Kernel SVM

Sriram Sankararaman

The instructor gratefully acknowledges Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

Outline

1. Review of last lecture
   - SVM – margin maximization view
2. SVM – Hinge loss
3. Lagrange duality theory

Support Vector Machine

Goal: A linear classifier (hyperplane) that separates positive and negative training examples

- Separable training dataset, i.e., we assume there exists a decision boundary that separates the two classes perfectly.
- However there are an *infinite* number of hyperplanes that separate the two classes.
- Select a hyperplane as far away from every training point as possible (has large margin).
- Finding the hyperplane that maximizes the margin leads to Support Vector Machines.
- Requires us to solve a constrained optimization problem.

SVM for separable data (Hard-margin SVM)

Assuming separable training data, we thus want to solve:

$$\max_{\boldsymbol{w},b} \underbrace{\frac{1}{\|\boldsymbol{w}\|_2}}_{\text{Margin}} \quad \text{such that} \quad y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad n \in 1, \ldots, N$$

This is equivalent to

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad n \in 1, \ldots, N \end{aligned}$$

SVM is called a *max margin* (or large margin) classifier.

SVM for separable data (Hard-margin SVM)

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad n \in 1, \ldots, N \end{aligned}$$

- Margin constraints are hard constraints: hence hard-margin SVM.
- If data is not linearly separable
  - There is no $(\boldsymbol{w}, b)$ that satisfies all the N constraints.
  - The optimization problem is infeasible.

SVM for non-separable data

Constraints in non-separable setting

- Modify our constraints to account for non-separability.
- Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad n \in \{1, \ldots, N\}$$

- Intuition: If the hyperplane misclassifies example n, then ξ_n is the distance we need to move this example to get it to the correct side of the hyperplane.
- But we need to pay a price for moving points across.

Soft-margin SVM formulation

We do not want ξ_n to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \underbrace{\frac{1}{2}\|\boldsymbol{w}\|_2^2}_{\text{Large margin}} + C\sum_n \underbrace{\xi_n}_{\text{Small slack}}$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad n \in \{1, \ldots, N\}$$
$$\xi_n \geq 0, \quad n \in \{1, \ldots, N\}$$

- But if example n is incorrectly classified, you can set the slack ξ_n to move this example across the hyperplane.

Soft-margin SVM formulation

We do not want ξ_n to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \underbrace{\frac{1}{2}\|\boldsymbol{w}\|_2^2}_{\text{Large margin}} + C\sum_n \underbrace{\xi_n}_{\text{Small slack}}$$

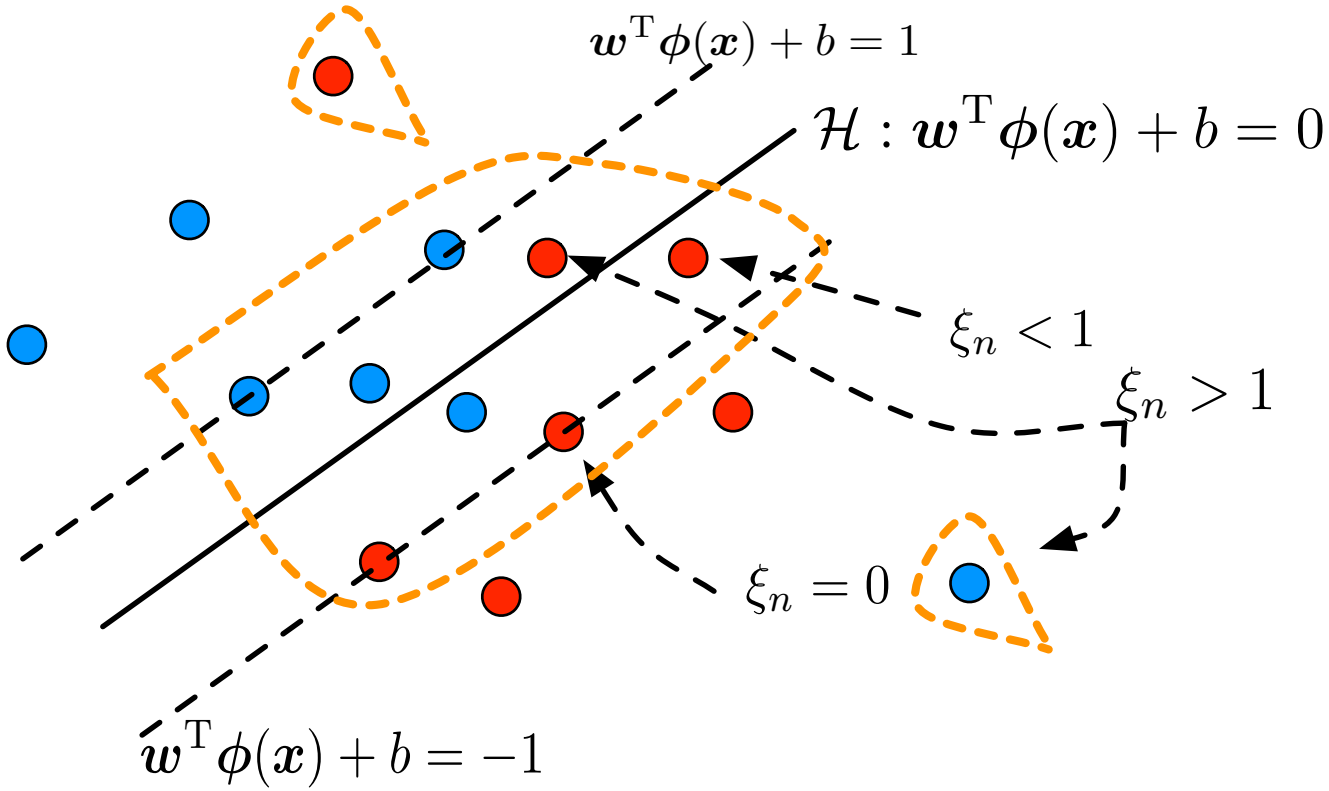
$$\text{s.t.} \quad y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad n \in \{1, \ldots, N\}$$
$$\xi_n \geq 0, \quad n \in \{1, \ldots, N\}$$

