Support Vector Machine (SVM)

DSA5103 Lecture 5

Yangjing Zhang

09-Feb-2023

NUS

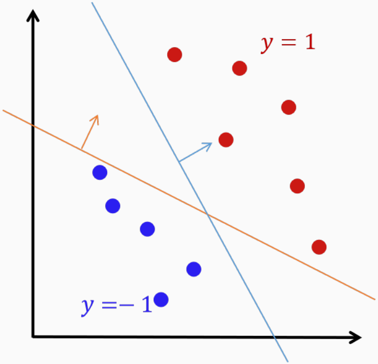
Today's content

1. SVM
2. Lagrange duality and KKT
3. Dual of SVM and kernels
4. SVM with soft constraints and algorithms

SVM

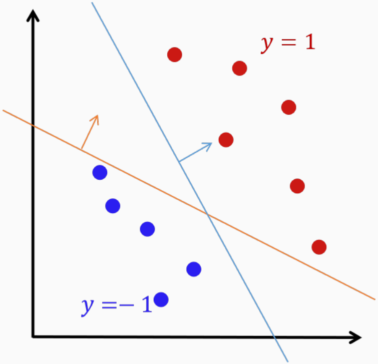
Idea of support vector machine (SVM)

- Data: $x_i \in \mathbb{R}^p$, $y_i \in \{-1, 1\}$ (instead of $\{0, 1\}$ in logistic regression), $i = 1, \ldots, n$
- The two classes are assumed to be linearly separable
- A linear classifier: $f(x) = \text{sign}(\beta^T x + \beta_0)$.



Idea of support vector machine (SVM)

- Data: $x_i \in \mathbb{R}^p$, $y_i \in \{-1, 1\}$ (instead of $\{0, 1\}$ in logistic regression), $i = 1, \ldots, n$
- The two classes are assumed to be linearly separable
- A linear classifier: $f(x) = \text{sign}(\beta^T x + \beta_0)$.

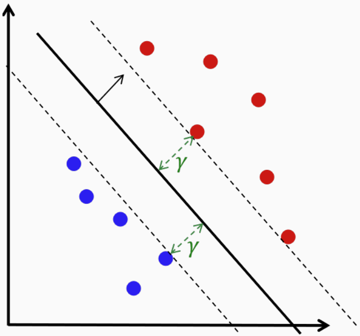


- Question: what is the "best" separating hyperplane?
- SVM answer: the hyperplane with maximum margin.
- Margin = the distance to the closet data points.

Maximum margin separating hyperplane

For the separating hyperplane with maximum margin,

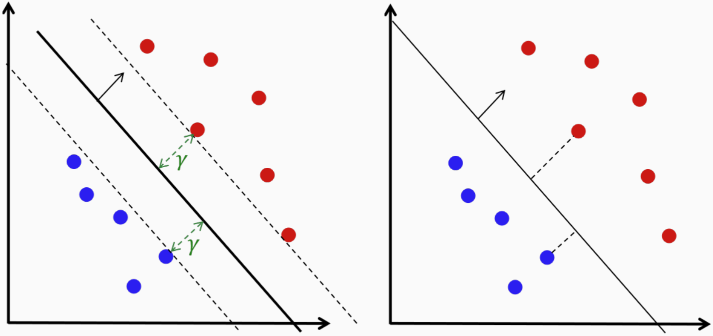
distance to points in positive class = distance to points in negative class



Maximum margin separating hyperplane

For the separating hyperplane with maximum margin,

distance to points in positive class = distance to points in negative class



Normal cone of a hyperplane

Hyperplane $H = H_{\beta,\beta_0} = \{x \in \mathbb{R}^p \mid \beta^T x + \beta_0 = 0\}$

- a linear decision boundary
- $(p-1)$-dimensional subspace, closed, convex

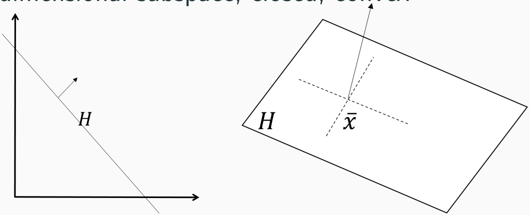


Figure 1: Left: $p = 2$, H is a line. Right: $p = 3$, H is a plane.

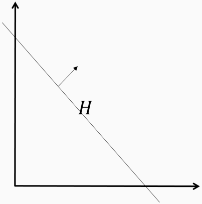
- For any $\bar{x} \in H$, normal cone $N_H(\bar{x}) = \{\lambda\beta \mid \lambda \in \mathbb{R}\}$
  - ▷ The normal cone must be 1-dimensional. We can show that

$$\beta \in N_H(\bar{x}), \text{ i.e., } \langle \beta, z - \bar{x} \rangle \le 0 \quad \forall z \in H.$$

Normal cone of a hyperplane

Hyperplane $H = H_{\beta,\beta_0} = \{x \in \mathbb{R}^p \mid \beta^T x + \beta_0 = 0\}$

- a linear decision boundary
- $(p-1)$-dimensional subspace, closed, convex



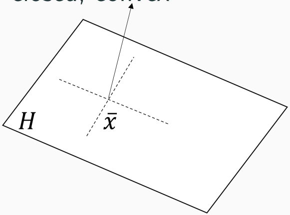


Figure 1: Left: $p = 2$, H is a line. Right: $p = 3$, H is a plane.

