18-661 Introduction to Machine Learning

SVM – II

Spring 2019

Prof. Gauri Joshi
Important Dates

- **Homework 4**: Due on Thursday, 2/28
Course Logistics

Important Dates

- **Homework 4:** Due on Thursday, 2/28
- **Midterm exam:** In-class on Tuesday, 3/5 (< 2 weeks away).
Important Dates

• **Homework 4:** Due on Thursday, 2/28
• **Midterm exam:** In-class on Tuesday, 3/5 (< 2 weeks away).
  • Topics: MLE/MAP, Linear Regression, Logistic Regression, and SVM
Important Dates

- **Homework 4:** Due on Thursday, 2/28
- **Midterm exam:** In-class on Tuesday, 3/5 (< 2 weeks away).
  - Topics: MLE/MAP, Linear Regression, Logistic Regression, and SVM
  - Multiple-choice questions
Important Dates

- **Homework 4**: Due on Thursday, 2/28
- **Midterm exam**: In-class on Tuesday, 3/5 (< 2 weeks away).
  - Topics: MLE/MAP, Linear Regression, Logistic Regression, and SVM
  - Multiple-choice questions
  - One handwritten US-letter size cheat-sheet (two-sided) allowed – printing notes written on an IPad/equivalent is NOT permitted.
Important Dates

- **Homework 4:** Due on Thursday, 2/28
- **Midterm exam:** In-class on Tuesday, 3/5 (< 2 weeks away).
  - Topics: MLE/MAP, Linear Regression, Logistic Regression, and SVM
  - Multiple-choice questions
  - One handwritten US-letter size cheat-sheet (two-sided) allowed – printing notes written on an IPad/equivalent is NOT permitted.
  - No need for a calculator. Only pen/pencil and scratch paper are allowed.
Important Dates

- **Homework 4**: Due on Thursday, 2/28
- **Midterm exam**: In-class on Tuesday, 3/5 (< 2 weeks away).
  - Topics: MLE/MAP, Linear Regression, Logistic Regression, and SVM
  - Multiple-choice questions
  - One handwritten US-letter size cheat-sheet (two-sided) allowed – printing notes written on an IPad/equivalent is NOT permitted.
  - No need for a calculator. Only pen/pencil and scratch paper are allowed.
- **Friday 2/22**: **NO** make-up class and recitation. Want to finish SVM today so that you can start your HW4 asap
Important Dates

- **Homework 4**: Due on Thursday, 2/28
- **Midterm exam**: In-class on Tuesday, 3/5 (< 2 weeks away).
  - Topics: MLE/MAP, Linear Regression, Logistic Regression, and SVM
  - Multiple-choice questions
  - One handwritten US-letter size cheat-sheet (two-sided) allowed – printing notes written on an IPad/equivalent is NOT permitted.
  - No need for a calculator. Only pen/pencil and scratch paper are allowed.
- **Friday 2/22**: **NO** make-up class and recitation. Want to finish SVM today so that you can start your HW4 asap
- **Tuesday 2/26**: Midterm review with practice questions
Midterm: Concepts That You Should Know

This is a quick overview of the most important concepts/methods/models that you should expect to see on the midterm.

- **MLE/MAP:** how to find the likelihood of one or more observations given a system model, how to incorporate knowledge of a prior distribution, how to optimize the likelihood, loss functions
- **Linear regression:** how to formulate the linear regression optimization problem, how it relates to MLE/MAP, ridge regression, overfitting and regularization, gradient descent, bias-variance trade-off
- **Naive Bayes:** Bayes’ rule, naive classification rule, why it is naive
- **Logistic regression:** how to formulate logistic regression, how it relates to MLE, comparison to naive Bayes, sigmoid function, softmax function, cross-entropy function
- **SVMs:** hinge loss formulation, max-margin formulation, dual of the SVM problem, how to find the Lagrangian, kernel functions
1. SVM: Hinge Loss Formulation