- Unlike the hard-margin SVM, the soft-margin SVM is always feasible, *i.e.* there exists values for the variables $(\boldsymbol{w}, b, \boldsymbol{\xi})$ that satisfy all the constraints.
- This is a convex program that can be solved with general-purpose or specialized solvers.

Meaning of "support vectors" in SVMs

- The SVM solution is only determined by a subset of the training instances.
- These instances are called *support vectors*
- All other training points do not affect the optimal solution, i.e.,if remove the other points and construct another SVM classifier on the reduced dataset, the optimal solution will be the same

Visualization of how training data points are categorized



Support vectors are highlighted by the dotted orange lines

Outline

1. Review of last lecture
2. SVM – Hinge loss
3. Lagrange duality theory

A general view of supervised learning

Definition Assume $y \in \{-1, 1\}$ and the decision rule is $h(\boldsymbol{x}) = \text{SIGN}(a(\boldsymbol{x}))$ with $a(\boldsymbol{x}) = \boldsymbol{w}^{\text{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b$

For classification: $0/1$ loss

$$\ell^{0/1}(y, a(\boldsymbol{x})) = \begin{cases} 0 & \text{if } ya(\boldsymbol{x}) \geq 0 \\ 1 & \text{otherwise} \end{cases}$$

Minimize weighted sum of empirical risk and regularizer

$$\begin{aligned}
&\arg\min_{\boldsymbol{w},b} R^{\mathrm{EMP}}[h_{\boldsymbol{w},b}(x)] + \lambda R(\boldsymbol{w}, b) \\
&= \arg\min_{\boldsymbol{w},b} \frac{1}{N} \sum_n \ell(y_n, a(\boldsymbol{x}_n)) + \lambda R(\boldsymbol{w}, b) \qquad (1)
\end{aligned}$$

Minimize weighted sum of empirical risk and regularizer

$$\begin{aligned} &\arg\min_{\boldsymbol{w},b} R^{\mathrm{EMP}}[h_{\boldsymbol{w},b}(x)] + \lambda R(\boldsymbol{w}, b) \\ &= \arg\min_{\boldsymbol{w},b} \frac{1}{N} \sum_{n} \ell(y_n, a(\boldsymbol{x}_n)) + \lambda R(\boldsymbol{w}, b) \end{aligned} \tag{1}$$

- Problem with minimizing the $0/1$ loss ?

Hinge loss

Definition Assume $y \in \{-1, 1\}$ and the decision rule is $h(\boldsymbol{x}) = \text{SIGN}(a(\boldsymbol{x}))$ with $a(\boldsymbol{x}) = \boldsymbol{w}^{\text{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b$,

$$\ell^{\text{HINGE}}(y, a(\boldsymbol{x})) = \begin{cases} 0 & \text{if } ya(\boldsymbol{x}) \geq 1 \\ 1 - ya(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

Intuition

Hinge loss

Definition Assume $y \in \{-1, 1\}$ and the decision rule is $h(\boldsymbol{x}) = \text{SIGN}(a(\boldsymbol{x}))$ with $a(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b$,

$$\ell^{\text{HINGE}}(y, a(\boldsymbol{x})) = \begin{cases} 0 & \text{if } ya(\boldsymbol{x}) \geq 1 \\ 1 - ya(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

Intuition

- No penalty if raw output, $a(\boldsymbol{x})$, has same sign and is far enough from decision boundary (i.e., if 'margin' is large enough)
- Otherwise pay a growing penalty, between 0 and 1 if signs match, and greater than one otherwise

Hinge loss

Definition Assume $y \in \{-1, 1\}$ and the decision rule is $h(\boldsymbol{x}) = \text{SIGN}(a(\boldsymbol{x}))$ with $a(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b$,

$$\ell^{\text{HINGE}}(y, a(\boldsymbol{x})) = \begin{cases} 0 & \text{if } ya(\boldsymbol{x}) \geq 1 \\ 1 - ya(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

