6. Support Vector Machine

Dongwoo Kim

dongwoo.kim@postech.ac.kr

CSED515 - 2023 Spring

Table of Contents

1 Maximum Margin Classifier

2 Understanding constraint optimization and duality

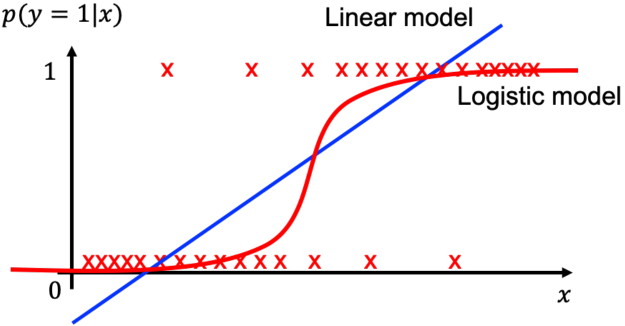
3 Getting to know binary Support Vector Machine (SVM)

4 Comparing SVMs and logistic regression

Recap) Regression for Classification

Approximating probability $p(y = 1 \mid x)$ from dataset $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1,\dots,N}$ where $y_i \in \{-1, 1\}$

- Linear regression: $p(y = 1 \mid x) \approx \boldsymbol{w}^\top \phi(x)$
- Logit regression: $p(y = 1 \mid x) \approx \frac{1}{1+\exp(-\boldsymbol{w}^\top \phi(x))}$

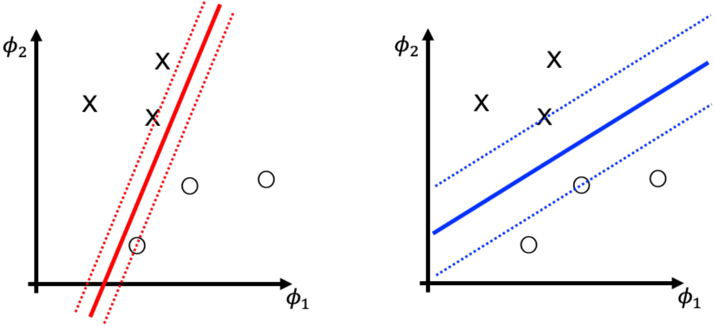


Note that each of those classifiers is drawing a separating hyperplane!

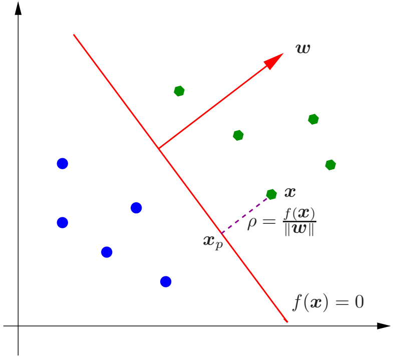
What is a good separating hyperplane?

Given dataset $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1,\ldots,N}$ of binary classification, which (linear) classifier do you prefer?

- You may prefer one with larger **margin** (the minimal distance from classifier to data points)



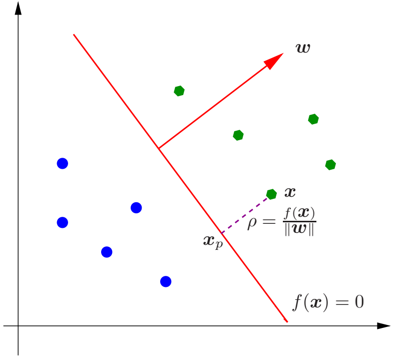
A Computation of Margin



- Consider decision boundary $f(\boldsymbol{x}) := \boldsymbol{w}^\top \boldsymbol{x} + b = 0$ (for given $b > 0$)
- Margin: distance from separating hyperplane to the closest examples in the dataset.
- Given $\boldsymbol{x}$, the length of the orthogonal projection of $\boldsymbol{x}$ onto the hyperplane is ρ such that

$$\boldsymbol{x} = \boldsymbol{x}_p + \rho \frac{\boldsymbol{w}}{||\boldsymbol{w}||}$$

