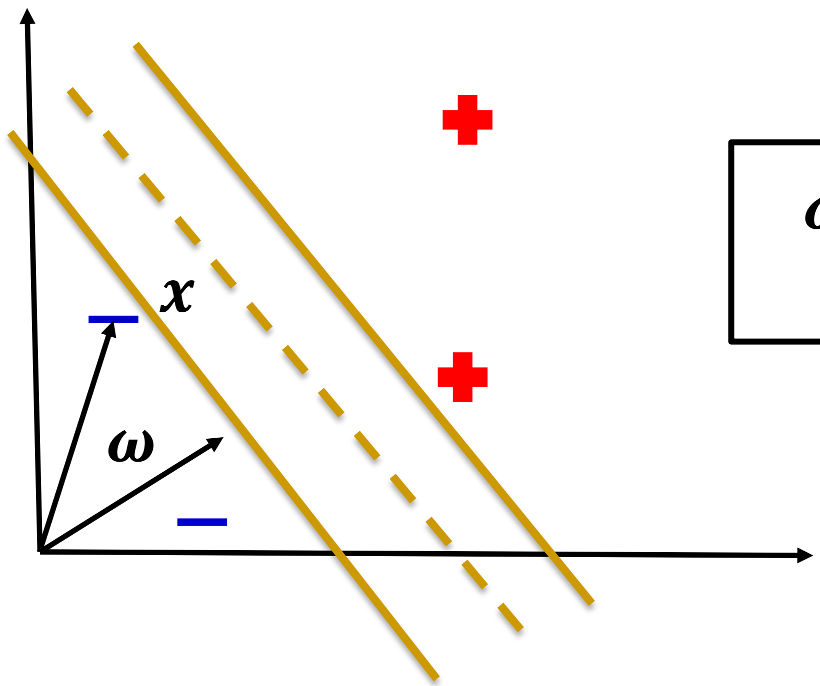
Robust to outliers!



Good according to intuition, theory, practice.

SVM became famous when, using images as input, it gave accuracy comparable to neural-network with hand-designed features in a handwriting recognition task

Decision Boundary - SVM

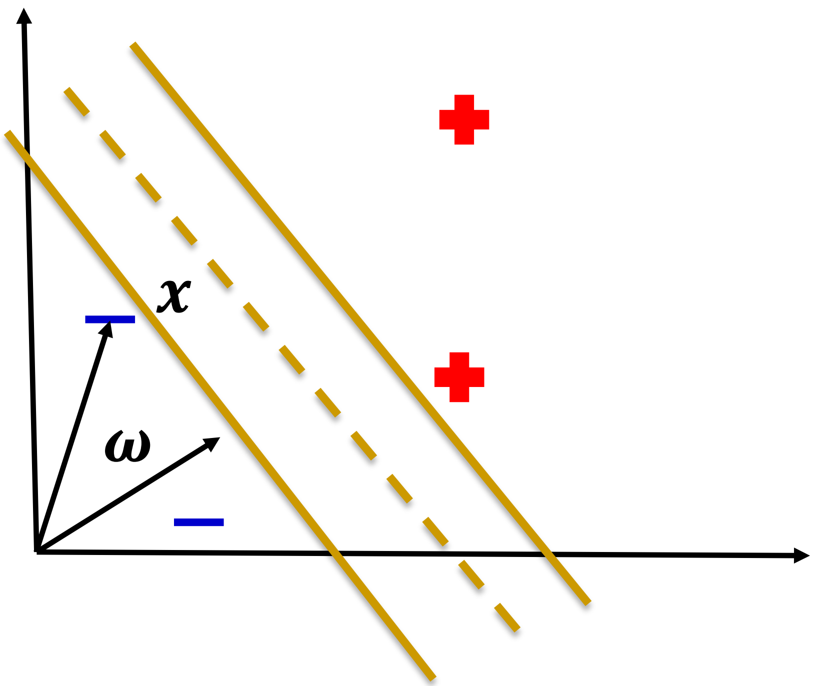


$\boldsymbol{\omega} \cdot \boldsymbol{x} \geq \boldsymbol{c}$ $\quad \boldsymbol{c} = -\boldsymbol{b}$

$\boldsymbol{\omega} \cdot \boldsymbol{x} + \boldsymbol{b} \geq 0$, *then class* +

Decision Rule

Decision Boundary - SVM



$$\boldsymbol{\omega} \cdot \boldsymbol{x} \geq \boldsymbol{c} \qquad \boldsymbol{c} = -\boldsymbol{b}$$

$$\boldsymbol{\omega} \cdot \boldsymbol{x} + \boldsymbol{b} \geq 0, \textit{ then class} +$$

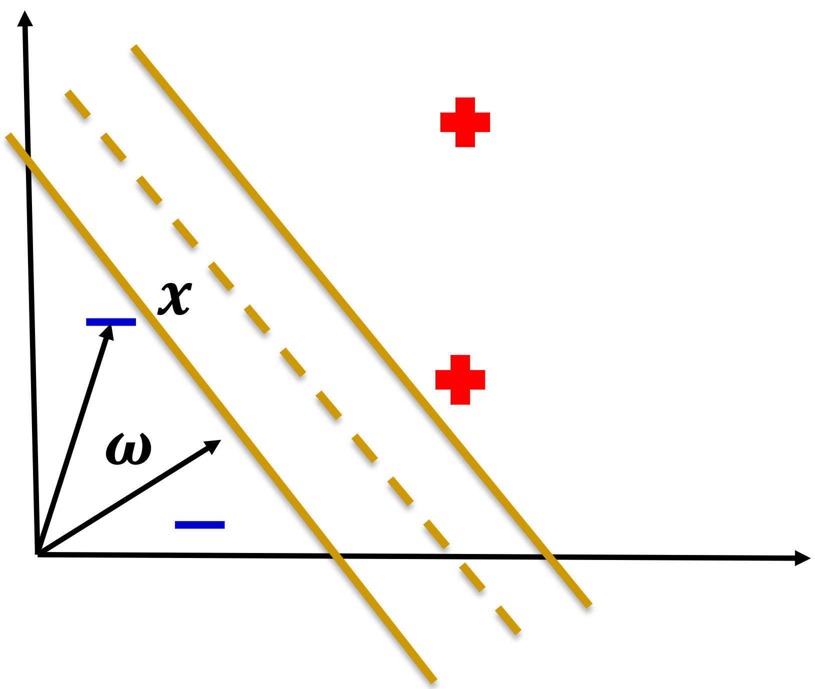
$$\boldsymbol{\omega} \cdot \boldsymbol{x}_{+} + \boldsymbol{b} \geq 1, \quad \textit{then class} +$$

$$\boldsymbol{\omega} \cdot \boldsymbol{x}_{-} + \boldsymbol{b} \leq -1, \textit{ then class} -$$

y_i such that: $y_i = +1$ for class +

$y_i = -1$ for class -

Decision Boundary - SVM

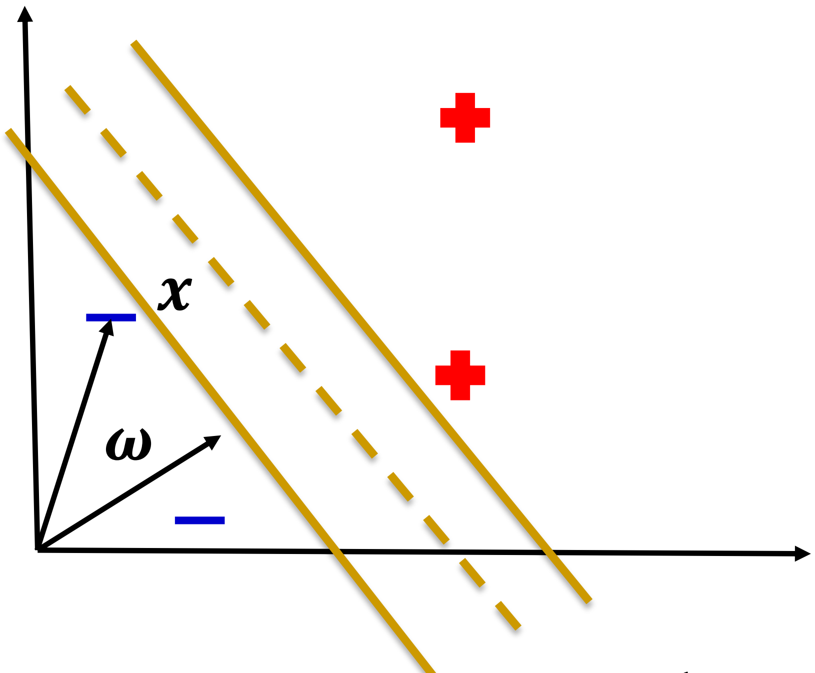


$$\boldsymbol{\omega} \cdot \boldsymbol{x}_+ + \boldsymbol{b} \geq 1, \quad then\ class\ +$$
$$\boldsymbol{\omega} \cdot \boldsymbol{x}_- + \boldsymbol{b} \leq -1, then\ class\ -$$

y_i such that: $y_i = +1$ for class +
$y_i = -1$ for class -

$$y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + \boldsymbol{b}) \geq 1, \quad for\ class\ +$$
$$y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + \boldsymbol{b}) \geq 1, \quad for\ class\ -$$

Decision Boundary - SVM



$$\boldsymbol{\omega} \cdot \boldsymbol{x}_+ + \boldsymbol{b} \geq 1, \quad \textit{then class} +$$
$$\boldsymbol{\omega} \cdot \boldsymbol{x}_- + \boldsymbol{b} \leq -1, \textit{then class} -$$

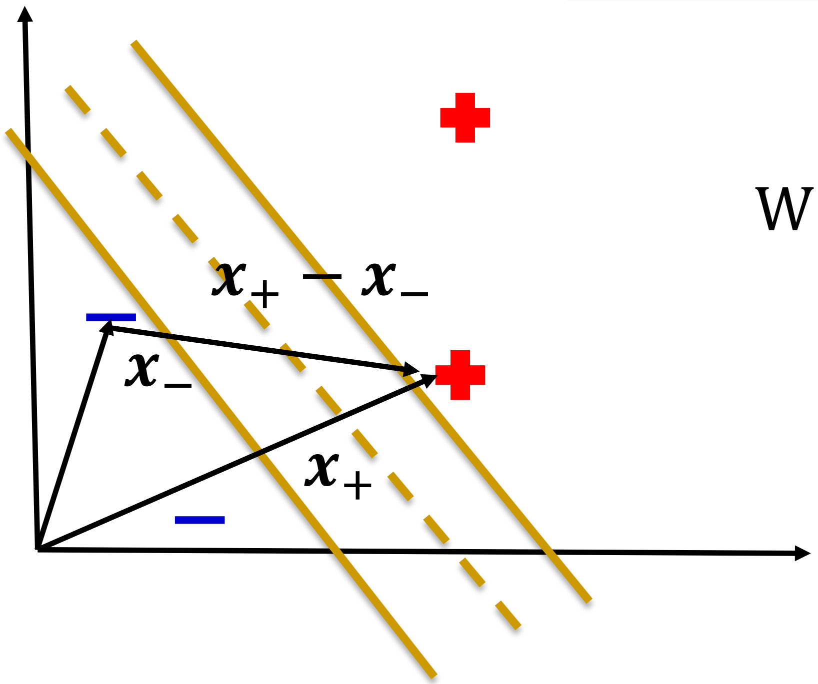
y_i such that: $y_i = +1$ for class +
$y_i = -1$ for class -

$$y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + \boldsymbol{b}) \geq 1, \quad \textit{for class} +$$
$$y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + \boldsymbol{b}) \geq 1, \quad \textit{for class} -$$

$$y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + \boldsymbol{b}) - 1 \geq 0,$$
$$y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + \boldsymbol{b}) - 1 = 0, \textit{for boundary cases}$$

Decision Boundary - SVM

$$\boxed{y_i(\boldsymbol{\omega}\cdot\boldsymbol{x}_i+\boldsymbol{b})-1=0,\ \mathit{for\ support\ vectors}}$$



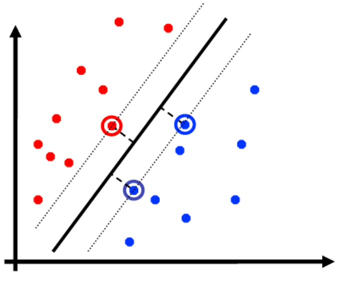
$$Width=(\boldsymbol{x}_+-\boldsymbol{x}_-)\cdot\frac{\boldsymbol{\omega}}{||\boldsymbol{\omega}||}=\frac{2}{||\boldsymbol{\omega}||}$$

$1-\boldsymbol{b}$ $\quad$ $1+\boldsymbol{b}$

$$\max\frac{2}{||\boldsymbol{\omega}||}\Leftrightarrow\max\frac{1}{||\boldsymbol{\omega}||}\Leftrightarrow min||\boldsymbol{\omega}||\Leftrightarrow min\frac{1}{2}||\boldsymbol{\omega}||^2$$

Support Vector Machines: 3 key ideas

- Use **optimization** to find solution (i.e. a hyperplane) with few errors
- Seek **large margin** separator to improve generalization
- Use **kernel trick** to make large feature spaces computationally efficient



The Primal Form of SVM

Maximum margin: finding the parameters $\boldsymbol{w}$ and b that maximize

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1, \quad \text{if } y_i = +1;$$
$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1, \quad \text{if } y_i = -1$$

$$\arg\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|}$$
$$\text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \; i = 1, 2, \ldots, m.$$

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \; i = 1, 2, \ldots, m.$$

This is an optimization problem with linear, inequality constraints.