2. SVM: Max Margin Formulation

3. Equivalence of These Two Formulations

4. Lagrange Duality and KKT conditions

5. Dual Derivation of SVMs

6. Kernel SVM
SVM: Hinge Loss Formulation
**Logistic Regression Loss: Illustration**

\[
\mathcal{L}(\mathbf{w}) = -\sum_n \{y_n \log \sigma(\mathbf{w}^\top \mathbf{x}_n) + (1 - y_n) \log[1 - \sigma(\mathbf{w}^\top \mathbf{x}_n)]\}
\]

- Loss grows approx. linearly as we move away from the boundary
- Alternative: **Hinge Loss Function**
Hinge Loss: Illustration

\[ \mathcal{L}(\mathbf{w}) = -\sum_n \{y_n \log \sigma(\mathbf{w}^\top \mathbf{x}_n) + (1 - y_n) \log[1 - \sigma(\mathbf{w}^\top \mathbf{x}_n)] \} \]

- Loss grows linearly as we move away from the boundary
- No penalty if a point is more than 1 unit from the boundary
- Makes the search for the boundary easier (as we will see later)
Hinge Loss: Mathematical Expression

\[ \mathcal{L}(\mathbf{w}) = -\sum_n \max(0, 1 - y_n (\mathbf{w}^\top \mathbf{x}_n + b)) \]

- Change of notation \( y = 0 \rightarrow y = -1 \)
- Separate the bias term \( b \) from \( \mathbf{w} \)
- Makes the mathematical expression more compact
Optimization Problem of support vector machines (SVM)

Minimizing the total hinge loss on all the training data

\[
\min_{w,b} \sum_{n} \max(0, 1 - y_n [w^T x_n + b]) + \frac{\lambda}{2} \|w\|^2
\]

Analogous to regularized least squares, as we balance between two terms (the loss and the regularizer).
Optimization Problem of support vector machines (SVM)

Minimizing the total hinge loss on all the training data

\[
\min_{w,b} \sum_n \max(0, 1 - y_n[w^\top x_n + b]) + \frac{\lambda}{2} \|w\|^2
\]

Analogous to regularized least squares, as we balance between two terms (the loss and the regularizer).

- Can solve using gradient descent to get the optimal \(w\) and \(b\)
- Gradient of the first term will be either 0, \(x_n\) or \(-x_n\) depending on \(y_n\) and \(w^\top x_n + b\)
- Much easier to compute than in logistic regression, where we need to compute the sigmoid function \(\sigma(w^\top x_n + b)\) in each iteration
SVM: Max Margin Formulation
Intuition: Where to put the decision boundary?

$\mathbf{w} \cdot \mathbf{x} + b = 0$

Idea: Find a decision boundary in the 'middle' of the two classes that:

- Perfectly classifies the training data
- Is as far away from every training point as possible
Intuition: Where to put the decision boundary?

Idea: Find a decision boundary in the ‘middle’ of the two classes that:

- Perfectly classifies the training data
- Is as far away from every training point as possible

Let us apply this intuition to build a classifier that MAXIMIZES THE MARGIN between training points and the decision boundary
How do we find the margin?

We want to derive a formula for the margin in terms of $w$, $b$, and training data $(x_n, y_n)$ for $n = 1, 2, \ldots, N$.

- Decision boundary equation is $w^\top x + b = 0$
- Divides the space in half, i.e., $w^\top x + b > 0$ and $w^\top x + b < 0$
- $w \in \mathbb{R}^d$ is a non-zero normal vector
How to find the distance of a point \( a \) to the hyperplane?

- If we define point \( a_0 \) on the line, then this distance corresponds to length of \( a - a_0 \) in direction of \( w^* = \frac{w}{||w||} \), which equals \( w^* \top (a - a_0) \).
- We know \( w^\top a_0 = -b \) since \( w^\top a_0 + b = 0 \).
- Then the distance equals \( \frac{1}{||w||}(w^\top a + b) \).
The *unsigned* distance from a point $x$ to decision boundary (hyperplane) $\mathcal{H}$ is

$$d_{\mathcal{H}}(x) = \frac{|w^\top x + b|}{\|w\|_2}$$
The *unsigned* distance from a point $x$ to decision boundary (hyperplane) $\mathcal{H}$ is

$$d_{\mathcal{H}}(x) = \frac{|w^\top x + b|}{\|w\|_2}$$

We can remove the absolute value $|\cdot|$ by exploiting the fact that the decision boundary classifies every point in the training dataset correctly.