- For any $\bar{x} \in H$, normal cone $N_H(\bar{x}) = \{\lambda\beta \mid \lambda \in \mathbb{R}\}$
  - ▷ The normal cone must be 1-dimensional. We can show that

$$\beta \in N_H(\bar{x}), \text{ i.e., } \langle \beta, z - \bar{x} \rangle \leq 0 \quad \forall z \in H.$$

This is true since
$z, \bar{x} \in H \Rightarrow \beta^T z + \beta_0 = 0,\ \beta^T \bar{x} + \beta_0 = 0 \Rightarrow \beta^T (z - \bar{x}) = 0.$
Obviously, we also have $-\beta \in N_H(\bar{x})$.

Distance of a point to a hyperplane

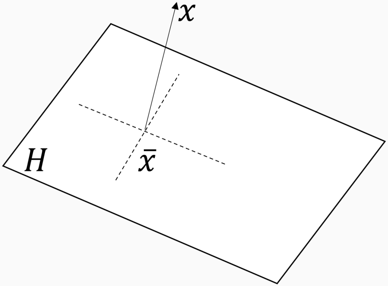
Compute the distance of a point x to a hyperplane
$H = \{x \in \mathbb{R}^p \mid \beta^T x + \beta_0 = 0\}$.

1. $\bar{x} = \Pi_H(x) \iff x - \bar{x} \in N_H(\bar{x}) \iff x - \bar{x} = \lambda\beta$ for some $\lambda \in \mathbb{R}$
2. $\bar{x} \in H \Rightarrow \beta^T \bar{x} + \beta_0 = 0$

   $\Rightarrow \beta^T(x - \lambda\beta) + \beta_0 = 0$

   $\Rightarrow \lambda = \dfrac{\beta^T x + \beta_0}{\beta^T \beta}$
3. $x - \bar{x} = \dfrac{\beta^T x + \beta_0}{\beta^T \beta}\beta$

   $\|x - \bar{x}\| = \dfrac{|\beta^T x + \beta_0|}{\|\beta\|}$



The distance of a point x to a hyperplane H is $\dfrac{|\beta^T x + \beta_0|}{\|\beta\|}$; it is invariant to scaling of the parameters β, β_0

Maximize margin

- Margin $\gamma = \gamma(\beta, \beta_0) = \min_{i=1,\ldots,n} \frac{|\beta^T x_i + \beta_0|}{\|\beta\|}$
- All data points must lie on the correct side:
  $\beta^T x_i + \beta_0 \geq 0$ when $y_i = 1$ $\beta^T x_i + \beta_0 \leq 0$ when $y_i = -1$

$$\iff y_i(\beta^T x_i + \beta_0) \geq 0, \quad \forall\, i \in [n] = \{1, \ldots, n\}$$

- Therefore, the optimization problem is

$$\max_{\beta, \beta_0} \quad \left\{ \min_{i=1,\ldots,n} \frac{|\beta^T x_i + \beta_0|}{\|\beta\|} \right\}$$

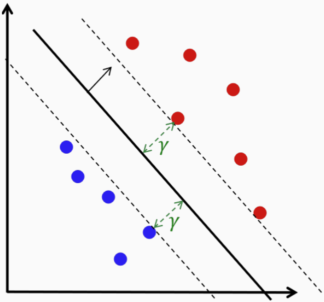
$$\text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 0, \quad \forall\, i \in [n]$$

Simplify the optimization problem

$$\max_{\beta,\beta_0} \quad \frac{1}{\|\beta\|}\left\{\min_{i=1,\ldots,n} |\beta^T x_i + \beta_0|\right\}$$
$$\text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 0 \quad \forall\, i$$

$$\iff$$

$$\max_{\beta,\beta_0} \quad \frac{1}{\|\beta\|}$$
$$\text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 0 \quad \forall\, i$$
$$\min_{i=1,\ldots,n} |\beta^T x_i + \beta_0| = 1$$

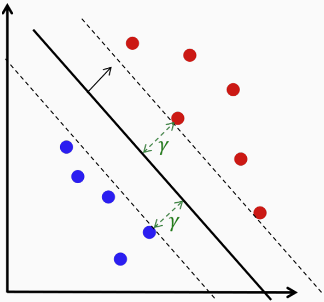


Simplify the optimization problem

$$\max_{\beta,\beta_0} \frac{1}{\|\beta\|}\left\{\min_{i=1,\dots,n} |\beta^T x_i + \beta_0|\right\}$$
$$\text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 0 \quad \forall\, i$$

$$\Longleftrightarrow$$

$$\max_{\beta,\beta_0} \frac{1}{\|\beta\|}$$
$$\text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 0 \quad \forall\, i$$
$$\min_{i=1,\dots,n} |\beta^T x_i + \beta_0| = 1$$



- The hyperplane and margin are scale invariant $(\beta, \beta_0) \to (c\beta, c\beta_0)$, for any $c \neq 0$
- If x_k is the closest point to H, i.e., $k = \arg\min_{i=1,\dots,n} |\beta^T x_i + \beta_0|$, we can scale β, β_0 such that $|\beta^T x_k + \beta_0| = 1$

Simplify the optimization problem

$$\begin{aligned} \max_{\beta,\beta_0} \quad & \frac{1}{\|\beta\|} \\ \text{s.t.} \quad & y_i(\beta^T x_i + \beta_0) \geq 0 \quad \forall\, i \\ & \min_{i=1,\ldots,n} |\beta^T x_i + \beta_0| = 1 \end{aligned} \iff \begin{aligned} \min_{\beta,\beta_0} \quad & \|\beta\|^2 \\ \text{s.t.} \quad & y_i(\beta^T x_i + \beta_0) \geq 1 \quad \forall\, i \end{aligned}$$