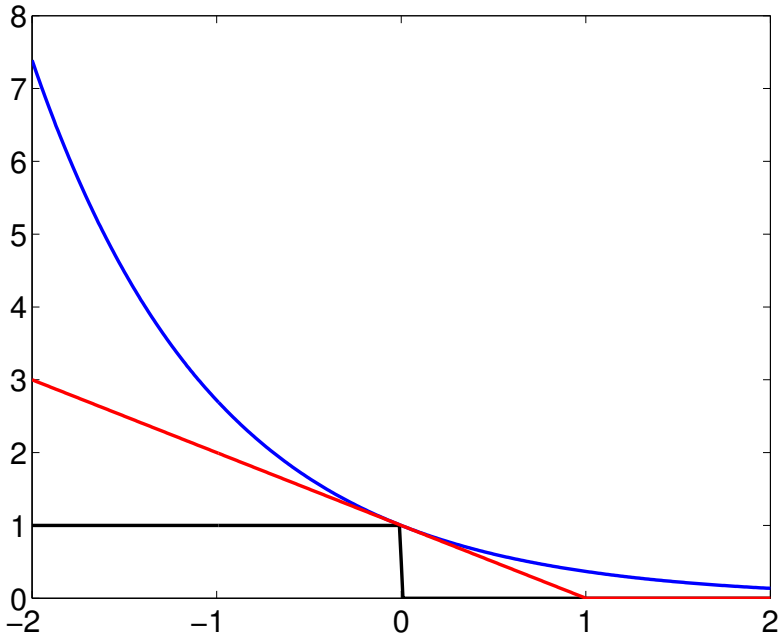
Intuition

- No penalty if raw output, $a(\boldsymbol{x})$, has same sign and is far enough from decision boundary (i.e., if 'margin' is large enough)
- Otherwise pay a growing penalty, between 0 and 1 if signs match, and greater than one otherwise

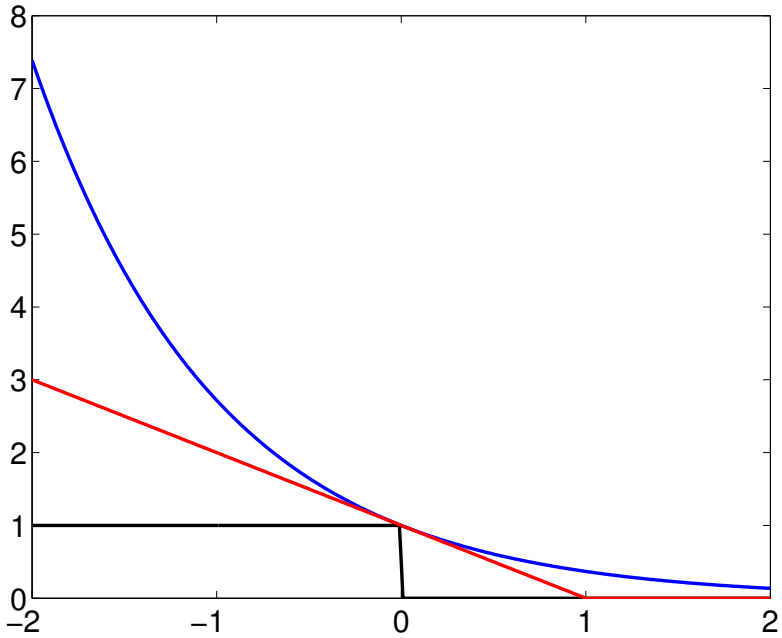
Convenient shorthand

$$\ell^{\text{HINGE}}(y, a(\boldsymbol{x})) = \max(0, 1 - ya(\boldsymbol{x})) = (1 - ya(\boldsymbol{x}))_+$$