Review of multivariable calculus

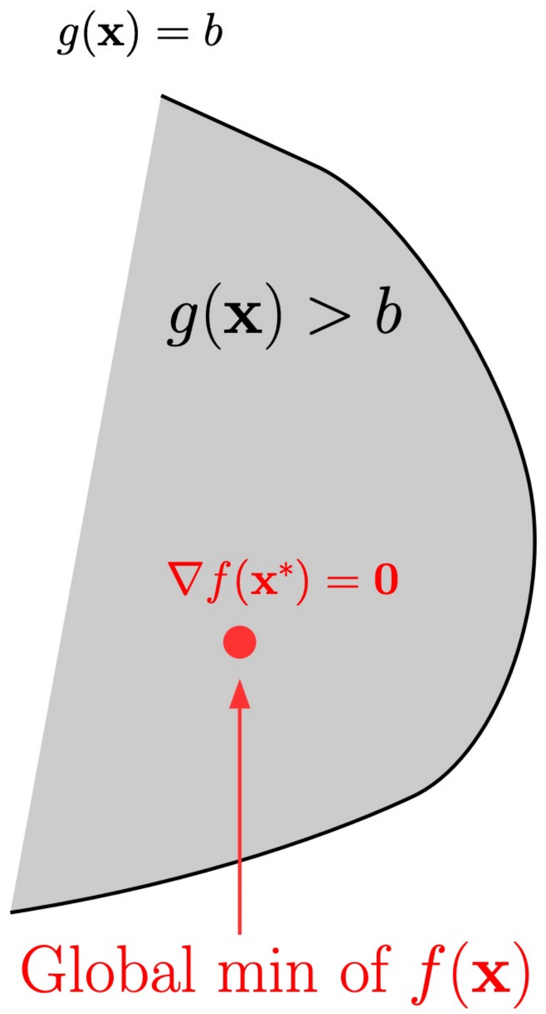
Consider the following constrained optimization problem

$$\min f(\mathbf{x}) \qquad \text{subject to} \quad g(\mathbf{x}) \geq b$$

There are two cases regarding where the global minimum of f(x) is attained:

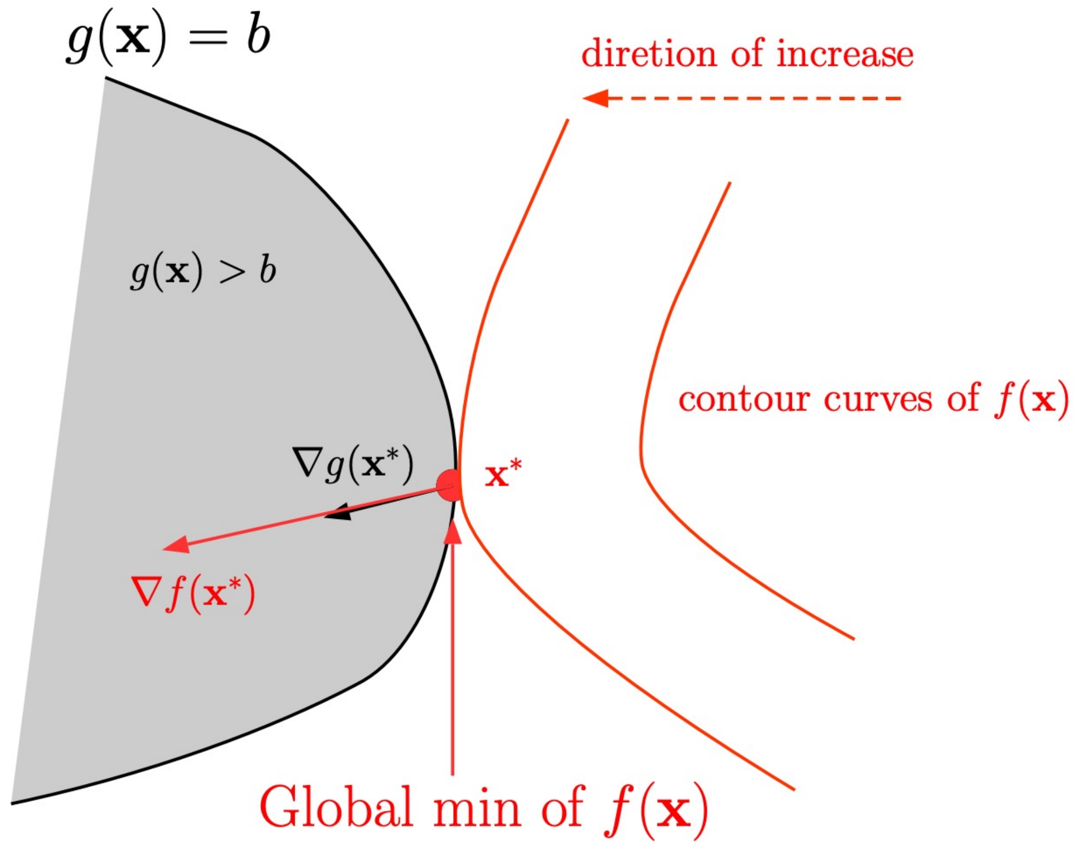
(1) At an interior point x^* $(i.e., g(x^*) > b)$. In this case x^* is just a critical point of $f(x)$.

The Lagrange Method



The Lagrange Method

(2) At a boundary point x^* (i.e., $g(x^*) = b$). In this case, there exists a constant $\lambda > 0$ such that $\nabla f(x^*) = \lambda \cdot \nabla g(x^*)$.



Why?

The Lagrange Method

The above two cases are unified by the **method of Lagrange multipliers**:

- Form the Lagrange function

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda(g(\mathbf{x}) - b)$$

- Find all critical points by solving

$$\begin{aligned} \nabla_{\mathbf{x}} L = \mathbf{0}: \quad \nabla f(\mathbf{x}) &= \lambda \nabla g(\mathbf{x}) \\ \lambda(g(\mathbf{x}) - b) &= 0 \\ \lambda &\geq 0 \\ g(\mathbf{x}) &\geq b \end{aligned}$$

Remark. The solutions give all candidate points for the global minimizer (one needs to compare them and pick the best one).

The Lagrange Method

Remarks:

• The above equations are called Karush-Kuhn-Tucker (**KKT**) conditions.

• When there are multiple inequality constraints

$$\min f(\mathbf{x}) \qquad \text{subject to} \quad g_1(\mathbf{x}) \geq b_1, \ldots, g_k(\mathbf{x}) \geq b_k$$

the method works very similarly:

The Lagrange Method

– Form the Lagrange function

$$L(\mathbf{x}, \lambda_1, \ldots, \lambda_k) = f(\mathbf{x}) - \lambda_1(g_1(\mathbf{x}) - b_1) - \cdots - \lambda_k(g_k(\mathbf{x}) - b_k)$$

– Find all critical points by solving

$$\begin{aligned} \nabla_{\mathbf{x}} L = \mathbf{0}: \quad \frac{\partial L}{\partial x_1} = 0, \ldots, \frac{\partial L}{\partial x_n} = 0 \\ \lambda_1(g_1(\mathbf{x}) - b_1) = 0, \ldots, \lambda_k(g_k(\mathbf{x}) - b_k) = 0 \\ \lambda_1 \geq 0, \ldots, \lambda_k \geq 0 \\ g_1(\mathbf{x}) \geq b_1, \ldots, g_k(\mathbf{x}) \geq b_k \end{aligned}$$

and compare them to pick the best one.

The Lagrange Method

Consider a general optimization problem (called as primal problem)

$$\begin{array}{ll} \min_x & f(x) \\ \text{subject to} & g_i(x) \geq 0, i = 1, \cdots, k \\ & h_j(x) = 0, j = 1, \cdots, m. \end{array}$$

We define its Lagrangian as

$$L(x, u, v) = f(x) - \sum_{i=1}^{k} \lambda_i g_i(x) + \sum_{j=1}^{m} u_j h(x)$$

Lagrangian multipliers $\lambda \in \mathbb{R}^k, u \in \mathbb{R}^m$.

The Lagrange Method

Lemma 1 *At each feasible* $x, f(x) = \sup_{\lambda\geq 0,\, u} L(x,\lambda,u)$, *and the supremum is taken iff* $\lambda \geq 0$ *satisfying* $\lambda_i g_i(x) = 0, i = 1, \dots, k$.

Proof: At each feasible x, we have $g_i(x) \geq 0$ and $h(x) = 0$, thus
$L(x,\lambda,u) = f(x) - \sum_{i=1}^{k} \lambda_i g_i(x) + \sum_{j=1}^{m} u_j h_j(x) \leq f(x)$.

Proposition 2 *The optimal value of primal problem, named as* f^*, *satisfies:*

$$f^* = \inf_x \sup_{\lambda\geq 0,\, u} L(x\ ,\lambda,u)$$

Proof: First considering feasible x (marked as $x \in C$), we have
$f^* = \inf_{x\in C} f(x) = \inf_x \sup_{\lambda\geq 0,\, u} L(x\ ,\lambda,u)$. Second considering non-feasible x, since $\sup_{\lambda\geq 0,\, u} L(x,\lambda,u) = \propto$ for any $x \notin C$, $\inf_{x\notin C} \sup_{\lambda\geq 0,\, u} L(x\ ,\lambda,u) = \propto$. In total,
$f^* = \inf_x \sup_{\lambda\geq 0,\, u} L(x\ ,\lambda,u)$.

The Dual Problem

A re-written Primal Problem :

$$\min_{x} \max_{\lambda \geq 0, u} L(x, \lambda, u)$$

The Dual Problem:

$$\max_{\lambda \geq 0, u} \min_{x} L(x, \lambda, u)$$

Although the primal problem is not required to be convex, the dual problem is always convex.