Distance from Hyperplane



- Note that $\boldsymbol{x}_p = \boldsymbol{x} - \rho \frac{\boldsymbol{w}}{||\boldsymbol{w}||}$
- $f(\boldsymbol{x}_p)$ needs to be 0. Therefore,

$$\langle \boldsymbol{w}, \boldsymbol{x} - \rho \frac{\boldsymbol{w}}{||\boldsymbol{w}||} \rangle + b = 0$$

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b - \rho \frac{\langle \boldsymbol{w}, \boldsymbol{w} \rangle}{||\boldsymbol{w}||} = 0$$

- The length ρ is then

$$\rho = \frac{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b}{||\boldsymbol{w}||} = \frac{f(\boldsymbol{x})}{||\boldsymbol{w}||}$$

Maximizing Margin (1): Canonical Hyperplane

Note that for any $\lambda \neq 0$, $(\lambda \boldsymbol{w}, \lambda b)$ describes the same hyperplane as $(\boldsymbol{w}, b)$, i.e.,

$$\{\boldsymbol{x} | \boldsymbol{w}^\top \boldsymbol{x} + b = 0\} = \{\boldsymbol{x} | \lambda(\boldsymbol{w}^\top \boldsymbol{x} + b) = 0\}$$

Hence, we say that $(\boldsymbol{w}, b)$ is canonical if

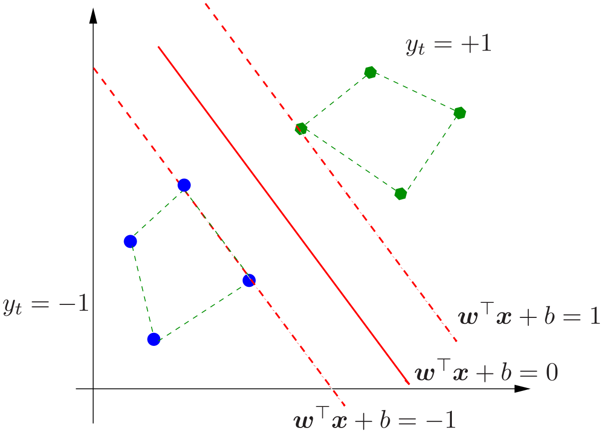
$$\min_i |\boldsymbol{w}^\top \boldsymbol{x}_i + b| = 1 \ .$$

Remark

We first assume that the dataset is linearly separable, and then relax the constraint later.

Maximizing Margin (1): Canonical Optimal Hyperplane

Support vectors!


$y_t = +1$
$y_t = -1$
$\boldsymbol{w}^\top \boldsymbol{x} + b = 1$
$\boldsymbol{w}^\top \boldsymbol{x} + b = 0$
$\boldsymbol{w}^\top \boldsymbol{x} + b = -1$

Margin $= 1/\|\boldsymbol{w}\|$

The geometric margin in previous slide is given by

$$\begin{aligned} \rho &= \frac{1}{2}\left\{\frac{f(\boldsymbol{x}^{+})}{\|\boldsymbol{w}\|} - \frac{f(\boldsymbol{x}^{-})}{\|\boldsymbol{w}\|}\right\} \\ &= \frac{1}{2}\frac{1}{\|\boldsymbol{w}\|}\left\{\underbrace{\boldsymbol{w}^{\top}\boldsymbol{x}^{+} - \boldsymbol{w}^{\top}\boldsymbol{x}^{-}}_{2}\right\} \\ &= \frac{1}{\|\boldsymbol{w}\|}. \end{aligned}$$

Thus, maximizing margin is equivalent to minimizing the norm of the weight vector in the discriminant function.

Maximizing Margin (2)

- Noting the canonical hyperplane, the problem of maximizing margin can be formally formulated as:

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{subject to} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \quad \forall i = 1, ..., N \end{aligned}$$

  where $y_i \in \{-1, +1\}$.