Visualization and Properties

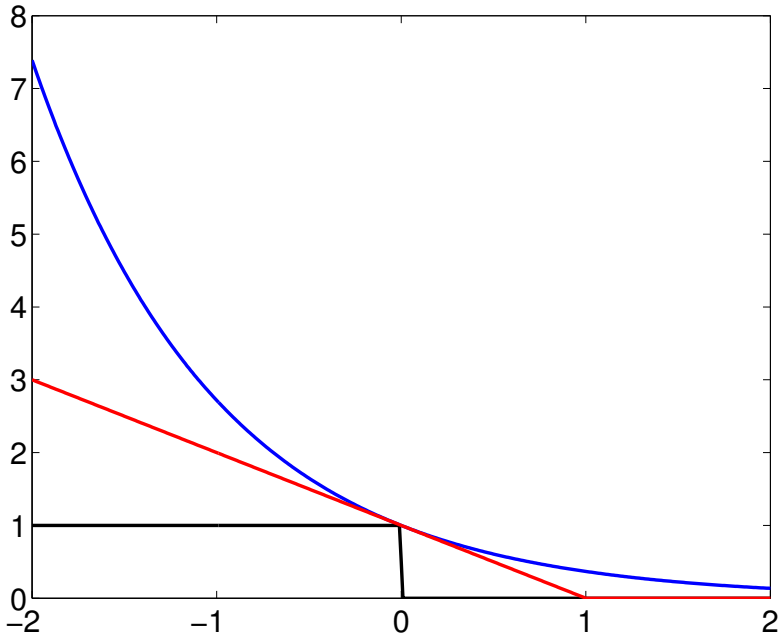


Visualization and Properties



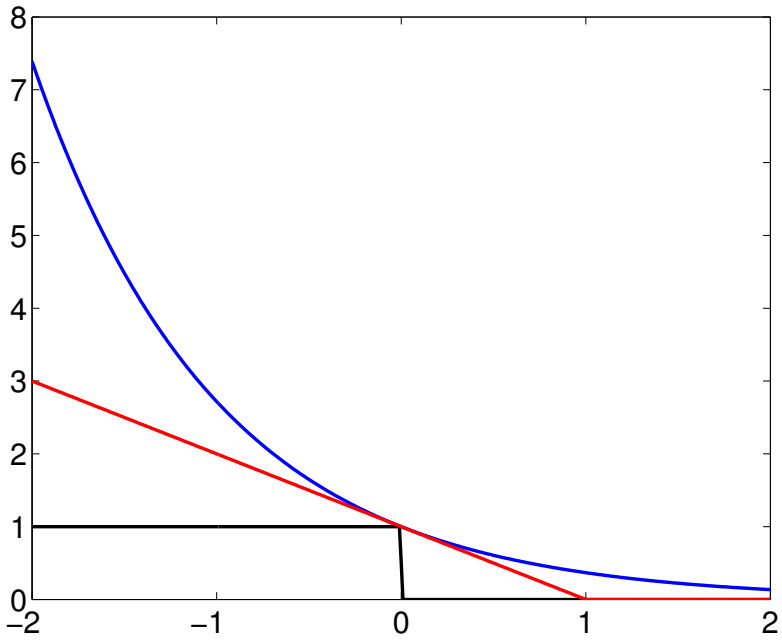
- Upper-bound for $0/1$ loss function (black line)
- We use hinge loss as a *surrogate* to $0/1$ loss – Why?

Visualization and Properties



- Upper-bound for $0/1$ loss function (black line)
- We use hinge loss as a *surrogate* to $0/1$ loss – Why?
- Hinge loss is convex, and thus easier to work with.

Visualization and Properties



- Other surrogate losses can be used, e.g., exponential loss for Adaboost (in blue), logistic loss (not shown) for logistic regression
- Hinge loss less sensitive to outliers than exponential (or logistic) loss

Hinge loss

Definition Assume $y \in \{-1, 1\}$ and the decision rule is $h(\boldsymbol{x}) = \text{SIGN}(a(\boldsymbol{x}))$ with $a(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b$,

$$\ell^{\text{HINGE}}(y, a(\boldsymbol{x})) = \begin{cases} 0 & \text{if } ya(\boldsymbol{x}) \geq 1 \\ 1 - ya(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

Intuition

- No penalty if raw output, $a(\boldsymbol{x})$, has same sign and is far enough from decision boundary (i.e., if 'margin' is large enough)
- Otherwise pay a growing penalty, between 0 and 1 if signs match, and greater than one otherwise

Convenient shorthand

$$\ell^{\text{HINGE}}(y, a(\boldsymbol{x})) = \max(0, 1 - ya(\boldsymbol{x})) = (1 - ya(\boldsymbol{x}))_+$$

SVM: the hinge loss minimization view

SVM is the linear classifier that minimizes the total hinge loss on all the training data (+ l_2 regularizer)

$$\min_{\boldsymbol{w},b} \sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

is equivalent to the previous viewpoint of the SVM as the linear classifier that has maximum margin

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \quad n \in \{1, \ldots, N\} \\ & 0 \leq \xi_n, \quad n \in \{1, \ldots, N\} \end{aligned}$$

Recovering our previous SVM formulation

Minimizing the total hinge loss on all the training data

$$\min_{\boldsymbol{w},b} \sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

Define $C = 1/\lambda$:

$$\min_{\boldsymbol{w},b} C\sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

Recovering our previous SVM formulation

Minimizing the total hinge loss on all the training data

$$\min_{\boldsymbol{w},b} \sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

Define $C = 1/\lambda$:

$$\min_{\boldsymbol{w},b} C\sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

Define $\xi_n = \max(0, 1 - y_n[\boldsymbol{w}^{\mathbf{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b])$

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) = \xi_n, \quad n \in \{1, \ldots, N\} \end{aligned}$$

Recovering our previous SVM formulation

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) = \xi_n, \quad n \in \{1, \ldots, N\} \end{aligned}$$

is equivalent to

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \quad n \in \{1, \ldots, N\} \\ & 0 \leq \xi_n, \quad n \in \{1, \ldots, N\} \end{aligned}$$

At optimal solution constraints are active so we have equality! Why?

At optimal solution constraints are active so we have equality! Why?

- If $\xi_n^* > \max(0, 1 - y_n f(\boldsymbol{x}_n))$, we could choose $\bar{\xi}_n < \xi_n^*$ and still satisfy the constraint while reducing our objective function!
- Since $c \geq \max(a, b) \iff c \geq a,\ c \geq b$, we recover previous formulation

Outline

1. Review of last lecture
2. SVM – Hinge loss
3. Lagrange duality theory
   - A simple example
   - SVM dual and kernel SVM
   - SVM Dual Formulation and Kernel SVM
   - SVM Dual Derivation and Support Vectors

Goal: Kernelize SVMs

- Rewrite the SVM optimization problem so that it no longer depends on $\boldsymbol{\phi}(\boldsymbol{x}_n)$ but instead depends only on inner products $\boldsymbol{\phi}(\boldsymbol{x}_n)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_m)$.
- If we can do this, we can then replace $\boldsymbol{\phi}(\boldsymbol{x}_n)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_m)$ with a kernel function $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$.
- Before doing that we will need to learn some more optimization.