Namely, $(w^\top x + b)$ and $x$’s label $y$ must have the same sign, so:

$$d_{\mathcal{H}}(x) = \frac{y[w^\top x + b]}{\|w\|_2}$$
Margin
Smallest distance between the hyperplane and all training points

$$\text{MARGIN}(\mathbf{w}, b) = \min_n y_n \frac{\mathbf{w}^\top \mathbf{x}_n + b}{\|\mathbf{w}\|_2}$$
Optimizing the Margin

How should we pick \((w, b)\) based on its margin?

We want a decision boundary that is as far away from all training points as possible, so we to maximize the margin!

\[
\max_{w, b} \left( \min_n \frac{y_n[w^\top x_n + b]}{\|w\|_2} \right) = \max_{w, b} \left( \frac{1}{\|w\|_2} \min_n y_n[w^\top x_n + b] \right)
\]

Only involves points near the boundary (more on this later).
Scale of $w$

Margin
Smallest distance between the hyperplane and all training points

$$
\text{MARGIN}(w, b) = \min_n \frac{y_n[w^\top x_n + b]}{\|w\|_2}
$$

Consider three hyperplanes

$$(w, b) \quad (2w, 2b) \quad (.5w, .5b)$$

Which one has the largest margin?

- The MARGIN doesn’t change if we scale $(w, b)$ by a constant $c$
- $w^\top x + b = 0$ and $(cw)^\top x + (cb) = 0$: same decision boundary!
- Can we further constrain the problem?
We can further constrain the problem by scaling \((\mathbf{w}, b)\) such that
\[
\min_{n} y_n [\mathbf{w}^\top \mathbf{x}_n + b] = 1
\]

We’ve fixed the numerator in the \text{MARGIN}(\mathbf{w}, b) equation, and we have:
\[
\text{MARGIN}(\mathbf{w}, b) = \frac{\min_{n} y_n [\mathbf{w}^\top \mathbf{x}_n + b]}{\|\mathbf{w}\|_2} = \frac{1}{\|\mathbf{w}\|_2}
\]
Hence the points closest to the decision boundary are at distance \(\frac{1}{\|\mathbf{w}\|_2}\)!
SVM: max margin formulation for separable data

Assuming separable training data, we thus want to solve:

$$\max_{w, b} \frac{1}{\|w\|_2^2} \quad \text{such that} \quad y_n[w^\top x_n + b] \geq 1, \quad \forall n$$

This is equivalent to

$$\min_{w, b} \frac{1}{2}\|w\|_2^2$$

s.t. \(y_n[w^\top x_n + b] \geq 1, \quad \forall n\)

Given our geometric intuition, SVM is called a max margin (or large margin) classifier. The constraints are called large margin constraints.
**SVM formulation for separable data**

\[
\begin{align*}
\min_{w,b} & \quad \frac{1}{2} ||w||^2_2 \\
\text{s.t.} & \quad y_n[w^T x_n + b] \geq 1, \quad \forall \ n
\end{align*}
\]

**Non-separable setting**

In practice our training data will not be separable. What issues arise with the optimization problem above when data is not separable?

- For every \( w \) there exists a training point \( x_i \) such that 
  \[
  y_i[w^T x_i + b] \leq 0
  \]

- There is no feasible \((w, b)\) as at least one of our constraints is violated!
SVM for non-separable data

Constraints in separable setting

\[ y_n [w^\top x_n + b] \geq 1, \quad \forall \ n \]

Constraints in non-separable setting

Idea: modify our constraints to account for non-separability! Specifically, we introduce slack variables \( \xi_n \geq 0 \):

\[ y_n [w^\top x_n + b] \geq 1 - \xi_n, \quad \forall \ n \]

- For “hard” training points, we can increase \( \xi_n \) until the above inequalities are met
- What does it mean when \( \xi_n \) is very large?
Soft-margin SVM formulation

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

$$\min_{w, b, \xi} \frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n$$

s.t. $y_n[w^\top x_n + b] \geq 1 - \xi_n, \ \forall \ n$

$\xi_n \geq 0, \ \forall \ n$

What is the role of $C$?