- The solution is called binary Support Vector Machine (SVM)
- How to solve the constrained optimization?

Table of Contents

1 Maximum Margin Classifier

2 Understanding constraint optimization and duality

3 Getting to know binary Support Vector Machine (SVM)

4 Comparing SVMs and logistic regression

Constrained optimization problem

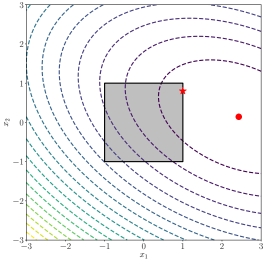


Figure: Constraints: $-1 \leq x_1, x_2 \leq 1$

- Sometimes, we want to restrict model parameters within a certain range.
- Consider a objective function $L : \mathbb{R}^D \to \mathbb{R}$ with constraints

  $$\min_{\boldsymbol{x}} L(\boldsymbol{x})$$

  subject to $g_i(\boldsymbol{x}) \leq 0$ for all $i = 1, ..., m.$

  where[1] $g_i : \mathbb{R}^D \to \mathbb{R}$.
- This is called a primal problem.

[1] $g_i(\boldsymbol{x}) \geq 0 \Rightarrow -g_i(\boldsymbol{x}) \leq 0$

From constrained to unconstrained

- Since multiple equations are hard to optimize together, we can convert the constrained optimization into an unconstrained one with an indicator function

  $$J(\boldsymbol{x}) = L(\boldsymbol{x}) + \sum_{i=1}^{m} \mathbf{1}(g_i(\boldsymbol{x}))$$

  where $\mathbf{1}(z)$ is an infinite step function

  $$\mathbf{1}(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ \infty & \text{otherwise} \end{cases}$$

- If $g_i(\boldsymbol{x})$ violates any constraint, $J(\boldsymbol{x})$ becomes ∞.

Lagrangian Dual

Definition (Lagrangian Dual)

Let the *Lagrangian* be

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = L(\boldsymbol{x}) + \sum_{i=1} \overbrace{\lambda_i}^{\text{Lagrangian multiplier}} g_i(\boldsymbol{x})$$

Then, the Lagrangian dual of the primal problem is given by

$$\begin{aligned} \max_{\boldsymbol{\lambda} \in \mathbb{R}^m} \quad & \mathcal{D}(\boldsymbol{\lambda}) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned}$$

where $\mathcal{D}(\boldsymbol{\lambda}) = \min_{\boldsymbol{x} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$.

Minimax Inequality

Definition (Minimax inequality)

$\max_{\boldsymbol{y}} \min_{\boldsymbol{x}} \rho(\boldsymbol{x}, \boldsymbol{y}) \leq \min_{\boldsymbol{x}} \max_{\boldsymbol{y}} \rho(\boldsymbol{x}, \boldsymbol{y})$

Proof.

Define $\psi(\boldsymbol{y}) \triangleq \min_{\boldsymbol{x}} \rho(\boldsymbol{x}, \boldsymbol{y})$.

$$\begin{aligned}
\forall \boldsymbol{x}, \boldsymbol{y}, \quad & \psi(\boldsymbol{y}) \leq \rho(\boldsymbol{x}, \boldsymbol{y}) \\
& \Rightarrow \forall \boldsymbol{x}, \max_{\boldsymbol{y}} \psi(\boldsymbol{y}) \leq \max_{\boldsymbol{y}} \rho(\boldsymbol{x}, \boldsymbol{y}) \\
& \Rightarrow \max_{\boldsymbol{y}} \psi(\boldsymbol{y}) \leq \min_{\boldsymbol{x}} \max_{\boldsymbol{y}} \rho(\boldsymbol{x}, \boldsymbol{y}) \\
& \Rightarrow \max_{\boldsymbol{y}} \min_{\boldsymbol{x}} \rho(\boldsymbol{x}, \boldsymbol{y}) \leq \min_{\boldsymbol{x}} \max_{\boldsymbol{y}} \rho(\boldsymbol{x}, \boldsymbol{y})
\end{aligned}$$