- "$\Rightarrow$" Note that $y_i \in \{-1, 1\}$
- "$\Leftarrow$" Note that we minimize $\|\beta\|$

SVM

SVM is a quadratic programming (QP) problem — it can be solved by generic QP solvers

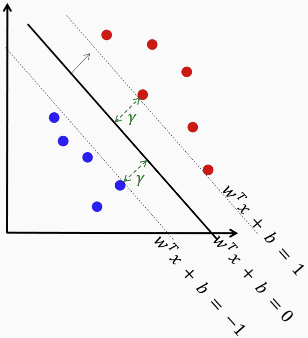
$$
\begin{aligned}
\min_{\beta,\beta_0} \quad & \frac{1}{2}\|\beta\|^2 \\
\text{s.t.} \quad & y_i(\beta^T x_i + \beta_0) \geq 1 \quad \forall\, i \in [n]
\end{aligned}
$$

- Later, we will discuss the Lagrangian duality and derive the dual problem of the above
- The dual problem will play a key role in allowing us to use kernels (introduced later)
- The dual problem will also allow us to derive an efficient algorithm better than generic QP solvers (especially when $n \ll p$)

Support vectors

Support vectors are some x_i having tight constraints

$$y_i(\beta^T x_i + \beta_0) = 1$$



- Support vectors must exist
- Number of support vectors $\ll$ sample size n
- The resulting hyperplane may change if some support vectors are removed

Lagrange duality and KKT

Primal problem

Consider a general nonlinear programming problem (NLP), which is known as a primal problem

$$
\text{(P)} \qquad \begin{aligned} \min_{x \in \mathbb{R}^p} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) = 0,\ i \in [m] \\ & h_j(x) \leq 0,\ j \in [l] \\ & x \in X \end{aligned}
$$

where $X \subseteq \mathbb{R}^p$. The set constraint $x \in X$ is to impose additional requirements, for example

1. $X = \mathbb{R}^p_+$ nonnegativity constraints
2. $X = \mathbb{R}^p$ if there is no special requirement, and $x \in X$ will be omitted in the formulation of the problem

Lagrangian

- Define the Lagrangian

$$L(x, v, u) = f(x) + \sum_{i=1}^{m} v_i g_i(x) + \sum_{j=1}^{l} u_j h_j(x)$$

for $v = [v_1; \ldots; v_m] \in \mathbb{R}^m$, $u = [u_1; \ldots; u_l] \in \mathbb{R}^l_+$.

Lagrangian

- Define the Lagrangian

$$L(x, v, u) = f(x) + \sum_{i=1}^{m} v_i g_i(x) + \sum_{j=1}^{l} u_j h_j(x)$$

  for $v = [v_1; \ldots; v_m] \in \mathbb{R}^m$, $u = [u_1; \ldots; u_l] \in \mathbb{R}^l_+$.
- Define the Lagrange dual function (a concave function)

$$\theta(v, u) = \inf_{x \in X} \; L(x, v, u)$$

- In evaluating $\theta(v, u)$ for each v, u, we must solve

$$\min_{x \in X} \quad L(x, v, u) = f(x) + \sum_{i=1}^{m} v_i g_i(x) + \sum_{j=1}^{l} u_j h_j(x)$$

  We may set $\dfrac{\partial L}{\partial x} = 0$ if $X = \mathbb{R}^p$ and f, g_i, h_j are differentiable

Lagrange dual problem

- Suppose x^* is an optimal solution of (P) (assumed to exist). Then for $v \in \mathbb{R}^m$, $u \in \mathbb{R}^l_+$

$$
\begin{aligned}
\theta(v,u) &= \inf_{x \in X} \; L(x,v,u) \\
&\le L(x^*,v,u) = f(x^*) + \sum_{i=1}^{m} v_i g_i(x^*) + \sum_{j=1}^{l} u_j h_j(x^*) \\
&\le f(x^*)
\end{aligned}
$$

- $\theta(v,u)$ is a lower bound of the primal optimal objective value $f(x^*)$ for any $v \in \mathbb{R}^m$, $u \in \mathbb{R}^l_+$

Lagrange dual problem

- Suppose x^* is an optimal solution of (P) (assumed to exist). Then for $v \in \mathbb{R}^m$, $u \in \mathbb{R}^l_+$

$$\begin{aligned}\theta(v,u) &= \inf_{x \in X} \; L(x,v,u) \\ &\le L(x^*,v,u) = f(x^*) + \sum_{i=1}^m v_i g_i(x^*) + \sum_{j=1}^l u_j h_j(x^*) \\ &\le f(x^*)\end{aligned}$$

- $\theta(v,u)$ is a lower bound of the primal optimal objective value $f(x^*)$ for any $v \in \mathbb{R}^m$, $u \in \mathbb{R}^l_+$
- We want to search for the largest lower bound for $f(x^*)$ — leading to the Lagrange dual problem

$$\text{(D)} \quad \begin{aligned} \max_{v,u} \quad & \theta(v,u) \\ \text{s.t.} \quad & v \in \mathbb{R}^m, \quad u \in \mathbb{R}^l_+ \end{aligned}$$

Here v_i,u_j are called Lagrange dual variables or Lagrange multipliers.