- User-defined hyperparameter
- Trades off between the two terms in our objective
- Same idea as the regularization term in ridge regression, i.e., $C = \frac{1}{\lambda}$
How to solve this problem?

\[
\min_{\mathbf{w}, b, \xi} \quad \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_n \xi_n
\]

s.t. \( y_n[\mathbf{w}^\top \mathbf{x}_n + b] \geq 1 - \xi_n, \ \forall \ n \)
\( \xi_n \geq 0, \ \forall \ n \)

- This is a \textit{convex quadratic program}: the objective function is quadratic in \( \mathbf{w} \) and linear in \( \xi \) and the constraints are linear (inequality) constraints in \( \mathbf{w}, b \) and \( \xi_n \).
- We can solve the optimization problem using general-purpose solvers, e.g., Matlab's \texttt{quadprog()} function.
The SVM solution is only determined by a subset of the training samples (as we will see in more detail in the next lecture).

These samples are called **support vectors**.

All other training points do not affect the optimal solution, i.e., if we remove the other points and construct another SVM classifier on the reduced dataset, the optimal solution will be the same.

These properties allow us to be more efficient than logistic regression or naive Bayes.
Visualization of how training data points are categorized

\[ \mathcal{H} : \mathbf{w}^T \phi(x) + b = 0 \]

\[ \mathbf{w}^T \phi(x) + b = 1 \]

\[ \mathbf{w}^T \phi(x) + b = -1 \]

Support vectors are highlighted by the dotted orange lines

Recall the constraints \( y_n[\mathbf{w}^T \mathbf{x}_n + b] \geq 1 - \xi_n \).
Example of SVM

What will be the decision boundary learnt by solving the SVM optimization problem?
What will be the decision boundary learnt by solving the SVM optimization problem?
Example of SVM

\[ x_1 - 2.5 = 0 \]

\[ y = 1 \]

\[ y = -1 \]

- Is this the right scaling of \( w \) and \( b \)?
  - No

- We need the support vectors to satisfy \( y_n (w^\top x_n + b) = 1 \), but currently \( y_n (w^\top x_n + b) = 1 \).

- Thus, we should divide \( w \) and \( b \) by 1.
Example of SVM

- Is this the right scaling of $w$ and $b$? NO
Example of SVM

- Is this the right scaling of $w$ and $b$? **NO**
- We need the support vectors to satisfy $y_n(w^\top x_n + b) = 1$, but currently $y_n(w^\top x_n + b) = 1.5$
- Thus, we should divide $w$ and $b$ by 1.5
Thus, our optimization problem will re-scale $w$ and $b$ to get this equation for the same decision boundary.
What if the data is not linearly separable?

The value of $C$ determines how much the boundary will shift.

Q: As $C$ increases how will the decision boundary move?
Example of SVM

\[
\min_{w,b,\xi} \quad \frac{1}{2}\|w\|_2^2 + C \sum_n \xi_n
\]

s.t. \quad y_n[w^\top x_n + b] \geq 1 - \xi_n, \quad \forall \ n

\xi_n \geq 0, \quad \forall \ n

• What if the data is not linearly separable?
• The value of $C$ determines how much the boundary will shift.
• Q: As $C$ increases how will be decision boundary move?
Example of SVM

\[
\begin{align*}
\min_{w, b, \xi} & \quad \frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n \\
\text{s.t.} & \quad y_n[w^T x_n + b] \geq 1 - \xi_n, \quad \forall \ n \\
& \quad \xi_n \geq 0, \quad \forall \ n
\end{align*}
\]

• What if the data is not linearly separable?
• The value of $C$ determines how much the boundary will shift.
• Q: As $C$ increases how will be decision boundary move?
• A: It will move to the left to reduce the $\xi$ corresponding to the outlier.
Outline