Kernel SVM Roadmap

Key concepts we'll cover

- Brief review of constrained optimization
  - "Primal" and "Dual" problems
  - Strong Duality and KKT conditions
- Dual SVM problem and Kernel SVM

How to solve a constrained optimization problem?

- We know how to solve an unconstrained optimization problem (for example, the least squares cost function).
- To solve a constrained optimization problem:
  - Done using the technique of Lagrange multipliers.

Constrained Optimization

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0, \quad i \in \{1, \ldots, m\} \\ & h_j(\boldsymbol{x}) = 0, \quad j \in \{1, \ldots, p\} \end{aligned}$$

This is the primal problem

- Variable $\boldsymbol{x}$
- Constraints : m inequality constraints, p equality constraints
- As a convention, we write all inequality constraints as $f(\boldsymbol{x}) \leq 0$.
- Assume $f_0, f_1, \ldots, f_m$ are convex. h_j are affine functions : $h_j(\boldsymbol{x}) = \boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x} + b_j$.

Constrained Optimization

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0, \quad i \in \{1, \ldots, m\} \\ & h_j(\boldsymbol{x}) = 0, \quad j \in \{1, \ldots, p\} \end{aligned}$$

We can define the Lagrangian for the primal problem:

$$L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \alpha_i f_i(\boldsymbol{x}) + \sum_{j=1}^{p} \beta_j h_j(\boldsymbol{x})$$

- Every constraint associated with a new variable called Lagrange multiplier.
- α_i is the Lagrange multiplier associated with the i^{th} inequality constraint.
- β_j is the Lagrange multiplier associated with the j^{th} equality constraint.

We can show that $\min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ has same solution as primal problem, which we denote as p^*

Constrained Optimization

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0, \quad i \in \{1, \ldots, m\} \\ & h_j(\boldsymbol{x}) = 0, \quad j \in \{1, \ldots, p\} \end{aligned}$$

Consider the following function:

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

- If $\boldsymbol{x}$ violates a primal constraint, this function $= \infty$; otherwise $= f_0(\boldsymbol{x})$
- Thus $\min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ has same solution as primal problem, which we denote as p^*

Constrained Optimization

Primal Problem

$$p^* = \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Dual Problem

Consider the function: $g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

Constrained Optimization

Primal Problem

$$p^* = \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Dual Problem

Consider the function: $g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

$$d^* = \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Constrained Optimization

Primal Problem

$$p^* = \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Dual Problem

Consider the function: $g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

$$d^* = \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Primal and dual are the same, except the max and min are exchanged!

Relationship between primal and dual?

Constrained Optimization

Primal Problem

$$p^* = \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Dual Problem

Consider the function: $g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

$$d^* = \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\alpha_i \geq 0} \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Primal and dual are the same, except the max and min are exchanged!

Relationship between primal and dual?

- In general, $d^* \leq p^*$ (weak duality): dual solution is a lower bound on the primal solution.
- $p^* = d^*$ under conditions that can be verified (strong duality)
- This property holds for the SVM optimization problem but does not hold in general!

Strong Duality

When $p^* = d^*$, we can solve the dual problem instead of the primal problem!

Strong Duality

When $p^* = d^*$, we can solve the dual problem instead of the primal problem!

Under these assumptions, there must exist $\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ such that:

- $\boldsymbol{x}$ are primal variables, $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are dual variables
- $\boldsymbol{x}^*$ is the solution to the primal and $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ is the solution to the dual
- $p^* = d^* = L(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$

Recap

- When working with constrained optimization problems, we can write down primal and dual problems
- The dual solution is always a lower bound on the primal solution (weak duality)
- The duality solution is equal to the primal solution for the SVM problem (strong duality).

A simple example

- We throw a die N times.
- In each independent trial, the die can land on one of K faces and the probability of landing on a face is given by θ_k.
- We let random variable X_n denote the outcome of trial n.

$$P(X_n = k) = \theta_k$$

For this to be a valid probability distribution, we need $\sum_{k=1}^{K} \theta_k = 1$.

Goal: estimate $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ from N trials

Maximize the log likelihood

$$\begin{aligned}\mathcal{LL}(\boldsymbol{\theta}) &= \sum_{n=1}^{N} \log P(X_n; \boldsymbol{\theta}) \\ &= \sum_{k=1}^{K} n_k \log \theta_k\end{aligned}$$

Here n_k is the number of times we observe face k. We need to solve this problem to find the maximum likelihood estimate of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$:

$$\max_{\boldsymbol{\theta}} \sum_{k} n_k \log \theta_k$$

Maximize the log likelihood

$$\mathcal{LL}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log P(X_n; \boldsymbol{\theta})$$
$$= \sum_{k=1}^{K} n_k \log \theta_k$$

Here n_k is the number of times we observe face k. We need to solve this problem to find the maximum likelihood estimate of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$:

$$\max_{\boldsymbol{\theta}} \sum_{k} n_k \log \theta_k$$

subject to the constraint

$$\sum_{k=1}^{K} \theta_k = 1$$

Likelihood maximization

We have the constrained optimization problem (minimize the negative log likelihood):

$$\min_{\boldsymbol{\theta}} -\sum_k n_k \log \theta_k$$

$$\text{s.t.} \quad \sum_k \theta_k - 1 = 0$$

The Lagrangian is (note that we do not have inequality constraints)

$$L(\boldsymbol{\theta}, \alpha) = -\sum_k n_k \log \theta_k + \alpha(\sum_k \theta_k - 1)$$