Primal and dual

Definition (Lagrangian dual problem)

For a primal nonlinear programming problem (P)

$$\begin{aligned} \text{(P)} \quad \min_{x \in \mathbb{R}^p} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) = 0,\ i \in [m] \\ & h_j(x) \le 0,\ j \in [l] \\ & x \in X \end{aligned}$$

where $X \subseteq \mathbb{R}^p$. The Lagrangian dual problem (D) is the following nonlinear programming problem

$$\text{(D)} \quad \max_{v,u} \quad \left\{ \theta(v,u) = \inf_{x \in X}\ f(x) + \sum_{i=1}^{m} v_i g_i(x) + \sum_{j=1}^{l} u_j h_j(x) \right\}$$

$$\text{s.t.} \quad v \in \mathbb{R}^m, \quad u \in \mathbb{R}^l_+$$

- Weak duality: optimal value for (D) $\le$ optimal value for (P)
- Under certain assumptions (see page 18),
  strong duality: optimal value for (D) $=$ objective value for (P)

Example

Find the Lagrange dual problem of the convex program

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 4 \end{aligned}$$

Example

Find the Lagrange dual problem of the convex program

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 & \qquad & f(x) = x_1^2 + x_2^2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 4 & & h_1(x) = 4 - x_1 - x_2 \leq 0 \quad \leftarrow u_1 \geq 0 \end{aligned}$$

Example

Find the Lagrange dual problem of the convex program

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 & \qquad & f(x) = x_1^2 + x_2^2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 4 & \qquad & h_1(x) = 4 - x_1 - x_2 \leq 0 \quad \leftarrow u_1 \geq 0 \end{aligned}$$

Solution. For $u_1 \geq 0$, the Lagrangian is

$$L(x_1, x_2, u_1) = f(x) + u_1 h_1(x) = x_1^2 + x_2^2 + u_1(4 - x_1 - x_2)$$

Example

Find the Lagrange dual problem of the convex program

$$\begin{array}{lll} \min & x_1^2 + x_2^2 & f(x) = x_1^2 + x_2^2 \\ \text{s.t.} & x_1 + x_2 \geq 4 & h_1(x) = 4 - x_1 - x_2 \leq 0 \quad \leftarrow u_1 \geq 0 \end{array}$$

Solution. For $u_1 \geq 0$, the Lagrangian is

$$L(x_1, x_2, u_1) = f(x) + u_1 h_1(x) = x_1^2 + x_2^2 + u_1(4 - x_1 - x_2)$$

The Lagrange dual function is

$$\begin{aligned} \theta(u_1) &= \inf_{x_1, x_2} x_1^2 + x_2^2 + u_1(4 - x_1 - x_2) \\ &= 4u_1 + \inf_{x_1} \{x_1^2 - u_1 x_1\} + \inf_{x_2} \{x_2^2 - u_1 x_2\} \\ &= 4u_1 - \frac{u_1^2}{2} \text{ (Attained at } x_1 = \frac{u_1}{2},\, x_2 = \frac{u_2}{2}) \end{aligned}$$

The Lagrange dual problem is

$$\begin{array}{ll} \max & 4u_1 - \dfrac{u_1^2}{2} \\ \text{s.t.} & u_1 \geq 0 \end{array}$$

Example: LP

Consider the linear programming (LP) problem in standard form

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $x \geq 0$ means $x_i \geq 0$, $i \in [n]$.
Find the Lagrange dual function and the Lagrange dual problem.

Example: LP

Consider the linear programming (LP) problem in standard form

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $x \geq 0$ means $x_i \geq 0$, $i \in [n]$.
Find the Lagrange dual function and the Lagrange dual problem.

Solution. Let $X = \mathbb{R}^n_+$ and $v \in \mathbb{R}^m$. The Lagrange dual function is

$$\begin{aligned} \theta(v) &= \inf_{x \in X} \{c^T x + v^T(b - Ax)\} = v^T b + \inf_{x \in \mathbb{R}^n_+} \{x^T(c - A^T v)\} \\ &= \begin{cases} v^T b, & \text{if } c - A^T v \in \mathbb{R}^n_+ \\ -\infty, & \text{otherwise} \end{cases} \end{aligned}$$

The Lagrange dual problem is
$$\begin{aligned} \max_{v} \quad & b^T v \\ \text{s.t.} \quad & A^T v \leq c \end{aligned}$$

Example: LP

Consider the LP in standard inequality form

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and the inequality in the constraint $Ax \leq b$ is interpreted component-wise. Find the Lagrange dual function and the Lagrange dual problem.

Example: LP

Consider the LP in standard inequality form

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and the inequality in the constraint $Ax \le b$ is interpreted component-wise. Find the Lagrange dual function and the Lagrange dual problem.

Solution. Let $u \in \mathbb{R}^m_+$. The Lagrange dual function is

$$\begin{aligned} \theta(u) &= \inf_{x \in \mathbb{R}^n} \{c^T x + u^T(Ax - b)\} = -u^T b + \inf_{x \in \mathbb{R}^n} \{x^T(c + A^T u)\} \\ &= \begin{cases} -u^T b, & \text{if } c + A^T u = 0 \\ -\infty, & \text{otherwise} \end{cases} \end{aligned}$$

The Lagrange dual problem is $\quad \begin{aligned} \max_u \quad & -b^T u \\ \text{s.t.} \quad & A^T u + c = 0 \\ & u \ge 0 \end{aligned}$