Theorem (weak duality):

$$d^* = \max_{\lambda \geq 0, u} \min_{x} L(x, \lambda, u) \leq \min_{x} \max_{\lambda \geq 0, u} L(x, \lambda, u) = p^*$$

Theorem (strong duality, *e.g.*, ***Slater's condition*****):**
If the primal is a convex problem, and there exists at least one strictly feasible $\tilde{x}$, meaning that $\exists \tilde{x}, g_i(\tilde{x}) > 0, i = 1, \ldots, k,\ h_j(\tilde{x}) = 0, j = 1, \ldots, m.$

$$d^* = p^*$$

Karush–Kuhn–Tucker (KKT) conditions

Necessary conditions

If x^* and λ^*, u^* are the primal and dual solutions respectively with zero duality gap, we will show that x^*, λ^*, u^* satisfy the KKT conditions.

$$
\begin{aligned}
f(x^*) &= d(\lambda^*, u^*) \text{ by zero duality gap assumption} \\
&= \min_x f(x) - \sum_{i=1}^{k} \lambda_i^* g_i(x) + \sum_{j=1}^{m} u_j^* h_j(x), \text{ by definition} \\
&\le f(x^*) - \sum_{i=1}^{k} \lambda_i^* g_i(x^*) + \sum_{j=1}^{m} u_j^* h_j(x^*) \\
&\le f(x^*)
\end{aligned}
$$

stationarity

$\Rightarrow$ equality: x^* minimizes $\mathrm{L}(x, \lambda^*, u^*)$

$\Rightarrow$ equality: $\lambda_i^* \, g_i(x^*) = 0$

complementary slackness

For convex problems with strong duality (e.g., when Slater's condition is satisfied), the KKT conditions are necessary and sufficient optimality conditions, i.e., x^* and (λ^*, u^*) are primal and dual optimal if and only if the KKT conditions hold.

Outline

- Margin and Support Vector
- Dual Problem
- Soft Margin and Regularization
- Kernel Function
- Support Vector Regression
- Kernel Methods

The Primal Form of SVM

Maximum margin: finding the parameters $\boldsymbol{w}$ and b that maximize

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i + b \geq 1, &\quad \text{if } y_i = +1; \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1, &\quad \text{if } y_i = -1 \end{aligned}$$

$$\Updownarrow$$

$$\begin{aligned} &\arg\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|} \\ &\quad \text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i = 1, 2, \ldots, m. \end{aligned}$$

$$\Downarrow$$

$$\begin{aligned} &\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\quad \text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i = 1, 2, \ldots, m. \end{aligned}$$

This is an optimization problem with linear, inequality constraints.

Dual problem

- Lagrange multipliers
  - Step-1: introducing a Lagrange multiplier $\alpha_i \geq 0$, gives the Lagrange function

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1\right)$$

  - Step-2: Setting the partial derivatives of $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ with respect to $\boldsymbol{w}$ and b to 0 gives

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i, \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

  - Step-3: Substituting back

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j - \sum_{i=1}^{m} \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0, \ \alpha_i \geq 0, \ i = 1, 2, \ldots, m. \end{aligned}$$

Sparsity of the solution

- desired model: $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \boxed{\sum_{i=1}^m \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x}} + b$

- KKT conditions: $\boldsymbol{w} = \sum_{i=1}^m \alpha_i y_i \boldsymbol{x}_i, \quad \sum_{i=1}^m \alpha_i y_i = 0.$ stationarity

$$\begin{cases} \alpha_i \geq 0, & \text{dual constraints} \\ y_i f(\boldsymbol{x}_i) \geq 1, & \text{primal constraints} \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0. & \text{complementary slackness} \end{cases}$$

$y_i f(\boldsymbol{x}_i) > 1 \Rightarrow \alpha_i = 0$

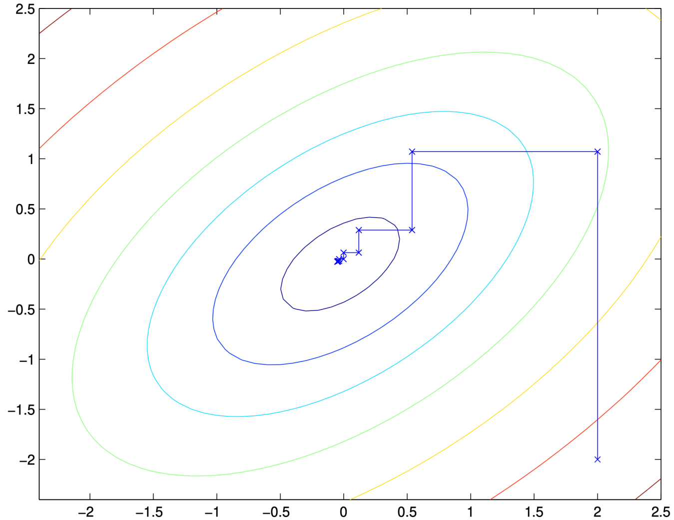
Sparsity of the solution of SVM: once the training completed, most training samples are no longer needed since the final model only depends on the support vectors.

Solving QP problem- Coordinate Ascent

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle.$$

$$\text{s.t.} \quad 0 \le \alpha_i \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y^{(i)} = 0.$$



Loop until convergence: {

For $i = 1, \ldots, n$, {

$\alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_n)$.

}

}

Solving QP problem- SMO

- Basic idea： repeats the following two steps until convergence. blocked coordinate descent
  - Step1： Select two variables to be updated: α_i and α_j
  - Step2： Fix all the parameters and solve dual problem to update α_i and α_j
- If we only consider α_i and α_j, then we can rewrite the constraints in dual problem as

$$\alpha_i y_i + \alpha_j y_j = -\sum_{k \neq i,j} \alpha_k y_k, \quad \alpha_i \geq 0, \quad \alpha_j \geq 0.$$

Eliminate the variable with another and substitute back to the dual problem leads to a univariate quadratic programming problem, which has closed-form solutions. We "clip" the value of α to respect the constraints.

- Bias term b： determined by support vectors

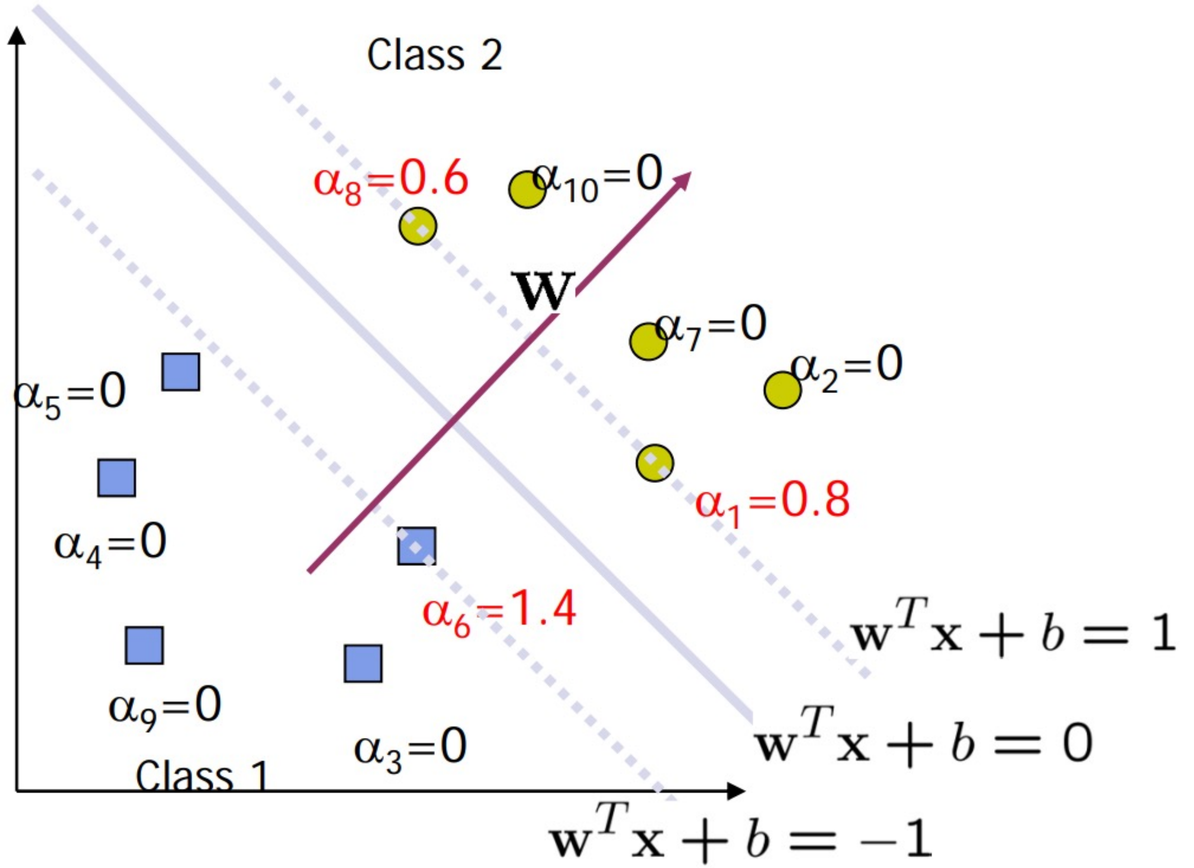
Solving QP problem- SMO

Repeat until convergence {

1. Heuristically choose a pair of α_i and α_j
2. Keeping all other α's fixed, optimize $W(\alpha)$ with respect to α_i and α_j.

}

Support vectors



The Lagrange dual problem

Primal problem

$\min\ f(\mathbf{x})\ \text{s.t.}\ g_i(\mathbf{x}) \geq b_i$

Lagrange function

$L(\mathbf{x}, \vec{\lambda}) = f(\mathbf{x}) - \sum \lambda_i (g_i(\mathbf{x}) - b_i)$

KKT Conditions

$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{0}$

$\lambda_i (g_i(\mathbf{x}) - b_i) = 0$

$\lambda_i \geq 0$

$g_i(\mathbf{x}) \geq b_i$

Lagrange dual function

$L^*(\vec{\lambda}) = \min_{\mathbf{x}} L(\mathbf{x}, \vec{\lambda})$

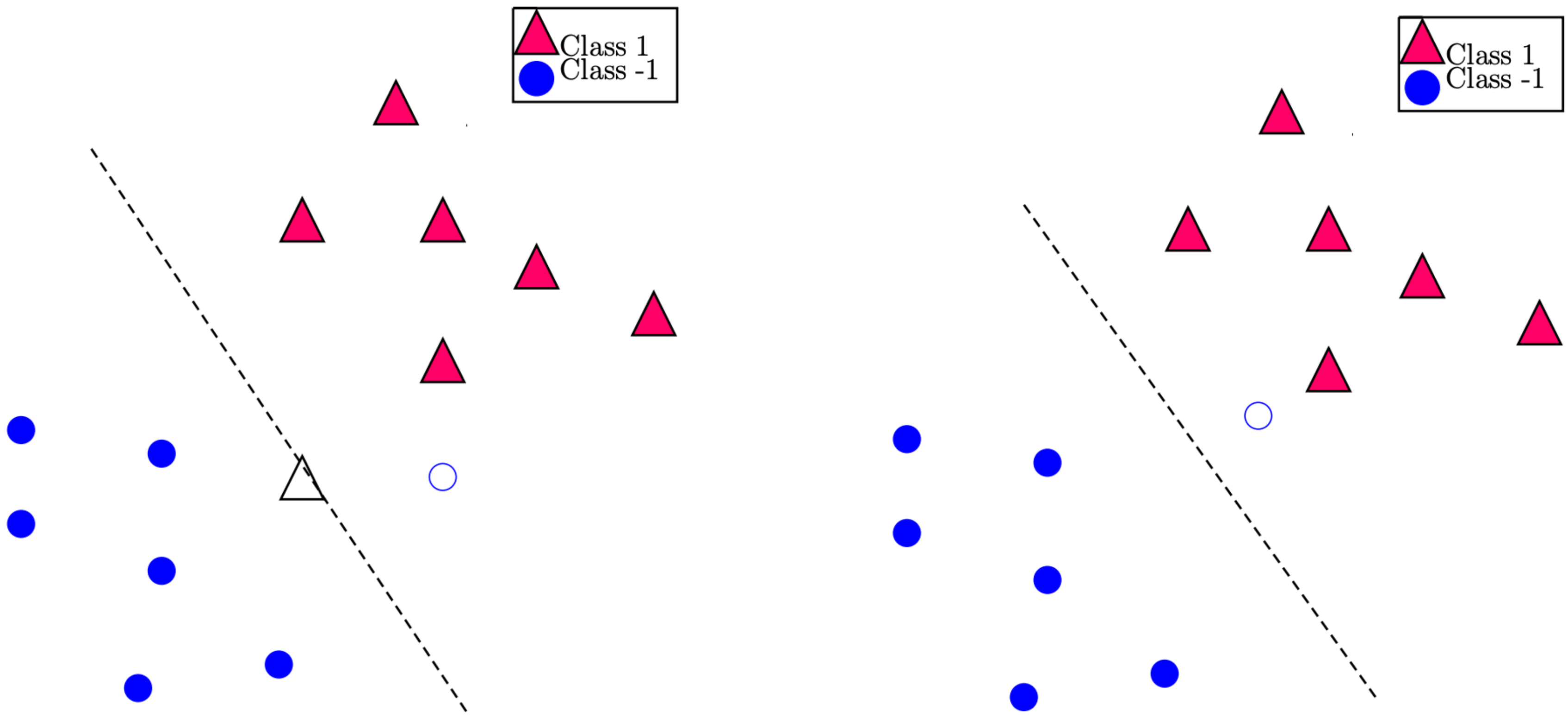
Lagrange dual problem

$\max\ L^*(\vec{\lambda})\ \text{s.t.}\ \lambda_i \geq 0$

Outline

- Margin and Support Vector
- Dual Problem
- Soft Margin and Regularization
- Kernel Function
- Support Vector Regression
- Kernel Methods

What is the optimal separating line?