Likelihood maximization

We have the constrained optimization problem (minimize the negative log likelihood):

$$\min_{\boldsymbol{\theta}} -\sum_k n_k \log \theta_k$$

$$\text{s.t.} \quad \sum_k \theta_k - 1 = 0$$

The Lagrangian is (note that we do not have inequality constraints)

$$L(\boldsymbol{\theta}, \alpha) = -\sum_k n_k \log \theta_k + \alpha(\sum_k \theta_k - 1)$$

Its dual problem is $\max_\alpha g(\alpha)$ where:

$$g(\alpha) = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \alpha)$$

Likelihood maximization

We now solve $\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \alpha)$ to find the dual. The optimal θ_k is obtained by

$$\frac{\partial L(\boldsymbol{\theta}, \alpha)}{\partial \theta_k} = 0 \Rightarrow \theta_k^* = \frac{n_k}{\alpha}$$

Likelihood maximization

We now solve $\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \alpha)$ to find the dual. The optimal θ_k is obtained by

$$\frac{\partial L(\boldsymbol{\theta}, \alpha)}{\partial \theta_k} = 0 \Rightarrow \theta_k^* = \frac{n_k}{\alpha}$$

Note that the optimal value of the primal variable $\boldsymbol{\theta}$ is a function of the dual variable α. We next substitute the solution back into the Lagrangian to get $g(\alpha)$:

$$\begin{aligned} g(\alpha) &= \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \alpha) \\ &= L(\boldsymbol{\theta}^*, \alpha) \\ &= -\sum_k n_k \log\left(\frac{n_k}{\alpha}\right) + \alpha\left(\sum_k \frac{n_k}{\alpha} - 1\right) \\ &= \sum_k n_k \log \alpha - \alpha + \text{CONSTANT} \end{aligned}$$

We will solve the dual next.

Solving the dual

$$g(\alpha) = \sum_k n_k \log \alpha - \alpha + \text{CONSTANT}$$

To maximize g:

$$\frac{dg(\alpha)}{d\alpha} = \sum_k \frac{n_k}{\alpha} - 1 = 0$$

$$\rightarrow \alpha^* = \sum_{k=1} n_k$$

- Remember the primal variables were expressed in terms of the dual variables $\theta_k^* = \frac{n_k}{\alpha}$.
- We have now found the optimal value for the dual variable $\alpha^* = \sum_{k=1} n_k$.
- We can substitute the optimal value for the dual variables into the expression for the primal.

$$\theta_k^* = \frac{n_k}{\sum_k n_k}$$

This makes sense: our estimate of the probability that the die will show k is simply the fraction of observations that are equal to k.

Dual formulation of SVM

Recall the primal

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \quad n \in \{1,\ldots,N\} \\ & 0 \leq \xi_n, \quad n \in \{1,\ldots,N\} \end{aligned}$$

Dual formulation of SVM

Dual is also a convex program

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\ \text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad n \in 1, \ldots, N \\ & \sum_n \alpha_n y_n = 0 \end{aligned}$$

Dual formulation of SVM

Dual is also a convex program

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\ \text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad n \in 1, \ldots, N \\ & \sum_n \alpha_n y_n = 0 \end{aligned}$$

There are N dual variable α_n, one for each constraint in the primal formulation

Kernel SVM

We replace the inner products $\phi(x_m)^{\mathbf{T}}\phi(x_n)$ with a kernel function

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n) \\ \text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad n \in 1, \ldots, N \\ & \sum_n \alpha_n y_n = 0 \end{aligned}$$

We can define a kernel function to work with nonlinear features and learn a nonlinear decision surface

Recovering solution to the primal formulation

Weights w (primal variables) relation to α (dual variables)

$$\boldsymbol{w} = \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) \leftarrow \text{ Linear combination of the input features}$$

Recovering solution to the primal formulation

Weights w (primal variables) relation to α (dual variables)

$$\boldsymbol{w} = \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) \leftarrow \text{ Linear combination of the input features}$$

Prediction on a test point x

$$h(\boldsymbol{x}) = \text{SIGN}(\boldsymbol{w}^{\text{T}} \boldsymbol{\phi}(\boldsymbol{x}) + b) = \text{SIGN}(\sum_n y_n \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b)$$

At test time it suffices to know the kernel function! This is similar to the primal and dual view of kernelized ridge regression.

Derivation of the dual

We will derive the dual formulation as the process will reveal some interesting and important properties of SVM. Particularly, what are "support vectors"?

Recipe

- Formulate the Lagrangian function that incorporates the constraints and introduces dual variables
- Minimize the Lagrangian function over the primal variables
- Substitute the primal variables for dual variables in the Lagrangian
- Maximize the Lagrangian with respect to dual variables
- Recover the solution (for the primal variables) from the dual variables

Deriving the dual for SVM

Primal SVM

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & 1 - \xi_n - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq 0 \quad n \in 1,\ldots,N \\ & -\xi_n \leq 0, \quad n \in 1,\ldots,N \end{aligned}$$

Deriving the dual for SVM

Primal SVM

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & 1 - \xi_n - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq 0 \quad n \in 1,\ldots,N \\ & -\xi_n \leq 0, \quad n \in 1,\ldots,N \end{aligned}$$

Lagrangian

$$\begin{aligned} L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ + \sum_n \alpha_n\{1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] - \xi_n\} - \sum_n \lambda_n\xi_n \end{aligned}$$

under the constraint that $\alpha_n \geq 0$ and $\lambda_n \geq 0$ (because these are lagrange multipliers corresponding to inequality constraints).