KKT

Assumptions

1. No additional constraint $x \in X$, i.e., $X = \mathbb{R}^p$
2. f, $h_j : \mathbb{R}^p \to \mathbb{R}$ differentiable and convex
3. $g_i : \mathbb{R}^p \to \mathbb{R}$ affine ($g_i(x) = a_i^T x + b_i$)
4. Slater's condition holds, i.e., there exits $\hat{x}$ such that

$$g_i(\hat{x}) = 0, \forall\, i \qquad h_j(\hat{x}) < 0, \forall\, j$$

Under the above assumptions, strong duality holds, and there exist a solution x^* to (P) and a solution (u^*, v^*) to (D) satisfying the Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial}{\partial x} L(x^*, u^*, v^*) = \nabla f(x^*) + \sum_{i=1}^{m} v_i^* \nabla g_i(x^*) + \sum_{j=1}^{l} u_j^* \nabla h_j(x^*) = 0$$

$$g_i(x^*) = 0,\ h_j(x^*) \leq 0,\ u_j^* \geq 0,\ u_j^* h_j(x^*) = 0, \quad \forall\, i \in [m], j \in [l]$$

KKT

- We say (x^*, u^*, v^*) (or simply x^*) is a KKT point or a KKT solution if (x^*, u^*, v^*) satisfies the KKT conditions
- Under the above assumptions, (x^*, u^*, v^*) is a KKT solution $\iff$ x^* is an optimal solution to (P) and (u^*, v^*) is an optimal solution to (D)
- We call

$$u_j^* h_j(x^*) = 0, \forall\, j \in [l]$$

complementary slackness condition. It implies

$$u_j^* = 0 \text{ if } h_j(x^*) < 0, \quad h_j(x^*) = 0 \text{ if } u_j^* > 0$$

$$\begin{cases} h_j(x^*) \leq 0 \\ u_j^* \geq 0 \\ u_j^* h_j(x^*) = 0 \end{cases}$$

KKT

- We say (x^*, u^*, v^*) (or simply x^*) is a KKT point or a KKT solution if (x^*, u^*, v^*) satisfies the KKT conditions
- Under the above assumptions, (x^*, u^*, v^*) is a KKT solution $\iff$ x^* is an optimal solution to (P) and (u^*, v^*) is an optimal solution to (D)
- We call

$$u_j^* h_j(x^*) = 0, \forall\, j \in [l]$$

complementary slackness condition. It implies

$$u_j^* = 0 \text{ if } h_j(x^*) < 0, \quad h_j(x^*) = 0 \text{ if } u_j^* > 0$$

$$\begin{cases} h_j(x^*) \leq 0 \\ u_j^* \geq 0 \\ u_j^* h_j(x^*) = 0 \end{cases}$$

- If the constraint $h_j(x^*) \leq 0$ is slack ($h_j(x^*) < 0$), then the constraint $u_j^* \geq 0$ is active ($u_j^* = 0$)
- If the constraint $u_j^* \geq 0$ is slack ($u_j^* > 0$), then the constraint $h_j(x^*) \leq 0$ is active ($h_j(x^*) = 0$)

Dual of SVM

Dual of SVM

Derive the dual of the following SVM problem

$$\begin{aligned} \min_{\beta,\beta_0} \quad & \frac{1}{2}\|\beta\|^2 \\ \text{s.t.} \quad & 1 - y_i(\beta^T x_i + \beta_0) \le 0 \quad \forall\, i \in [n] \end{aligned}$$

For $\alpha \in \mathbb{R}^n_+$, the Lagrangian is

$$L(\beta, \beta_0, \alpha) = \frac{1}{2}\|\beta\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i(\beta^T x_i + \beta_0))$$

The Lagrange dual function

$$\begin{aligned} \theta(\alpha) &= \inf_{\beta,\beta_0} \quad L(\beta, \beta_0, \alpha) \\ &= \inf_{\beta,\beta_0} \quad \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{n} \alpha_i y_i x_i^T \beta - \sum_{i=1}^{n} \alpha_i y_i \beta_0 + \sum_{i=1}^{n} \alpha_i \end{aligned}$$

Dual of SVM

We need to solve the optimization problem

$$\min_{\beta,\beta_0} \quad \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{n}\alpha_i y_i x_i^T\beta - \sum_{i=1}^{n}\alpha_i y_i \beta_0 + \sum_{i=1}^{n}\alpha_i$$

Setting $\quad \dfrac{\partial}{\partial\beta}L = \beta - \sum_{i=1}^{n}\alpha_i y_i x_i = 0, \qquad \dfrac{\partial}{\partial\beta_0}L = -\sum_{i=1}^{n}\alpha_i y_i = 0,$

Dual of SVM

We need to solve the optimization problem

$$\min_{\beta,\beta_0} \quad \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^n \alpha_i y_i x_i^T \beta - \sum_{i=1}^n \alpha_i y_i \beta_0 + \sum_{i=1}^n \alpha_i$$

Setting $\dfrac{\partial}{\partial \beta} L = \beta - \sum_{i=1}^n \alpha_i y_i x_i = 0, \qquad \dfrac{\partial}{\partial \beta_0} L = -\sum_{i=1}^n \alpha_i y_i = 0$, we obtain that

$$\theta(\alpha) = \begin{cases} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j & \text{if } \sum_{i=1}^n \alpha_i y_i = 0 \\ -\infty & \text{otherwise} \end{cases}$$

The Lagrange dual problem is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0,\ i \in [n] \end{aligned}$$

KKT of SVM

Primal

$$\begin{aligned} \min_{\beta,\beta_0} \quad & \frac{1}{2}\|\beta\|^2 \\ \text{s.t.} \quad & y_i(\beta^T x_i + \beta_0) \geq 1,\ i \in [n] \end{aligned}$$

Dual

$$\begin{aligned} \max_{\alpha} \ & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ \text{s.t.} \ & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \geq 0,\ i \in [n] \end{aligned}$$