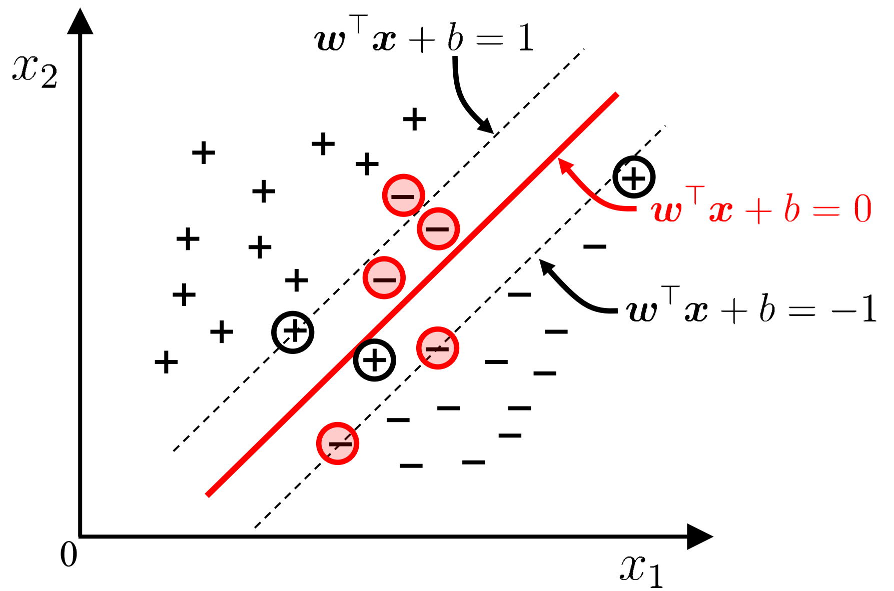
(Both data sets are much better linearly separated if several points are ignored).

Key idea #2: the slack variables

-Q: It is often difficult to find an appropriate kernel function to make the training samples linearly separable in the feature space. Even if we do find such a kernel function, it is hard to tell if it is a result of overfitting.

-A: Allow a support vector machine to make mistakes on a few samples: *soft margin.*

Instances violating the constraint

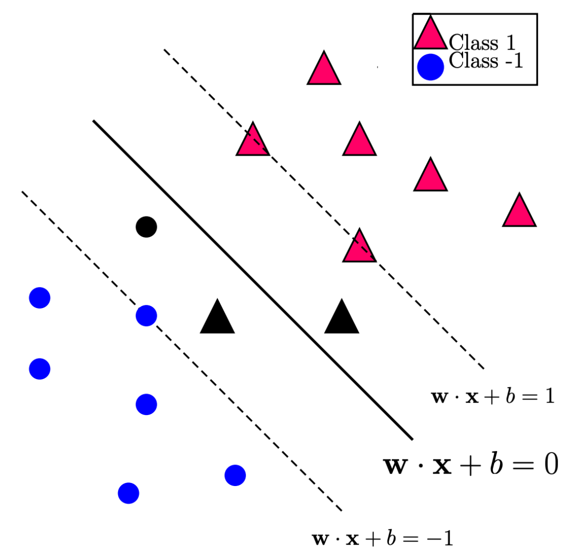


Key idea #2: the slack variables

To find a linear boundary with a large margin, we must allow violations of the constraint $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1$.

That is, we allow a few points to fall within the margin. They will satisfy

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) < 1$$



There are two cases:

- $y_i = +1$: $\boldsymbol{w} \cdot \boldsymbol{x}_i + b < 1$;
- $y_i = -1$: $\boldsymbol{w} \cdot \boldsymbol{x}_i + b > -1$.

Key idea #2: the slack variables

Formally, we introduce *slack variables* $\xi_1, \ldots, \xi_n \geq 0$ (one for each sample) to allow for exceptions:

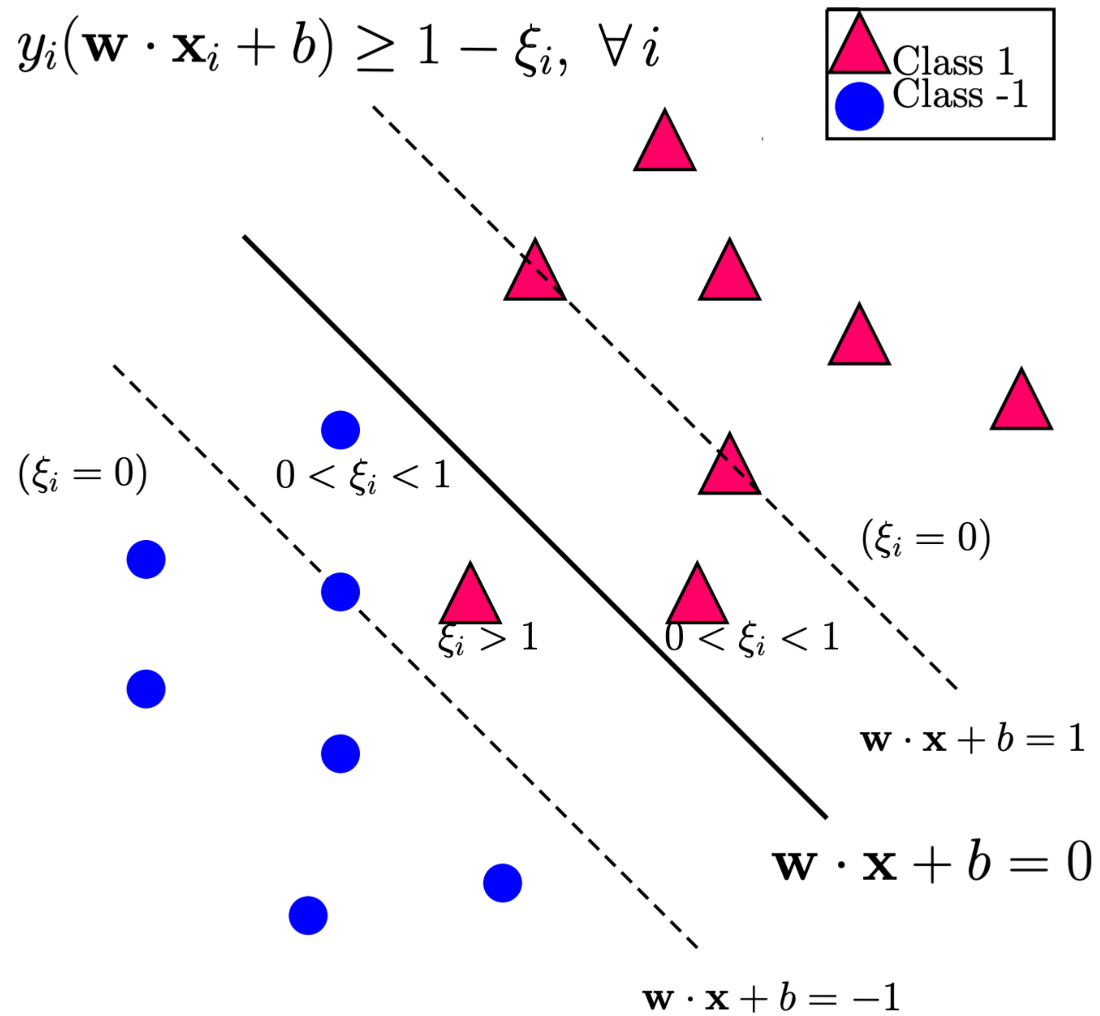
$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \xi_i, \qquad \forall i$$

where $\xi_i = 0$ for the points in ideal locations, and $\xi_i > 0$ for the violations (chosen precisely so that the equality will hold true):

- $0 < \xi_i < 1$: Still on correct side of hyperplane but within the margin
- $\xi_i > 1$: Already on wrong side of hyperplane

We say that such an SVM has a ***soft margin*** to distinguish from the previous hard margin.

Key idea #2: the slack variables



Introducing Slack Variables

Because we want most of the points to be in ideal locations, we incorporate the slack variables into the objective function as follows

$$\min_{\mathbf{w},b,\vec{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C \cdot \underbrace{\sum_i 1_{\xi_i>0}}_{\#\text{ exceptions}}$$

where $C > 0$ is a regularization constant:

• Larger C leads to fewer exceptions (smaller margin, possible overfitting).

• Smaller C tolerates more exceptions (larger margin, possible underfitting).

Clearly, there must be a tradeoff between margin and #exceptions when selecting the optimal C (often based on cross validation).

ℓ_1 relaxation of the penalty term

The discrete nature of the penalty term on previous slide, $\sum_i 1_{\xi_i > 0} = ||\vec{\xi}||_0$, makes the problem intractable.

A common strategy is to replace the ℓ_0 penalty with a ℓ_1 penalty: $\sum_i \xi_i = ||\vec{\xi}||_1$, resulting in the following full problem

$$\min_{\mathbf{w},b,\vec{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C \cdot \sum_i \xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ for all } i.$$

Remarks:
(1) Also a quadratic program with linear ineq. constraints (just more variables): $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) + \xi_i \geq 1$.

The Lagrange dual problem

The associated Lagrange function is

$$L(\mathbf{w}, b, \vec{\xi}, \vec{\lambda}, \vec{\mu}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\mu_i\xi_i$$

To find the dual problem we need to fix $\vec{\lambda}, \vec{\mu}$ and maximize over $\boldsymbol{w}, b, \vec{\xi}$:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum \lambda_i y_i \mathbf{x}_i = 0$$
$$\frac{\partial L}{\partial b} = \sum \lambda_i y_i = 0$$
$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0, \quad \forall\, i$$

The Lagrange dual problem

This yields the Lagrange dual function

$$L^*(\vec{\lambda}, \vec{\mu}) = \sum \lambda_i - \frac{1}{2} \sum \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \qquad \text{where}$$

$$\lambda_i \geq 0,\ \ \mu_i \geq 0,\ \ \lambda_i + \mu_i = C,\ \ \text{and} \ \sum \lambda_i y_i = 0.$$

The dual problem would be to maximize L^* over $\vec{\lambda}, \vec{\mu}$ subject to the constraints.

Since L^* is constant with respect to the μ_i, we can eliminate them to obtain a reduced dual problem:

$$\max_{\lambda_1,\ldots,\lambda_n} \ \sum \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$\text{subject to} \quad \underbrace{0 \leq \lambda_i \leq C}_{\text{box constraints}} \ \ \text{and} \ \sum \lambda_i y_i = 0.$$

What changed?

What about the KKT conditions?