1. SVM: Hinge Loss Formulation

2. SVM: Max Margin Formulation

3. Equivalence of These Two Formulations

4. Lagrange Duality and KKT conditions

5. Dual Derivation of SVMs

6. Kernel SVM
Equivalence of These Two Formulations
Recovering our previous SVM formulation

Rewrite the geometric formulation as the hinge loss formulation:

\[
\min_{w, b} \sum_n \max(0, 1 - y_n [w^T x_n + b]) + \frac{\lambda}{2} \|w\|^2
\]

Here's the geometric formulation again:

\[
\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_n \xi_n \quad \text{s.t. } y_n [w^T x_n + b] \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad \forall n
\]
Recovering our previous SVM formulation

Rewrite the geometric formulation as the hinge loss formulation:

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

Here’s the geometric formulation again:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n \text{ s.t. } y_n [w^T x_n + b] \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad \forall \ n$$

Now since $y_n [w^T x_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n [w^T x_n + b]$:
Recovering our previous SVM formulation

Rewrite the geometric formulation as the hinge loss formulation:

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

Here’s the geometric formulation again:

$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2_2 + C \sum_n \xi_n \quad \text{s.t.} \quad y_n[w^T x_n + b] \geq 1 - \xi_n, \; \xi_n \geq 0, \; \forall \; n$$

Now since $y_n[w^T x_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n[w^T x_n + b]$:

$$\min_{w, b, \xi} C \sum_n \xi_n + \frac{1}{2} \|w\|^2_2 \quad \text{s.t.} \quad \max(0, 1 - y_n[w^T x_n + b]) \leq \xi_n, \; \forall \; n$$
Recovering our previous SVM formulation

Rewrite the geometric formulation as the hinge loss formulation:

\[
\min_{w, b} \sum_n \max(0, 1 - y_n [w^\top x_n + b]) + \frac{\lambda}{2} \|w\|_2^2
\]

Here’s the geometric formulation again:

\[
\min_{w, b, \xi} \frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n \quad \text{s.t.} \quad y_n [w^\top x_n + b] \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad \forall \ n
\]

Now since \( y_n [w^\top x_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n [w^\top x_n + b] \):

\[
\min_{w, b, \xi} C \sum_n \xi_n + \frac{1}{2} \|w\|_2^2 \quad \text{s.t.} \quad \max(0, 1 - y_n [w^\top x_n + b]) \leq \xi_n, \quad \forall \ n
\]

Now since the \( \xi_n \) should always be as small as possible, we obtain:
Recovering our previous SVM formulation

Rewrite the geometric formulation as the hinge loss formulation:

\[ \min_{w, b} \sum_n \max(0, 1 - y_n[w^\top x_n + b]) + \frac{\lambda}{2} \|w\|^2 \]

Here’s the geometric formulation again:

\[ \min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_n \xi_n \quad \text{s.t.} \quad y_n[w^\top x_n + b] \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad \forall \ n \]

Now since \( y_n[w^\top x_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n[w^\top x_n + b] \):

\[ \min_{w, b, \xi} \sum_n \xi_n + \frac{1}{2} \|w\|^2 \quad \text{s.t.} \quad \max(0, 1 - y_n[w^\top x_n + b]) \leq \xi_n, \quad \forall \ n \]

Now since the \( \xi_n \) should always be as small as possible, we obtain:

\[ \min_{w, b} C \sum_n \max(0, 1 - y_n[w^\top x_n + b]) + \frac{1}{2} \|w\|^2 \]
We’ve seen that the geometric formulation of SVM is equivalent to minimizing the empirical hinge loss. This explains why SVM:

1. **Is less sensitive to outliers.**
2. **Maximizes distance of training data from the boundary**
3. Generalizes well to many nonlinear models.
4. Only requires a subset of the training points.
5. Scales better with high-dimensional data.

We will need to use **duality** to show the next three properties.
Lagrange Duality and KKT conditions
What is duality?

Duality is a way of transforming a constrained optimization problem. It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

• Dual problem is always convex—easy to solve.
What is duality?

Duality is a way of transforming a constrained optimization problem. It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Dual problem is always convex—easy to solve.
- Primal and dual problems are not always equivalent.
What is duality?

Duality is a way of transforming a constrained optimization problem. It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Dual problem is always convex—easy to solve.
- Primal and dual problems are not always equivalent.
- Gives a systematic way of solving for the optimal variables.
What is duality?