- Verify the assumptions (in Page 18) for strong duality and the existence of KKT points: (Slater's condition) there exists $\hat{\beta}, \hat{\beta}_0$ such that $y_i(\hat{\beta}^T x_i + \hat{\beta}_0) > 1,\ i \in [n]$. It requires that the two classes are strictly separable.
- KKT conditions:

$$\sum_{i=1}^n \alpha_i^* y_i x_i = \beta^*, \quad \sum_{i=1}^n \alpha_i^* y_i = 0$$

$$y_i((\beta^*)^T x_i + \beta_0^*) \geq 1,\ \alpha_i^* \geq 0,\ \alpha_i^*(1 - y_i((\beta^*)^T x_i + \beta_0^*)) = 0,\ i \in [n]$$

Insights from the dual

- If we obtain a solution u^* (via solving the SVM dual problem), then we can construct a primal solution (β^*, β_0^*) from KKT conditions

$$\beta^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

$$\beta_0^* = y_k - \sum_{i=1}^{n} \alpha_i^* y_i \langle x_i, x_k \rangle, \text{ for some } k \text{ satisfying } u_k > 0$$

- If $\alpha_i^* > 0$, then x_i is a support vector. By strict complementarity

$$\alpha_i^* > 0 \Rightarrow y_i((\beta^*)^T x_i + \beta_0^*) = 1$$

- Decision boundary

$$0 = (\beta^*)^T x + \beta_0^* = \sum_{i=1}^{n} \alpha_i^* y_i \langle x_i, x \rangle + \beta_0^*$$

- For a new test point x, the prediction only depends on $\langle x_i, x \rangle$ where $\alpha_i^* > 0$, namely, the inner products between x and support vectors. (The number of support vectors is usually small)

Insights from the dual

Primal

$$\min_{\beta,\beta_0} \quad \frac{1}{2}\|\beta\|^2$$
$$\text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 1,\ i \in [n]$$

Classifier

$$f(x) = \text{sign}(\beta^T x + \beta_0)$$

Dual

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$
$$\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$\alpha_i \geq 0,\ i \in [n]$$

Classifier

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + \beta_0\right)$$

Many α_i's are zero (sparse solutions)

- Optimize $p+1$ variables for primal, n variables for dual
- When $n \ll p$, it might be more efficient to solve the dual
- Dual problem only involves $\langle x_i, x_j \rangle$ — allowing the use of kernels

Feature mapping

- Recall feature expansion, for example,

$$i\text{-th feature vector } x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \quad \overset{\text{feature}}{\underset{\text{expansion}}{\longrightarrow}} \quad \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i1}^2 \\ x_{i2}^2 \\ x_{i1}x_{i2} \end{bmatrix}$$

- Let ϕ denote the feature mapping, which maps from original features to new features

$$\text{For example,} \quad \phi\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right) = \begin{bmatrix} z_1 \\ z_2 \\ z_1^2 \\ z_2^2 \\ z_1 z_2 \end{bmatrix}$$

- Instead of using the original feature vectors x_i, $i \in [n]$, we may apply SVM using new features $\phi(x_i)$, $i \in [n]$
- New feature space can be very high dimensional

Kernel

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \, i \in [n]$$

- To use feature expansion, simply replace $\langle x_i, x_j \rangle$ with $\langle \phi(x_i), \phi(x_j) \rangle$
- Given a feature mapping ϕ, we define the corresponding kernel

$$K(a, b) = \langle \phi(a), \phi(b) \rangle, \quad a, b \in \mathbb{R}^p$$

- Usually computing $K(a, b)$ may be very cheap, even though computing $\phi(a), \phi(b)$ (high dimensional vectors) may be expensive
- The dual of SVM only requires the computation of kernels $K(x_i, x_j)$. Explicitly calculating $\phi(x_i)$ is not necessary

Example: (homogeneous) polynomial kernel

For $a, b \in \mathbb{R}^p$, consider

$$K(a,b) = (a^T b)^2$$

It can be written as

$$\begin{aligned} K(a,b) &= \left(\sum_{i=1}^{p} a_i b_i\right)\left(\sum_{j=1}^{p} a_j b_j\right) = \sum_{i,j=1}^{p} a_i a_j b_i b_j \\ &= \sum_{i,j=1}^{p} (a_i a_j)(b_i b_j) \end{aligned}$$

Thus, we see that $K(a,b) = \langle \phi(a), \phi(b) \rangle$, where the feature mapping $\phi : \mathbb{R}^p \to \mathbb{R}^{p^2}$ is given by

$$\phi(a) = \phi\left(\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}\right) = \begin{bmatrix} a_1 a_1 \\ a_1 a_2 \\ a_1 a_3 \\ \vdots \\ a_p a_p \end{bmatrix}$$

Computing $\phi(a)$: $O(p^2)$ operations; computing $K(a,b)$: $O(p)$ operations

Example: (inhomogeneous) polynomial kernel

Given $c \geq 0$. For $a, b \in \mathbb{R}^p$, consider

$$\begin{aligned} K(a,b) &= (a^T b + c)^2 \\ &= \sum_{i,j=1}^{p} (a_i a_j)(b_i b_j) + \sum_{i=1}^{p} (\sqrt{2c} a_i)(\sqrt{2c} b_i) + c^2 \end{aligned}$$

Thus, we see that $K(a,b) = \langle \phi(a), \phi(b) \rangle$, where the feature mapping $\phi : \mathbb{R}^p \to \mathbb{R}^{(p^2+p+1)}$ is given by

$$\phi(a) = \Big(\underbrace{a_1 a_1, a_1 a_2, a_1 a_3, \ldots, a_p a_p,}_{\text{second order terms}} \underbrace{\sqrt{2c} a_1, \sqrt{2c} a_2, \ldots, \sqrt{2c} a_p,}_{\text{first order terms}} \; c \Big)^T$$

Parameter c controls the relative weighting between first order and second order terms.