The KKT conditions are the following

$$\mathbf{w} = \sum \lambda_i y_i \mathbf{x}_i, \quad \sum \lambda_i y_i = 0, \quad \lambda_i + \mu_i = C$$

$$\lambda_i(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) = 0, \quad \mu_i \xi_i = 0$$

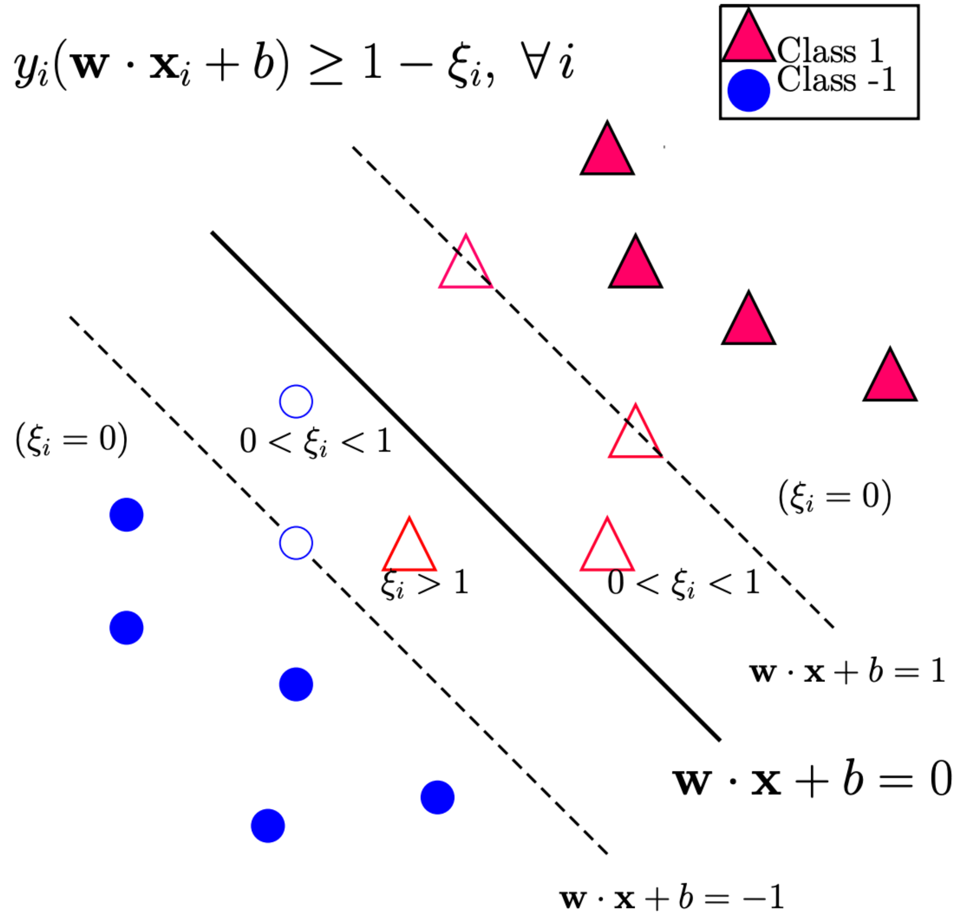
$$\lambda_i \geq 0, \quad \mu_i \geq 0$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

We see that

- The optimal **w** has the same formula: $\boldsymbol{w} = \sum \lambda_i y_i \boldsymbol{x}_i$.
- Any point with $\lambda_i > 0$ and correspondingly $y_i(\boldsymbol{w} \cdot \boldsymbol{x} + b) = 1 - \xi_i$ is a support vector (not just those on the margin boundary $\boldsymbol{w} \cdot \boldsymbol{x} + b = \pm 1$).

To find b

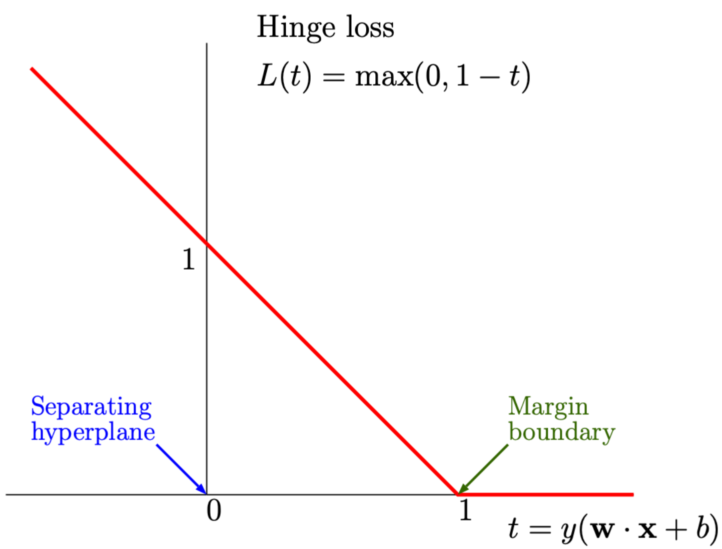


To find b, choose any support vector $\boldsymbol{x}_i$ with $0 < \lambda_i < C$ (which implies that $\mu_i > 0$ and $\xi_i = 0$), and use the formula $b = \frac{1}{y_i} - \boldsymbol{w} \cdot \boldsymbol{x}_i$.

ℓ_1 relaxation of the penalty term

The problem may be rewritten as an unconstrained optimization problem

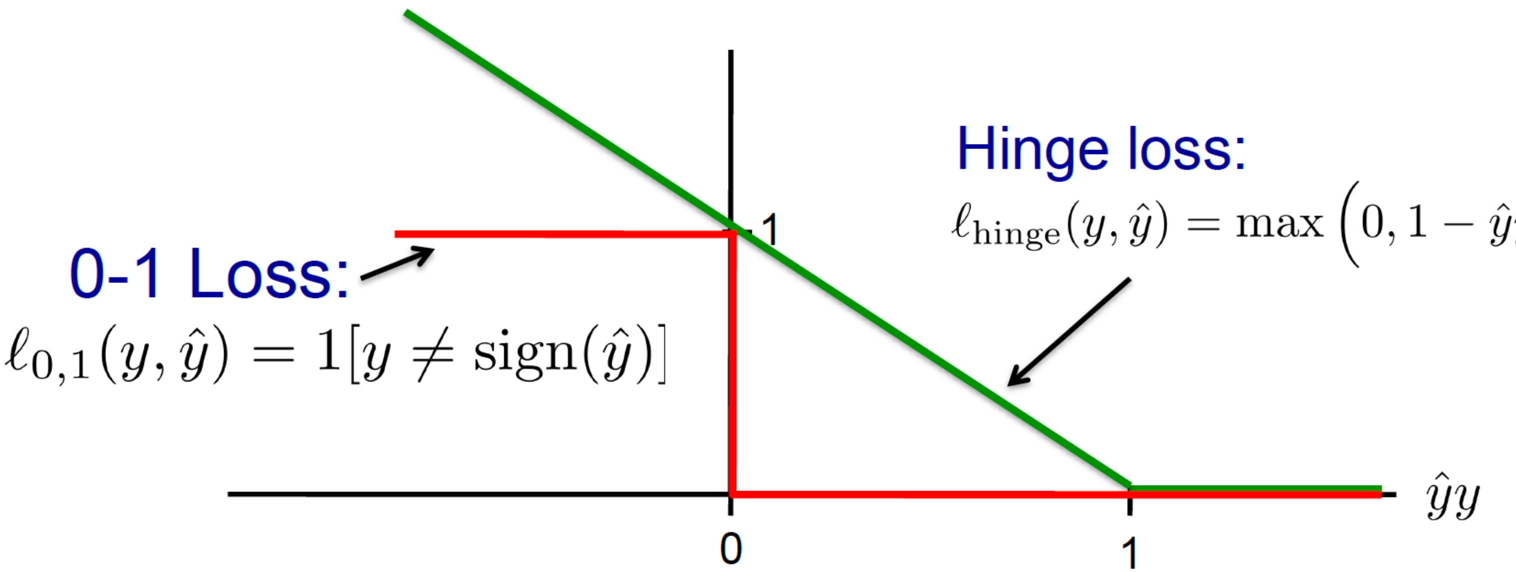
$$\min_{\mathbf{w},b} \quad \underbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}_{\text{regularization}} + C \cdot \underbrace{\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b))}_{\text{hinge loss}}$$



sub-gradient descent

Hinge loss vs. 0/1 loss

$$\zeta_i = \max\left(0, 1 - y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i - b)\right)$$

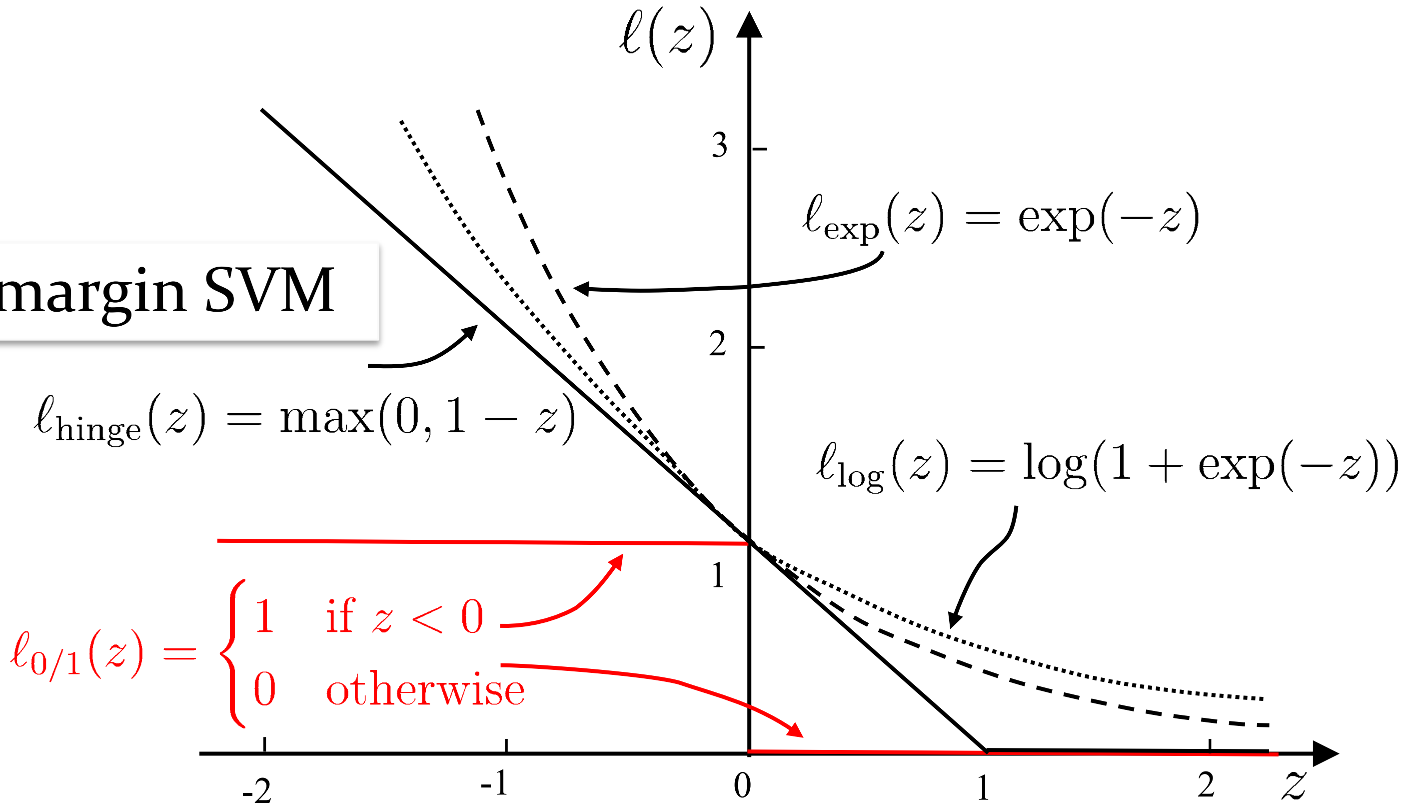


Hinge loss upper bounds 0/1 loss!