□

Weak Duality with Minimax Inequality

- Let's apply minimax inequality to $J(\boldsymbol{x})$ and $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$.

$$J(\boldsymbol{x}) = L(\boldsymbol{x}) + \sum_{i=1}^{m} \mathbf{1}(g_i(\boldsymbol{x})) = L(\boldsymbol{x}) + \max_{\lambda \geq \mathbf{0}} \sum_{i=1} \lambda_i g_i(\boldsymbol{x}) = \max_{\lambda \geq \mathbf{0}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$$

Recall the aim of the primal problem is to minimize $J(\boldsymbol{x})$:

$$\min_{\boldsymbol{x}} J(\boldsymbol{x}) = \min_{\boldsymbol{x}} \max_{\lambda \geq \mathbf{0}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$$

By applying minimax inequality, we obtain

$$\underbrace{\min_{\boldsymbol{x}} \max_{\lambda \geq \mathbf{0}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})}_{\text{Primal solution}} \geq \underbrace{\max_{\lambda \geq \mathbf{0}} \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})}_{\text{Dual solution}}$$

- Therefore, by maximizing Lagrangian, we can obtain the lower bound of the primal problem *(i.e. weak duality)*.

Why Dual?

- The inner part of r.h.s is the dual objective:

$$\min_{\boldsymbol{x}} \max_{\lambda \geq \mathbf{0}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \geq \max_{\lambda \geq \mathbf{0}} \underbrace{\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})}_{\text{unconstrained}}$$

- Given $\boldsymbol{\lambda}$, $\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is unconditional optimization problem.
- If $\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is easy given $\boldsymbol{\lambda}$, then the overall problem is easy to solve.
  - The outer problem is a maximum over a set of affine functions, and hence is a concave function.
  - Even though $\mathcal{L}$ may be nonconvex. The maximum of a concave function can be efficiently computed.

Equality constraints

- To handle equality constraints $h(\boldsymbol{x}) = 0$, we can add two inequalities:

$$\begin{aligned} h(\boldsymbol{x}) &\leq 0 \\ h(\boldsymbol{x}) &\geq 0 \quad \Leftrightarrow \quad -h(\boldsymbol{x}) \leq 0 \end{aligned}$$

- The resulting Lagrange multipliers are then unconstrained.

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}, \lambda) &= L(\boldsymbol{x}) + \lambda_1 h(\boldsymbol{x}) - \lambda_2 h(\boldsymbol{x}), \qquad \lambda_1, \lambda_2 \geq 0 \\ &= L(\boldsymbol{x}) - \lambda h(\boldsymbol{x}), \qquad \lambda \in \mathbb{R} \end{aligned}$$

With Equality Constraints

Consider a primal form of constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & \mathcal{J}(\boldsymbol{w}), \\ \text{subject to} & g_i(\boldsymbol{w}) \leq 0, \quad i = 1, \ldots, M, \\ & h_j(\boldsymbol{w}) = 0, \quad j = 1, \ldots, L. \end{array}$$

Lagrangian is given by

$$\mathcal{L}(w, \nu, \lambda) = \mathcal{J}(w) + \sum_{i=1}^{M} \nu_i g_i(w) + \sum_{j=1}^{L} \lambda_j h_j(w)$$

where $\nu_i \geq 0$ for $i = 1, \ldots, M$ and λ_j for $j = 1, \ldots, L$ are unrestricted in sign.