Common kernels

- Polynomials of degree d

$$K(a,b) = (a^T b)^d$$

- Polynomials up to degree d

$$K(a,b) = (a^T b + 1)^d$$

- Gaussian kernel — polynomials of all orders[1]

$$K(a,b) = \exp\left(-\frac{\|a-b\|^2}{2\sigma^2}\right), \quad \sigma > 0$$

[1] $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$

Kernel

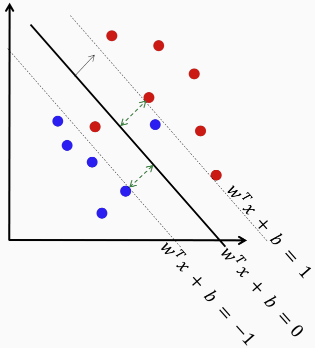
- SVM can be applied in high dimensional feature spaces, without explicitly applying the feature mapping
- The two classes might be separable in high dimensional space, but not separable in the original feature space
- Kernels can be used efficiently in the dual problem of SVM because the dual only involves inner products

SVM with soft constraints

SVM with soft constraints

When the two classes are not separable, no feasible separating hyperplane exits. We allow the constraints to be violated slightly ($C > 0$ is given)

$$\begin{aligned}
\min_{\beta,\beta_0,\varepsilon} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n} \varepsilon_i \\
\text{s.t.} \quad & y_i(\beta^T x_i + \beta_0) \geq 1 - \varepsilon_i \quad \forall\, i \in [n] \\
& \varepsilon_i \geq 0,\, i \in [n]
\end{aligned}$$



SVM with soft constraints

When the two classes are not separable, no feasible separating hyperplane exits. We allow the constraints to be violated slightly ($C > 0$ is given)

$$\begin{aligned} \min_{\beta,\beta_0,\varepsilon} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n} \varepsilon_i \\ \text{s.t.} \quad & y_i(\beta^T x_i + \beta_0) \geq 1 - \varepsilon_i \quad \forall\, i \in [n] \\ & \varepsilon_i \geq 0,\ i \in [n] \end{aligned}$$

$$\begin{aligned} \varepsilon_i &= \begin{cases} 1 - y_i(\beta^T x_i + \beta_0), & \text{if } y_i(\beta^T x_i + \beta_0) < 1 \\ 0, & \text{if } y_i(\beta^T x_i + \beta_0) \geq 1 \end{cases} \\ &= \max\{1 - y_i(\beta^T x_i + \beta_0), 0\} \end{aligned}$$

SVM with soft constraints

When the two classes are not separable, no feasible separating hyperplane exits. We allow the constraints to be violated slightly ($C > 0$ is given)

$$\begin{aligned} \min_{\beta,\beta_0,\varepsilon} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n} \varepsilon_i \\ \text{s.t.} \quad & y_i(\beta^T x_i + \beta_0) \geq 1 - \varepsilon_i \quad \forall\, i \in [n] \\ & \varepsilon_i \geq 0,\, i \in [n] \end{aligned}$$

$$\begin{aligned} \varepsilon_i &= \begin{cases} 1 - y_i(\beta^T x_i + \beta_0), & \text{if } y_i(\beta^T x_i + \beta_0) < 1 \\ 0, & \text{if } y_i(\beta^T x_i + \beta_0) \geq 1 \end{cases} \\ &= \max\{1 - y_i(\beta^T x_i + \beta_0), 0\} \end{aligned}$$

SVM with soft constraints solves

$$\min_{\beta,\beta_0} \quad \underbrace{\frac{1}{2}\|\beta\|^2}_{\text{ridge regularization}} \quad \underbrace{+\, C\sum_{i=1}^{n} \max\{1 - y_i(\beta^T x_i + \beta_0), 0\}}_{\text{hinge-loss function}}$$

Logistic regression

$$\text{logistic-loss} = \begin{cases} \log\left(1 + e^{-(\beta^T x_i + \beta_0)}\right), & \text{if } y_i = 1 \\ \log\left(1 + e^{\beta^T x_i + \beta_0}\right), & \text{if } y_i = 0 \end{cases}$$

Logistic regression

$$\text{logistic-loss} = \begin{cases} \log\left(1 + e^{-(\beta^T x_i + \beta_0)}\right), & \text{if } y_i = 1 \\ \log\left(1 + e^{\beta^T x_i + \beta_0}\right), & \text{if } y_i = 0 \end{cases}$$

Change label $y_i = 0 \to y_i = -1$,

$$\text{logistic-loss} = \log\left(1 + e^{-y_i(\beta^T x_i + \beta_0)}\right), \quad y_i \in \{-1, 1\}$$

Logistic regression with ridge regularization

$$\min_{\beta, \beta_0} \quad \sum_{i=1}^{n} \log\left(1 + e^{-y_i(\beta^T x_i + \beta_0)}\right) + \lambda \|\beta\|^2$$

SVM vs. logistic regression

SVM with soft constraints

$$\min_{\beta,\beta_0} \; C\sum_{i=1}^{n} \max\{1 - y_i(\beta^T x_i + \beta_0), 0\} + \frac{1}{2}\|\beta\|^2$$