It is the tightest convex upper bound on the 0/1 loss

Surrogate loss functions

Soft margin SVM



Surrogate loss functions have nice mathematical properties, e.g., convex, continuous, and are upper bound of 0/1 loss function

Regularization

- General form of SVM models:

$$\min_f \ \Omega(f) + C \sum_{i=1}^{m} l(f(\boldsymbol{x}_i), y_i)$$

Structural risk, representing some properties of the model

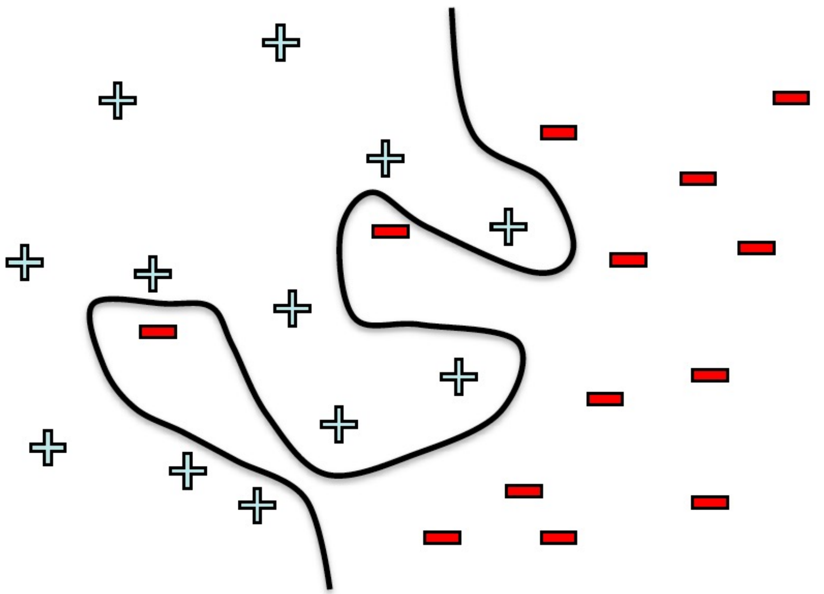
Empirical risk, describing how well the model matches the training data

- Other learning models can be derived by substituting the above components
  - Logistic Regression
  - LASSO
  - ……

Outline

- Margin and Support Vector
- Dual Problem
- Soft Margin and Regularization
- Kernel Function
- Support Vector Regression
- Kernel Methods

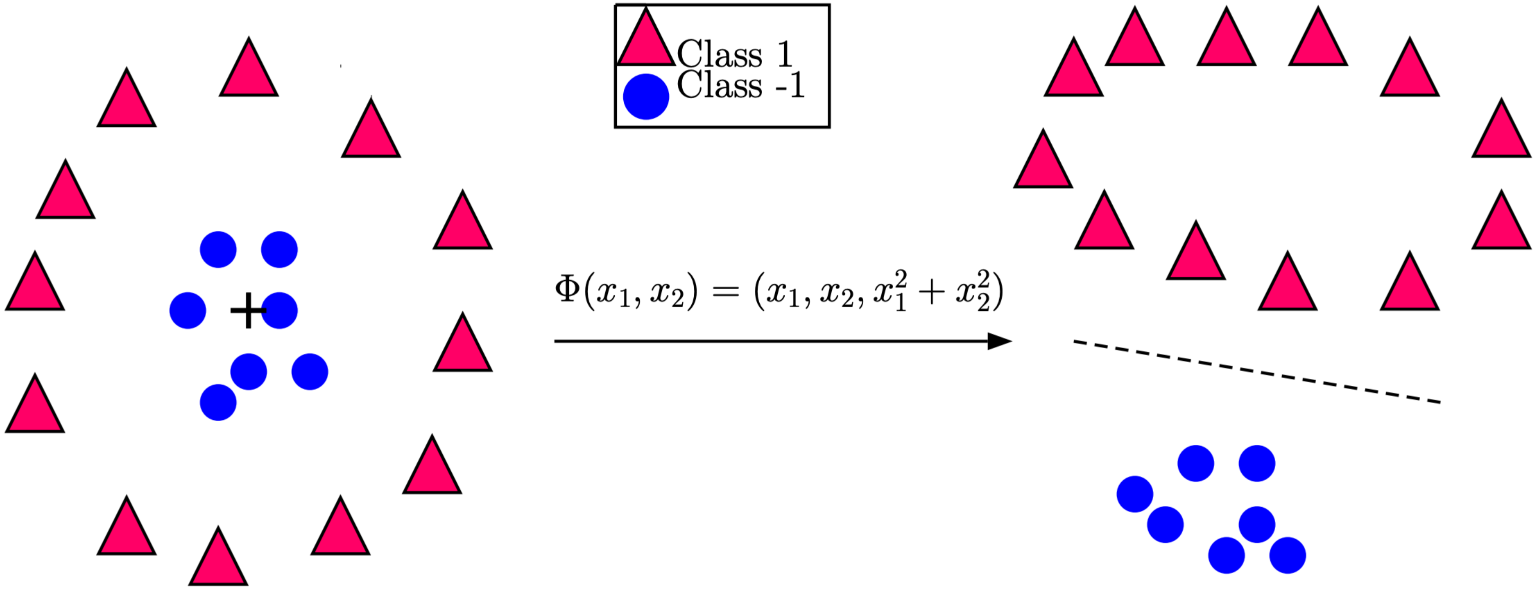
What if the data is not linearly separable?



Use features of features of features of features....

$$\phi(x) = \begin{pmatrix} x^{(1)} \\ \dots \\ x^{(n)} \\ x^{(1)}x^{(2)} \\ x^{(1)}x^{(3)} \\ \dots \\ e^{x^{(1)}} \\ \dots \end{pmatrix}$$

Use Feature Map

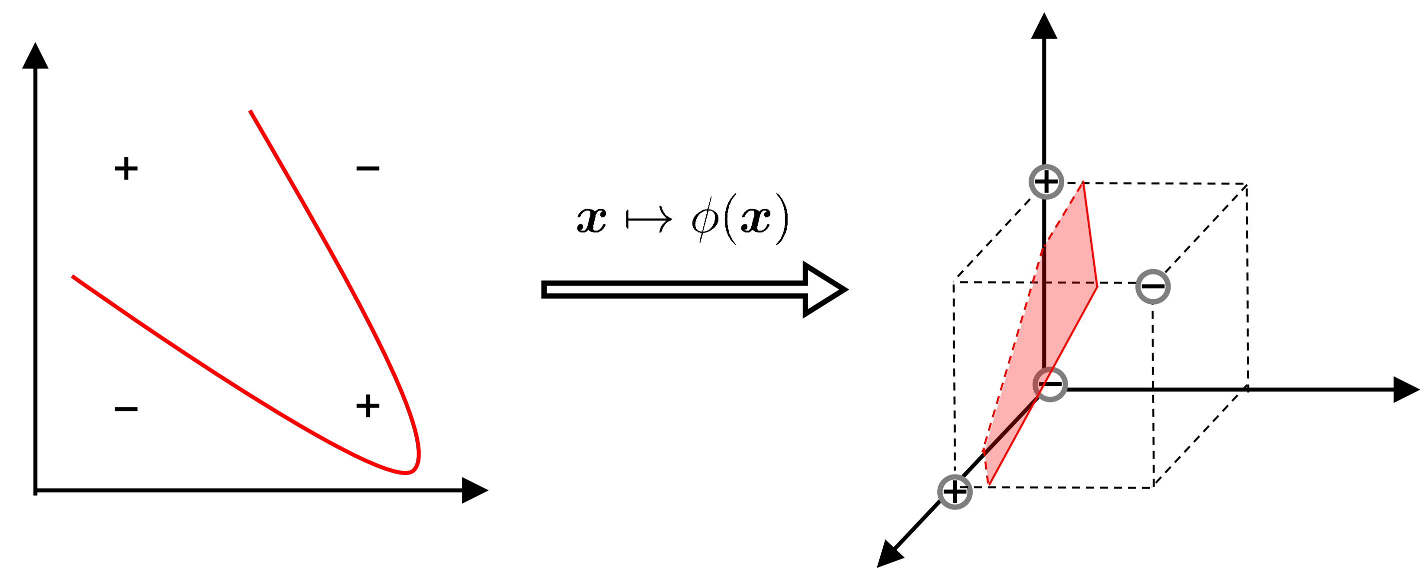


Feature space can get really large really quickly!

Key idea #3: the kernel trick

- High dimensional feature spaces at no extra cost!
- Map the samples from the original feature space to a

higher dimensional feature space. That way the samples become linearly separable.



Kernel SVM

Let $\phi(\boldsymbol{x})$ denote the mapped feature vector of $\boldsymbol{x}$, the separating hyperplane $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b$ can be expressed as

Original Problem

$$\min_{\boldsymbol{w},b} \ \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t.} \ \ y_i(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b) \geq 1, \ \ i = 1, 2, \ldots, m.$$

Dual Problem

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)} - \sum_{i=1}^{m} \alpha_i$$
$$\text{s.t.} \ \ \sum_{i=1}^{m} \alpha_i y_i = 0, \ \ \alpha_i \geq 0, \ \ i = 1, 2, \ldots, m.$$

Prediction

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x})} + b$$

Kernel function

- Basic idea： design kernel function instead of kernel mapping explicitly

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$$

- Mercer's theorem (sufficient, nonessential)： if only the corresponding kernel matrix of a symmetric function is positive-definite, it can act as a kernel function. *(analogous to the definition of a positive-semidefinite matrix)*
- Common kernel functions：

Name	Expression	Parameters
Linear kernel	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$	
Polynomial kernel	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)^d$	$d \geqslant 1$ is the degree of the polynomial.
Gaussian kernel	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\Vert\boldsymbol{x}_i - \boldsymbol{x}_j\Vert^2}{2\sigma^2}\right)$	$\sigma > 0$ is the width of the Gaussian kernel.
Laplacian kernel	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\Vert\boldsymbol{x}_i - \boldsymbol{x}_j\Vert}{\sigma}\right)$	$\sigma > 0$.
Sigmoid kernel	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j + \theta)$	tanh is the hyperbolic tangent function, $\beta > 0$, $\theta < 0$.

What are good kernel functions?

- Linear kernel
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$
- Polynomial
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^{\mathrm{n}}$
- Gaussian (also called Radial Basis Function, or RBF)
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}}$
- …

Kernel algebra

kernel composition	feature composition
a) $k(\mathbf{x},\mathbf{v}) = k_a(\mathbf{x},\mathbf{v}) + k_b(\mathbf{x},\mathbf{v})$	$\boldsymbol{\phi}(\mathbf{x}) = (\boldsymbol{\phi}_a(\mathbf{x}), \boldsymbol{\phi}_b(\mathbf{x}))$,
b) $k(\mathbf{x},\mathbf{v}) = f k_a(\mathbf{x},\mathbf{v}),\ f > 0$	$\boldsymbol{\phi}(\mathbf{x}) = \sqrt{f}\boldsymbol{\phi}_a(\mathbf{x})$
c) $k(\mathbf{x},\mathbf{v}) = k_a(\mathbf{x},\mathbf{v}) k_b(\mathbf{x},\mathbf{v})$	$\phi_m(\mathbf{x}) = \phi_{ai}(\mathbf{x})\phi_{bj}(\mathbf{x})$
d) $k(\mathbf{x},\mathbf{v}) = \mathbf{x}^T A\mathbf{v}$, A positive semi-definite	$\boldsymbol{\phi}(\mathbf{x}) = L^T\mathbf{x}$, where $A = LL^T$.
e) $k(\mathbf{x},\mathbf{v}) = f(\mathbf{x})f(\mathbf{v})k_a(\mathbf{x},\mathbf{v})$	$\phi(\mathbf{x}) = f(\mathbf{x})\phi_a(\mathbf{x})$

Quadratic kernel

$$
\begin{aligned}
k(\mathbf{x},\mathbf{z}) &= (\mathbf{x}^T\mathbf{z}+c)^2 = \left(\sum_{j=1}^{n} x^{(j)} z^{(j)} + c\right)\left(\sum_{\ell=1}^{n} x^{(\ell)} z^{(\ell)} + c\right) \\
&= \sum_{j=1}^{n}\sum_{\ell=1}^{n} x^{(j)} x^{(\ell)} z^{(j)} z^{(\ell)} + 2c\sum_{j=1}^{n} x^{(j)} z^{(j)} + c^2 \\
&= \sum_{j,\ell=1}^{n} (x^{(j)} x^{(\ell)})(z^{(j)} z^{(\ell)}) + \sum_{j=1}^{n} (\sqrt{2c}x^{(j)})(\sqrt{2c}z^{(j)}) + c^2,
\end{aligned}
$$