Karush-Kuhn-Tucker (KKT) Necessary Condition

If $\boldsymbol{w}$ is solution, then

1. Optimality

$$\nabla \mathcal{L} = \nabla \mathcal{J}(\boldsymbol{w}) + \sum_{i=1}^{M} \nu_i \nabla g_i(\boldsymbol{w}) + \sum_{j=1}^{L} \lambda_j \nabla h_j(\boldsymbol{w}) = 0$$

2. Feasibility

$$\begin{aligned} g_i(\boldsymbol{w}) &\leq 0, \quad i = 1, \ldots, M \\ h_j(\boldsymbol{w}) &= 0, \quad j = 1, \ldots, L \end{aligned}$$

3. Complementary slackness

$$\nu_i g_i(\boldsymbol{w}) = 0, \quad i = 1, \ldots, M \quad (\nu_i \geq 0)\,.$$

Table of Contents

Max Margin Classifier: Primal Form

Primal Problem

$$\min_{\boldsymbol{w}} \mathcal{J}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{subject to } y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \geq 1, \quad i = 1, \ldots, N$$

Proposition

Given a linearly separable training sample $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, the hyperplane $\boldsymbol{w}^\top \boldsymbol{x}_i + b = 0$ that solves the above optimization problem realizes the maximal margin hyperplane with geometric margin $\rho = 1/\|\boldsymbol{w}\|$.

Primal Lagrangian

Primal Lagrangian $\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})$ is given by

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

$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0$ and $\frac{\partial \mathcal{L}}{\partial b} = 0$ yield respectively $\boxed{\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i}$ and $\boxed{\sum_{i=1}^{N} \alpha_i y_i = 0}$. Substitute these relations into the primal form, leading to

$$\mathcal{L} = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j + \sum_{i=1}^{N} \alpha_i.$$

Max Margin Classifier: Dual Form

Dual Problem

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \mathcal{G}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j + \sum_{i=1}^{N}\alpha_i, \\
\text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \\
& \alpha_i \geq 0, \quad i = 1, \ldots, N.
\end{aligned}
$$

Proposition

Suppose that $\boldsymbol{\alpha}^*$ solves the dual problem, given a linearly separable training sample $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. Then the weight vector $\boldsymbol{w}^* = \sum_{i=1}^N \alpha_i^* y_i \boldsymbol{x}_i$ realizes the maximal margin hyperplane with geometric margin $\rho = 1/\|\boldsymbol{w}\|$.

Dual SVM

- Consider the input $\mathbf{x} \in \mathbb{R}^D$ with D features.
- The dimension of variable $\mathbf{w}$ is also D.
- The number of parameters, D, increases linearly with the number of features.
- What if we have infinitely many number of features?
- With dual form, we only need to find N α's

Support Vectors

Note that $\boxed{\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i}$, i.e., $\boldsymbol{w}$ is a linear combination of training data points $\boldsymbol{x}_i$.

It follows from KKT complimentary slackness condition that we have

$$\alpha_i \left[1 - y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right)\right] = 0.$$

- $y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) \neq 1$ (x_i is not support vector) $\Rightarrow \alpha_i = 0$ (x_i is irrelevant).
- $\alpha_i \neq 0 \Rightarrow y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right) = 1$ (x_i is support vector).

Only support vectors influence the computation of $\boldsymbol{w}$.

Slack Variables

In cases where data are not linearly separable

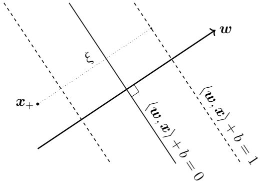

$\boldsymbol{w}$
ξ
$\boldsymbol{x}_+$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 1$

Figure: Slack variable ζ (ξ in figure)

- To handle the data point on the wrong side of hyperplane
- We introduce *slack variable* ξ that measures the relative distance to positive margin hyperplane.
- With the slack variable,

$$y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq 1 - \zeta_n$$

$$n = 1, ..., N$$

Soft Margin Classifier: Primal - L2

In cases where data are not linearly separable, the optimization problem cannot be solved as the primal has an empty feasible region and the dual an unbounded objective function.