Hinge-loss

hinge-loss $= \max\{1 - z, 0\}$

$z = y_i(\beta^T x_i + \beta_0)$

hope $z \geq 1$

Logistic regression with ridge regularization

$$\min_{\beta,\beta_0} \; \sum_{i=1}^{n} \log\left(1 + e^{-y_i(\beta^T x_i + \beta_0)}\right) + \lambda\|\beta\|^2$$

Logistic-loss

logistic-loss $= \log(1 + e^{-z})$

$z = y_i(\beta^T x_i + \beta_0)$

hope $z \gg 0$

SVM vs. logistic regression

SVM with soft constraints

$$\min_{\beta,\beta_0} \; C\sum_{i=1}^{n} \max\{1-y_i(\beta^T x_i+\beta_0),0\} + \frac{1}{2}\|\beta\|^2$$

Hinge-loss

hinge-loss $= \max\{1-z,0\}$

$z = y_i(\beta^T x_i + \beta_0)$

hope $z \geq 1$

Logistic regression with ridge regularization

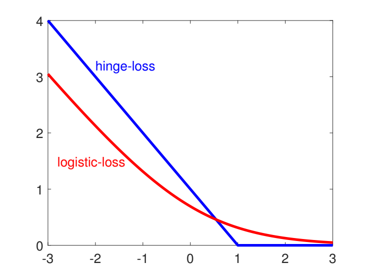
$$\min_{\beta,\beta_0} \; \sum_{i=1}^{n} \log\left(1+e^{-y_i(\beta^T x_i+\beta_0)}\right) + \lambda\|\beta\|^2$$

Logistic-loss

logistic-loss $= \log(1+e^{-z})$

$z = y_i(\beta^T x_i + \beta_0)$

hope $z \gg 0$



logistic loss is a "smoothed version" of hinge loss

SVM with soft constraints: dual and KKT

SVM with soft constraints

$$\begin{aligned} \min_{\beta,\beta_0,\varepsilon} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \varepsilon_i \\ \text{s.t.} \quad & 1 - \varepsilon_i - y_i(\beta^T x_i + \beta_0) \le 0 \quad \forall\, i \in [n] \\ & -\varepsilon_i \le 0,\ i \in [n] \end{aligned}$$

For $u \in \mathbb{R}^n_+, r \in \mathbb{R}^n_+$, the Lagrangian $L(\beta, \beta_0, \varepsilon, u, r) =$

$$\frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \varepsilon_i + \sum_{i=1}^n \alpha_i(1 - \varepsilon_i - y_i(\beta^T x_i + \beta_0)) - \sum_{i=1}^n r_i\varepsilon_i$$

[Exercise] Derive Lagrange dual problem:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C,\ i \in [n] \end{aligned}$$

SMO

Next, SMO (sequential minimal optimization) [1] algorithm for solving the dual problem with kernel techniques

$$
\begin{aligned}
\min_{\alpha} \quad & \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_{ij} - \sum_{i=1}^{n} \alpha_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& 0 \le \alpha_i \le C,\ i \in [n]
\end{aligned}
$$

Here one may choose a feature mapping ϕ and compute the $n \times n$ kernel matrix K: $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$

Idea of SMO: block coordinate descent

Block coordinate descent

Idea: minimizing of a multivariable function with k blocks

$$\min_{x \in \mathbb{R}^n} f(x_1, x_2, \ldots, x_l)$$

$$x_j \in \mathbb{R}^{n_j}, n_1 + \cdots + n_l = n$$

can be achieved by minimizing it over one block x_j at a time.

Algorithm (Block coordinate descent)

Choose $x^{(0)}$. Set $k \leftarrow 0$

repeat until convergence

for $j = 1, \ldots, l$

Update the j-th block $x_j^{(k)}$ with all other blocks fixed

end(for)

$k \leftarrow k + 1$

end(repeat)

Cyclic coordinate descent

The simplest and most often used case might be cyclic coordinate descent

Algorithm (Cyclic coordinate descent)

Choose $x^{(0)}$. Set $k \leftarrow 0$

repeat until convergence

for $i = 1, \ldots, n$

$$x_i^{(k+1)} = \arg\min_y \, f(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, y, x_{i+1}^{(k)}, \ldots, x_n^{(k)})$$

end(for)

$k \leftarrow k + 1$

end(repeat)

SMO

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j K_{ij} - \sum_{i=1}^{n}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \le \alpha_i \le C,\, i \in [n] \end{aligned}$$

SMO has the following iterations:

Step 1. Select a pair (α_i, α_j) to update; see [1, Sec 2.2]

Step 2. Update (α_i, α_j) with all other α_k's $(k \neq i, j)$ fixed

For example, $i = 1$, $j = 2$, we update (α_1, α_2) via

$$\begin{aligned} \min_{\alpha_1,\alpha_2} \quad & \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_1 y_2 K_{12}\alpha_1\alpha_2 - \alpha_1 - \alpha_2 \\ \text{s.t.} \quad & y_1\alpha_1 + y_2\alpha_2 = \zeta \\ & 0 \le \alpha_1 \le C,\, 0 \le \alpha_2 \le C \end{aligned}$$

The solution of the above problem has an explicit form; see [1, Sec 2.1]

In practice

You are encouraged to learn two popular open source machine learning libraries:

LIBLINEAR `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`

LIBSVM `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

References i

J. Platt.
Sequential minimal optimization: A fast algorithm for training support vector machines.
Technical Report MSR-TR-98-14, Microsoft, April 1998.