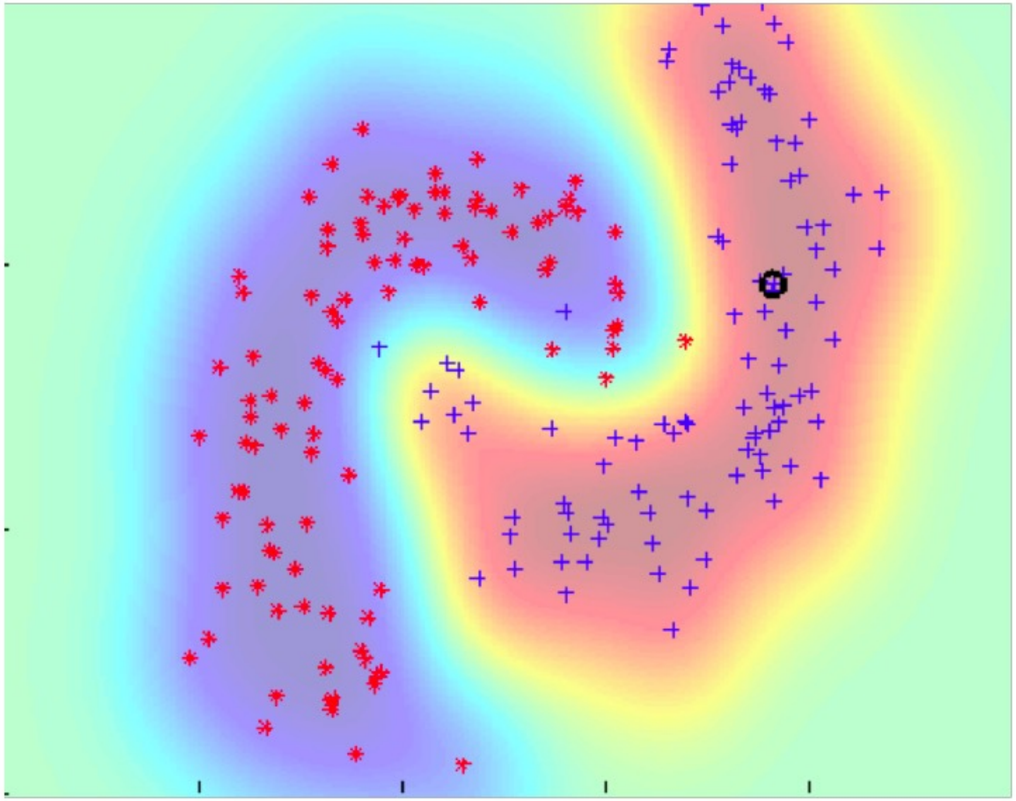
Feature mapping given by:

$$
\mathbf{\Phi}(\mathbf{x}) = [x^{(1)2}, x^{(1)}x^{(2)}, \ldots, x^{(3)2}, \sqrt{2c}x^{(1)}, \sqrt{2c}x^{(2)}, \sqrt{2c}x^{(3)}, c]
$$

Gaussian kernel (RBF)

$$K(\vec{u},\vec{v}) = \exp\left(-\frac{||\vec{u}-\vec{v}||_2^2}{2\sigma^2}\right)$$

$$y \leftarrow \text{sign}\left[\sum_i \alpha_i y_i \exp\left(-\frac{\|\vec{x}-\vec{x}_i\|_2^2}{2\sigma^2}\right) + b\right]$$



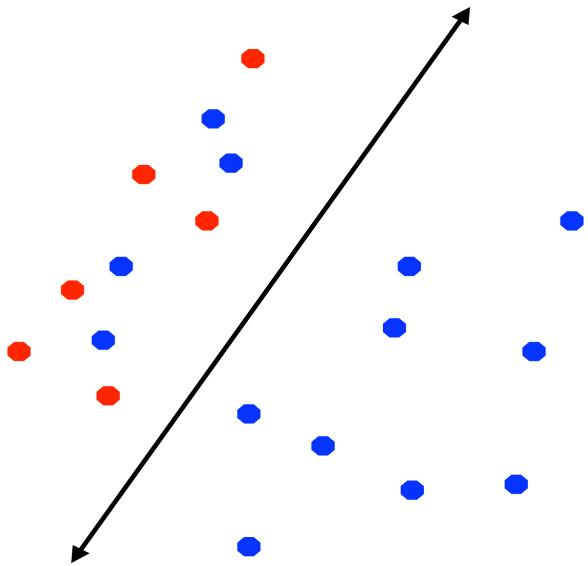
$$\psi_{\text{RBF}} : \mathbb{R}^n \rightarrow \mathbb{R}^\infty$$

Proof?

Hint:
Taylor expansion of exponential function

The feature mapping is infinite dimensional!

How to deal with imbalanced data?



- In many practical applications we may have **imbalanced** data sets
- We may want errors to be equally distributed between the positive and negative classes
- A slight modification to the SVM objective does the trick!

$$N = N_+ + N_-$$

$$\min_{w,b} \; ||w||_2^2 + \frac{CN}{2N_+} \sum_{j:y_j=+1} \xi_j + \frac{CN}{2N_-} \sum_{j:y_j=-1} \xi_j$$

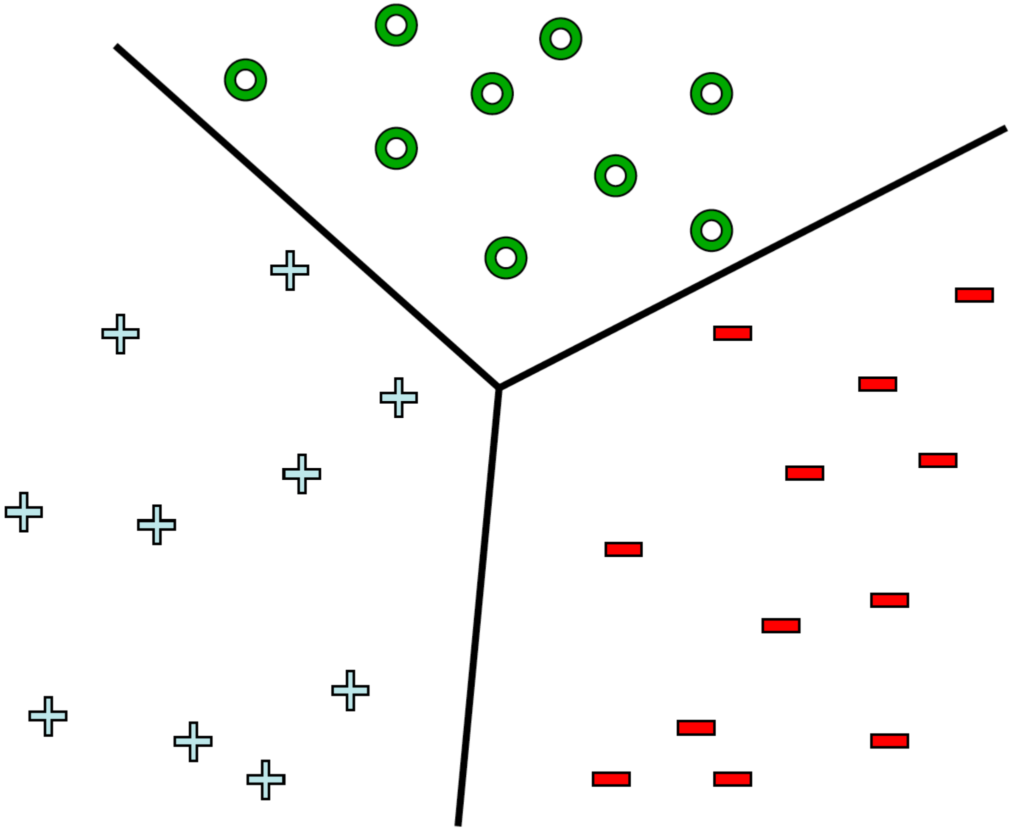
Class-specific weighting of the slack variables

Overfitting?

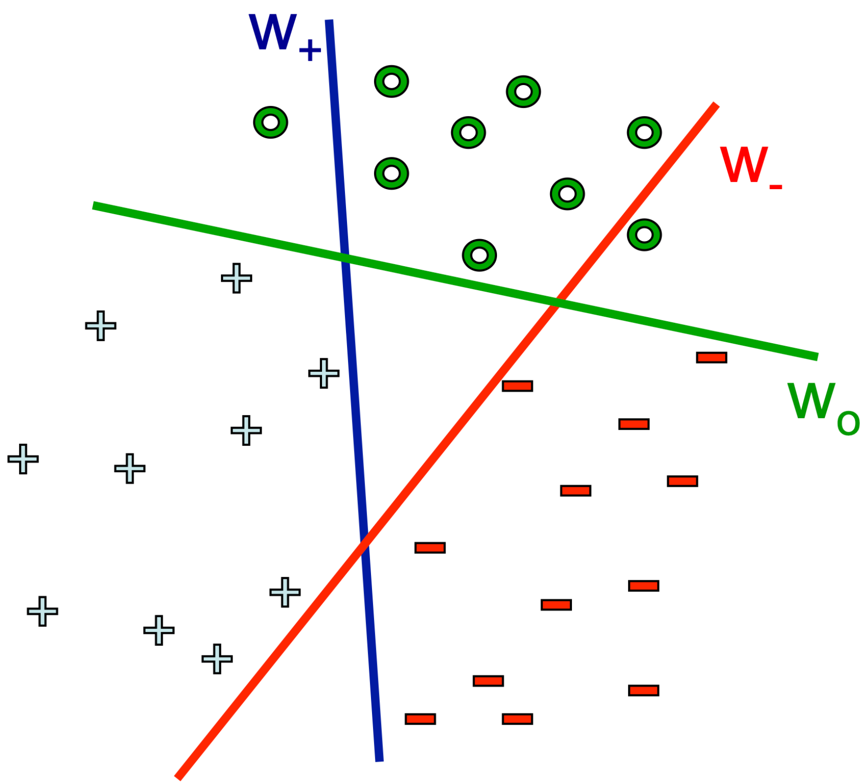
- Huge feature space with kernels: should we worry about overfitting?
  - SVM objective seeks a solution with large margin
    - Theory says that large margin leads to good generalization

      (we will see this in a couple of lectures)
  - But everything overfits sometimes!!!
  - Can control by:
    - Setting C
    - Choosing a better Kernel
    - Varying parameters of the Kernel (width of Gaussian, etc.)

How do we do multi-class classification?



One versus rest classification



Learn 3 classifiers:

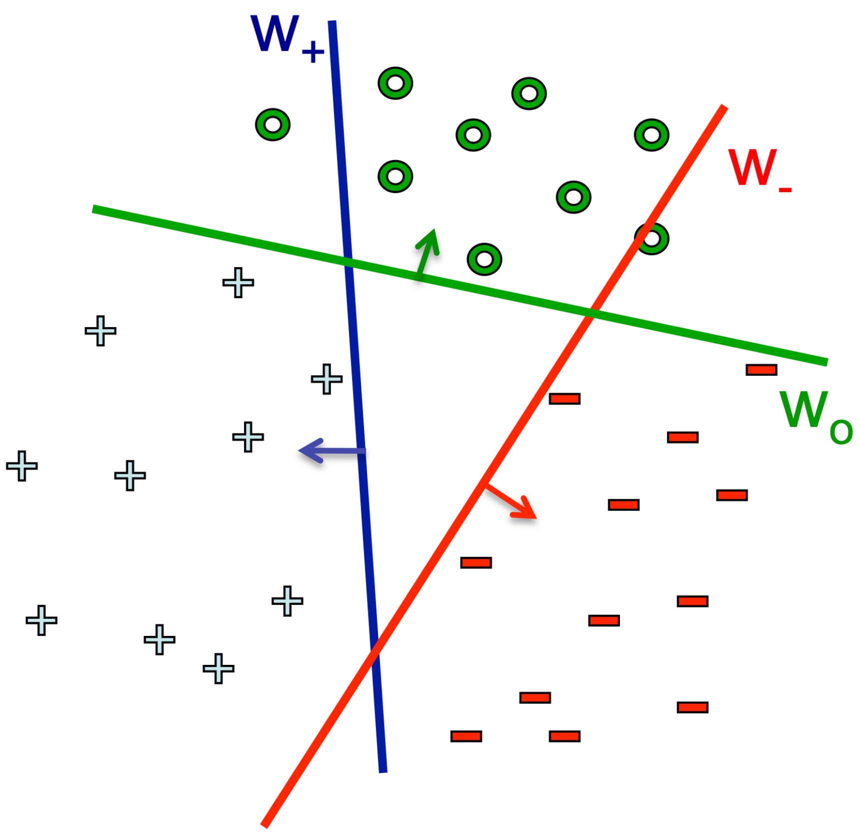
- \- vs {o,+}, weights w-
- \+ vs {o,-}, weights w+
- o vs {+,-}, weights wo

Predict label using:

$$\hat{y} \leftarrow \arg\max_k \; w_k \cdot x + b_k$$

Any problems?