The dual problem

$$\max_{\alpha_n \geq 0, \lambda \geq 0} g(\{\alpha_n\}, \{\lambda_n\})$$

$$g(\{\alpha_n\}, \{\lambda_n\}) = \min_{\boldsymbol{w}, b, \{\xi_n\}} L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\})$$

To compute g, we need to minimize the Lagrangian with respect to the primal variables.

Minimizing the Lagrangian with respect to the primal variables

Taking derivatives with respect to the primal variables

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \mathbf{0}$$

$$\frac{\partial L}{\partial b} = \sum_n \alpha_n y_n = 0$$

$$\frac{\partial L}{\partial \xi_n} = C - \lambda_n - \alpha_n = 0$$

Minimizing the Lagrangian with respect to the primal variables

Taking derivatives with respect to the primal variables

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \mathbf{0}$$

$$\frac{\partial L}{\partial b} = \sum_n \alpha_n y_n = 0$$

$$\frac{\partial L}{\partial \xi_n} = C - \lambda_n - \alpha_n = 0$$

These equations link the primal variables and the dual variables and provide new constraints on the dual variables:

$$\boldsymbol{w} = \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

$$\sum_n \alpha_n y_n = 0$$

$$C - \lambda_n - \alpha_n = 0$$

Substitute the solution back into the Lagrangian

$$g(\{\alpha_n\},\{\lambda_n\}) = L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\})$$

Substitute the solution back into the Lagrangian

$$
\begin{aligned}
g(\{\alpha_n\},\{\lambda_n\}) &= L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) \\
&= \sum_n (C - \alpha_n - \lambda_n)\xi_n + \frac{1}{2}\| \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 + \sum_n \alpha_n \\
&+ \left(\sum_n \alpha_n y_n \right) b - \sum_n \alpha_n y_n \left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m) \right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)
\end{aligned}
$$

Substitute the solution back into the Lagrangian

$$
\begin{aligned}
g(\{\alpha_n\},\{\lambda_n\}) &= L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) \\
&= \sum_n (C - \alpha_n - \lambda_n)\xi_n + \frac{1}{2}\| \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 + \sum_n \alpha_n \\
&+ \left(\sum_n \alpha_n y_n \right) b - \sum_n \alpha_n y_n \left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m) \right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\
&= \sum_n \alpha_n + \frac{1}{2}\| \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_{m,n} \alpha_n \alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)
\end{aligned}
$$

Substitute the solution back into the Lagrangian

$$
\begin{aligned}
g(\{\alpha_n\},\{\lambda_n\}) &= L(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) \\
&= \sum_n (C - \alpha_n - \lambda_n)\xi_n + \frac{1}{2}\| \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 + \sum_n \alpha_n \\
&+ \left(\sum_n \alpha_n y_n \right) b - \sum_n \alpha_n y_n \left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m) \right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\
&= \sum_n \alpha_n + \frac{1}{2}\| \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_{m,n} \alpha_n \alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\
&= \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} \alpha_n \alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)
\end{aligned}
$$

Several terms vanish because of the constraints $\sum_n \alpha_n y_n = 0$ and $C - \lambda_n - \alpha_n = 0$.

The dual problem

Maximizing the dual under the constraints

$$\max_{\boldsymbol{\alpha}} \quad g(\{\alpha_n\}, \{\lambda_n\}) = \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad \alpha_n \geq 0, \quad n \in 1, \ldots, N$$

$$\sum_n \alpha_n y_n = 0$$

$$C - \lambda_n - \alpha_n = 0, \quad n \in 1, \ldots, N$$

$$\lambda_n \geq 0, \quad n \in 1, \ldots, N$$

The dual problem

Maximizing the dual under the constraints

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & g(\{\alpha_n\}, \{\lambda_n\}) = \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n) \\
\text{s.t.} \quad & \alpha_n \geq 0, \quad n \in 1, \ldots, N \\
& \sum_n \alpha_n y_n = 0 \\
& C - \lambda_n - \alpha_n = 0, \quad n \in 1, \ldots, N \\
& \lambda_n \geq 0, \quad n \in 1, \ldots, N
\end{aligned}$$

We can simplify as the objective function does not depend on λ_n. Specifically, we can combine the constraints involving λ_n resulting in the following inequality constraint: $\alpha_n \leq C$:

$$\begin{aligned}
C - \lambda_n - \alpha_n = 0, \, \lambda_n \geq 0 \iff & \lambda_n = C - \alpha_n \geq 0 \\
\iff & \alpha_n \leq C
\end{aligned}$$

Simplified Dual

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\
\text{s.t.} \quad & 0 \leq \alpha_n \leq C, \quad n \in 1, \ldots, N \\
& \sum_n \alpha_n y_n = 0
\end{aligned}$$

Recovering solution to the primal formulation

We already identified the primal variable $\boldsymbol{w}$ as

$$\boldsymbol{w} = \sum_n \alpha_n y_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

- From the dual, we know that $0 \leq \alpha_n \leq C$.
- If $\alpha_n = 0$, then the example n does not affect the hyperplane/decision boundary computed by SVM.
- Only those examples with $\alpha_n > 0$ affect the hyerplane/decision boundary. These examples are termed support vectors.
- When will $\alpha_n > 0$?