To sidestep this problem, we introduce slack variables $\zeta_i \geq 0$ to allow the margin constraints to be violated:

$$y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right) \geq 1 - \zeta_i, \quad i = 1, \ldots, N.$$

Primal Problem

$$\begin{array}{ll} \displaystyle\min_{\boldsymbol{w}, b, \boldsymbol{\zeta}} & \mathcal{J}(\boldsymbol{w}, \boldsymbol{\zeta}) = \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \zeta_i^2, \\ \text{subject to} & y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right) \geq 1 - \zeta_i, \quad i = 1, \ldots, N. \end{array}$$

- The primal Lagrangian is

$$\mathcal{L}(\boldsymbol{w}, b, \zeta) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\zeta_i^2 + \sum_{i=1}^{N}\alpha_i\left(1 - \zeta_i - y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right)\right).$$

- $\frac{\partial \mathcal{J}(\boldsymbol{w},\boldsymbol{\zeta})}{\partial \boldsymbol{w}} = 0$ leads to $\boldsymbol{w} = \sum_{i=1}^{N}\alpha_i y_i \boldsymbol{x}_i$.
- $\frac{\partial \mathcal{J}(\boldsymbol{w},\boldsymbol{\zeta})}{\partial b} = 0$ leads to $\sum_{i=1}^{N}\alpha_i y_i = 0$.
- $\frac{\partial \mathcal{J}(\boldsymbol{w},\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} = 0$ leads to $\boldsymbol{\zeta} = \frac{\boldsymbol{\alpha}}{C}$.
- Substitute these relations into the primal Lagrangian to obtain the dual Lagrangian function:

$$\mathcal{G}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j - \frac{1}{2C}\sum_{i=1}^{N}\alpha_i^2 + \sum_{i=1}^{N}\alpha_i.$$

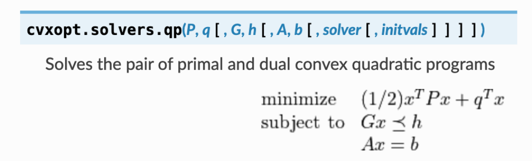
Soft Margin Classifier: Dual - L2

Dual Problem

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \mathcal{G}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \left(\boldsymbol{x}_i^\top \boldsymbol{x}_j + \frac{1}{C}\delta_{ij}\right) + \sum_{i=1}^{N}\alpha_i, \\
\text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \\
& \alpha_i \geq 0, \quad i = 1, \dots, N.
\end{aligned}
$$

Soft Margin Classifier: Optimization

- The primal or dual can be solved by commonly used convex optimization packages.
- `cvxopt`: Python library for convex optimization
- `https://cvxopt.org/userguide/coneprog.html#quadratic-programming`



`cvxopt.solvers.qp`(*P*, *q* [, *G*, *h* [, *A*, *b* [, *solver* [, *initvals*]]]])

Solves the pair of primal and dual convex quadratic programs

$$\begin{array}{ll} \text{minimize} & (1/2)x^T Px + q^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

Soft-Margin SVM: Loss Perspective

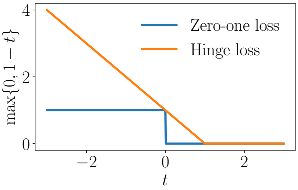


Figure: Hinge Loss

- When slack ζ_n is greater than 0, we can think it as a loss.
- What would be a corresponding loss in this case?
- $\ell(t) = \max\{0, 1 - (\boldsymbol{w}^\top \boldsymbol{x} + b)\}$
- This is called a hinge loss.

- Note that if $t \in (0, 1)$ we can still classify $\boldsymbol{x}$ correctly, however this will incur some loss.
- Because we want to have no data points within a margin.

Unconstrained Optimization Problem

- With the hinge loss, we can reformulate the soft-margin SVM as an unconstrained optimization problem.

$$\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N} \max\{0, 1 - y_n(\langle \boldsymbol{w}, \boldsymbol{x}_n \rangle) - b\}$$

- From empirical risk minimization perspective, $\frac{1}{2}||\boldsymbol{w}||^2$ can be interpreted as a regularizer of parameter $\boldsymbol{w}$.
- This optimization can be solved by (sub-)gradient descent.