Multi-class SVM



Simultaneously learn 3 sets of weights:

- How do we guarantee the correct labels?
- Need new constraints!

The "score" of the correct class must be better than the "score" of wrong classes:

$$w^{(y_j)} \cdot x_j + b^{(y_j)} > w^{(y)} \cdot x_j + b^{(y)} \qquad \forall j,\ y \neq y_j$$

Multi-class SVM

As for the SVM, we introduce slack variables and maximize margin:

$$\text{minimize}_{\mathbf{w},b} \quad \sum_y \mathbf{w}^{(y)}.\mathbf{w}^{(y)} + C\sum_j \xi_j$$
$$\mathbf{w}^{(y_j)}.\mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y')}.\mathbf{x}_j + b^{(y')} + 1 - \xi_j, \ \forall y' \neq y_j, \ \forall j$$
$$\xi_j \geq 0, \ \forall j$$

To predict, we use:

$$\hat{y} \leftarrow \arg\max_k \; w_k \cdot x + b_k$$

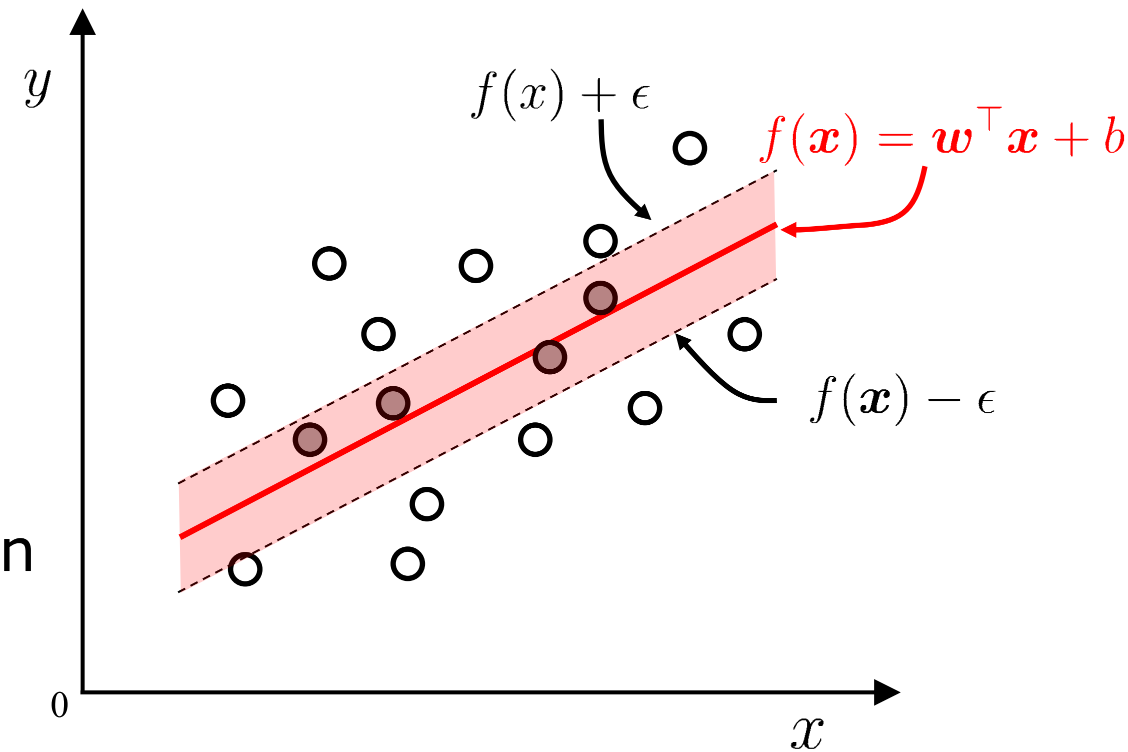
Outline

- Margin and Support Vector
- Dual Problem
- Soft Margin and Regularization
- Kernel Function
- Support Vector Regression
- Kernel Methods

Support vector regression

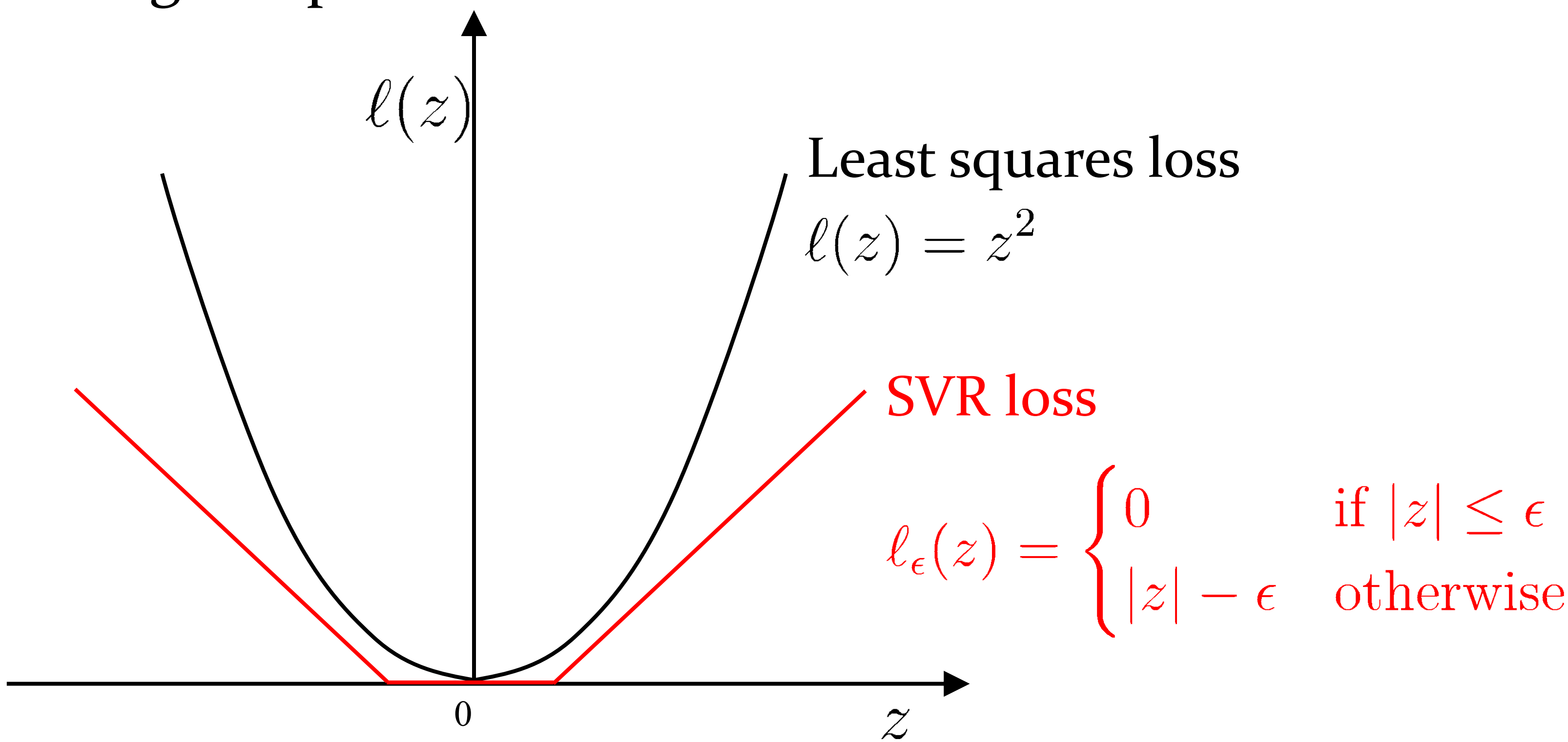
Allows an error 2ϵ between model output and ground truth

2ϵ region



Loss function

Training samples falling within 2ϵ region are considered as correctly predicted, that is, no loss. The solution of SVR is sparse since the support vectors are only a subset of the training samples.



Formulation

Original Problem

$$\begin{aligned}\min_{\boldsymbol{w},b,\xi_i,\hat{\xi}_i} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \hat{\xi}_i)\\ \text{s.t.} \quad & y_i - \boldsymbol{w}^\top\phi(\boldsymbol{x}_i) - b \leq \epsilon + \xi_i,\\ & y_i - \boldsymbol{w}^\top\phi(\boldsymbol{x}_i) - b \geq -\epsilon - \hat{\xi}_i,\\ & \xi_i \geq 0,\ \hat{\xi}_i \geq 0,\ i = 1, 2, \ldots, m.\end{aligned}$$

Dual Problem

$$\begin{aligned}\min_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{m}(\alpha_i(\epsilon - y_i) + \hat{\alpha}_i(\epsilon + y_i))\\ \text{s.t.} \quad & \sum_{i=1}^{m}(\alpha_i - \hat{\alpha}_i) = 0,\\ & 0 \leq \alpha_i \leq C,\ 0 \leq \hat{\alpha}_i \leq C.\end{aligned}$$

Prediction $f(\boldsymbol{x}) = \boldsymbol{w}^\top\phi(\boldsymbol{x}) + b = \sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i)y_i\kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b$

Outline

- Margin and Support Vector
- Dual Problem
- Soft Margin and Regularization
- Kernel Function
- Support Vector Regression
- Kernel Methods

Representer theorem

SVM $$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

SVR $$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i) y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

Conclusion: The learned models of SVM and SVR can be expressed as a linear combination of the kernel functions.

A more generalized conclusion(representer theorem): for any monotonically increasing function Ω and any non-negative loss function l, the optimization problem

$$\min_{h \in \mathbb{H}} \quad F(h) = \Omega(\|h\|_{\mathbb{H}}) + l(h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_m))$$

Solution can be written in the form of $h^* = \sum_{i=1}^{m} \alpha_i \kappa(\cdot, \boldsymbol{x}_i)$

Summary

- Unusual choice of separation strategy:
  - Maximize “street” between groups
- Attack maximization problem:
  - Lagrange multipliers + hairy mathematics
- New problem is a quadratic minimization
  - Susceptible to fancy numerical methods
- Result depends on dot products only
  - Enables use of kernel methods

Credits

- The flow of this SVM lecture goes to
  - Patrick Winston, Professor of Artificial Intelligence
  - Director of MIT Artificial Intelligence Lab (1992-1997)
  - Taught 6.034: Artificial Intelligence

https://ocw.mit.edu/courses/6-034-artificial-intelligence-fall-2010/

1943-2019

Take Home Message

- The "large margin" idea of SVM
- Dual problem and the sparsity of the solution
- Solving linear inseparable problems by projecting to high-dimensional space
- Solving linear inseparable problems in the feature space by introducing "soft margin"
- Utilizing the idea of support vectors into regression tasks and get SVR
- Extending kernel methods to other learning models

Mature SVM packages

- LIBSVM
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- LIBLINEAR
  http://www.csie.ntu.edu.tw/~cjlin/liblinear/
- SVM^{light}、SVM^{perf}、SVM^{struct}
  http://svmlight.joachims.org/svm_struct.html
- Scikit-learn
  http://scikit-learn.org/stable/modules/svm.html