Complementary slackness and support vectors

At the optimal solution to both primal and dual, the following condition must hold due to the KKT (Karsh-Kuhn-Tucker) conditions:

$$\lambda_n \xi_n = 0$$
$$\alpha_n \{1 - \xi_n - y_n[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b]\} = 0$$

- At the optimum, product of lagrange multipler for a constraint and the value of the constraint equals zero

Complementary slackness and support vectors

$$\lambda_n \xi_n = 0$$

$$\alpha_n \{1 - \xi_n - y_n[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b]\} = 0$$

From the second condition, if $\alpha_n > 0$, then

$$1 - \xi_n - y_n[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b] = 0$$

What are support vectors?

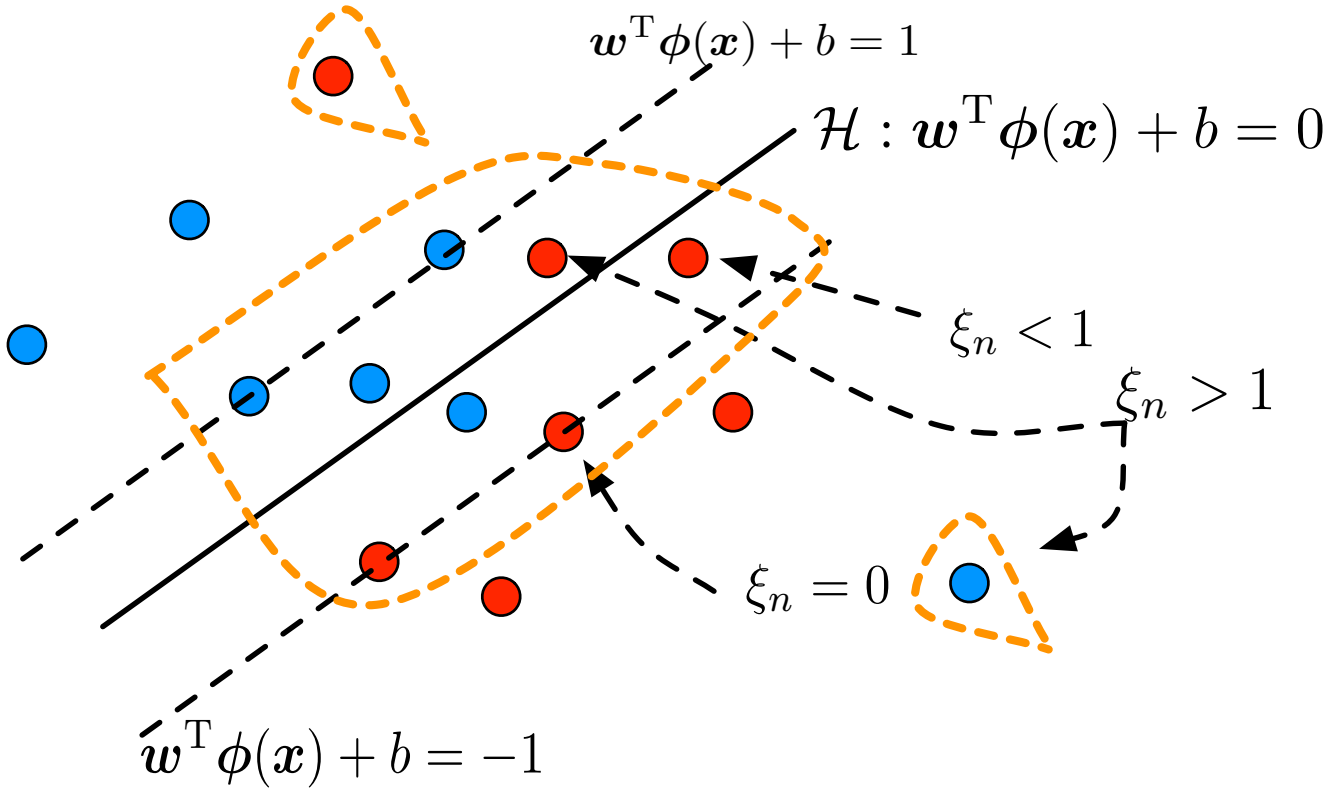
Case analysis Since, we have

$$1 - \xi_n - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]\} = 0$$

We have

- $\xi_n = 0$. This implies $y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] = 1$. They are on points that are $1/\|\boldsymbol{w}\|_2$ away from the decision boundary.
- $\xi_n < 1$. These are points that can be classified correctly but do not satisfy the large margin constraint – they have smaller distances to the decision boundary.
- $\xi_n > 1$. These are points that are misclassified.

Visualization of how training data points are categorized



Support vectors are highlighted by the dotted orange lines

Summary

- By converting the SVM problem from primal to dual, we can also kernelize the SVM.
- Led us to study optimization problems with constraints.
  - Lagrangian and lagrange multipliers (how to deal with constraints)
  - Primal vs dual
  - Complementary slackness
- The dual also allows us to understand what the "support vectors" are.