Table of Contents

1. Maximum Margin Classifier
2. Understanding constraint optimization and duality
3. Getting to know binary Support Vector Machine (SVM)
4. Comparing SVMs and logistic regression

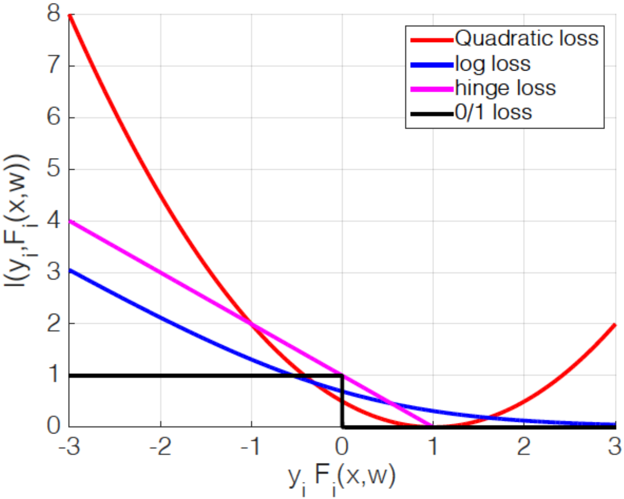
Summary

$$\text{Linear regression:} \quad \min_{\boldsymbol{w},b} \quad \frac{C'}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \frac{1}{2}\left(1 - y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)\right)^2$$

$$\text{Logit regression:} \quad \min_{\boldsymbol{w},b} \quad \frac{C'}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)\right)\right)$$

$$\text{Binary SVM:} \quad \min_{\boldsymbol{w},b} \quad \frac{C'}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \max\left\{0, 1 - y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)\right\}$$

Summary


Quadratic loss
log loss
hinge loss
0/1 loss
$l(y_i, F_i(x,w))$
$y_i\ F_i(x,w)$
8
7
6
5
4
3
2
1
0
-3
-2
-1
0
1
2
3

Generalization

Linear regression: $$\min_{\boldsymbol{w},b} \quad \frac{C'}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \frac{1}{2}\left(1 - y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)\right)^2$$

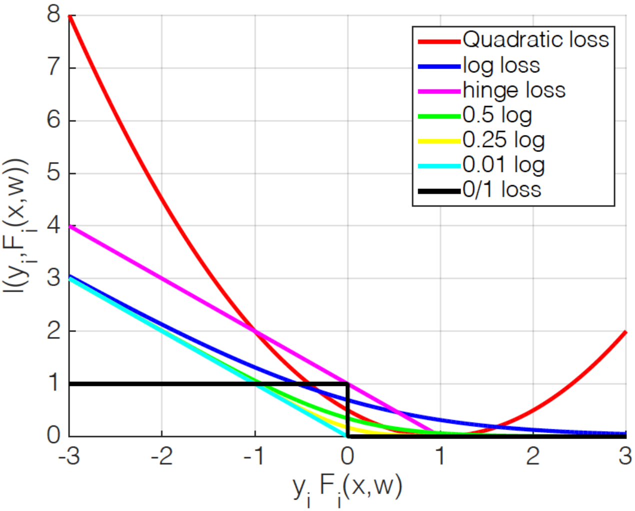
Logit regression: $$\min_{\boldsymbol{w},b} \quad \frac{C'}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)\right)\right)$$

Binary SVM: $$\min_{\boldsymbol{w},b} \quad \frac{C'}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \max\left\{0, 1 - y_i(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i))\right\}$$

General binary cls: $$\min_{\boldsymbol{w},b} \quad \frac{C'}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \varepsilon \log\left(1 + \exp\left(\frac{L - y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)}{\varepsilon}\right)\right)$$

Noting that for $L = 1$, as $\varepsilon \to 0$, the formulation of general classification converges to binary SVM

Generalization



Further Readings

- Textbook: Chapter 17.3 (Support vector machines)
- Textbook: Chapter 8.5 (Constrained optimization)