Duality is a way of transforming a constrained optimization problem. It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Dual problem is always convex—easy to solve.
- Primal and dual problems are not always equivalent.
- Gives a systematic way of solving for the optimal variables.
- Dual variables tell us “how bad” constraints are.
What is duality?

Duality is a way of transforming a constrained optimization problem.
It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Dual problem is always convex—easy to solve.
- Primal and dual problems are not always equivalent.
- Gives a systematic way of solving for the optimal variables.
- Dual variables tell us “how bad” constraints are.
What is duality?

Duality is a way of transforming a constrained optimization problem. It tells us sometimes-useful information about the problem structure, and can sometimes make the problem easier to solve.

- Dual problem is always convex—easy to solve.
- Primal and dual problems are not always equivalent.
- Gives a systematic way of solving for the optimal variables.
- Dual variables tell us “how bad” constraints are.

The following material requires some advanced concepts. The main point you should understand is that we will solve the dual SVM problem in lieu of the max margin (primal) formulation.
This is the ‘primal’ problem. The generalized Lagrangian is defined as follows:
Constrained Optimization

\[
\begin{align*}
\min_x & \quad f(x) \\
\text{s.t.} & \quad g_i(x) \leq 0, \quad \forall \ i \\
& \quad h_j(x) = 0, \quad \forall \ j
\end{align*}
\]

This is the ‘primal’ problem. The generalized Lagrangian is defined as follows:

\[
L(x, \alpha, \beta) = f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x)
\]

\(\theta_P(x) = \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta)\)

\(\theta_P(x) = \begin{cases} \\
\infty & \text{if } x \text{ violates a primal constraint} \\
& \text{otherwise}
\end{cases}\)

\(\min_x \theta_P(x) = \min_x \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta)\)

\(\text{has same solution as the primal problem, which we denote as } p^*\)
Constrained Optimization

\[
\begin{aligned}
\min_{x} & \quad f(x) \\
\text{s.t.} & \quad g_i(x) \leq 0, \quad \forall \ i \\
& \quad h_i(x) = 0, \quad \forall \ j
\end{aligned}
\]

This is the ‘primal’ problem. The generalized **Lagrangian** is defined as follows:

\[
L(x, \alpha, \beta) = f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x)
\]

Consider the following function:

\[
\theta_P(x) = \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta)
\]
This is the ‘primal’ problem. The generalized Lagrangian is defined as follows:

\[
L(x, \alpha, \beta) = f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x)
\]

Consider the following function:

\[
\theta_P(x) = \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta)
\]

- If \( x \) violates a primal constraint, \( \theta_P(x) = \infty \);
- otherwise \( \theta_P(x) = f(x) \)
Constrained Optimization

\[
\begin{aligned}
\min_x & \quad f(x) \\
\text{s.t.} & \quad g_i(x) \leq 0, \quad \forall \ i \\
& \quad h_i(x) = 0, \quad \forall \ j
\end{aligned}
\]

This is the ‘primal’ problem. The generalized **Lagrangian** is defined as follows:

\[
L(x, \alpha, \beta) = f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x)
\]

Consider the following function:

\[
\theta_P(x) = \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta)
\]

- If \( x \) violates a primal constraint, \( \theta_P(x) = \infty \); otherwise \( \theta_P(x) = f(x) \)
- Thus \( \min_x \theta_P(x) = \min_x \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta) \) has same solution as the primal problem, which we denote as \( p^* \)
Primal Problem

\[ p^* = \min_x \theta_P(x) = \min_x \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta) \]

Dual Problem

Consider the function:

\[ \theta_D(\alpha, \beta) = \min_x L(x, \alpha, \beta) \]

\[ d^* = \max_{\alpha, \beta, \alpha_i \geq 0} \theta_D(\alpha, \beta) = \max_{\alpha, \beta, \alpha_i \geq 0} \min_x L(x, \alpha, \beta) \]

Relationship between primal and dual?

- \( p^* \geq d^* \) (weak duality)
- ‘min max’ of any function is always greater than the ‘max min’
- [https://en.wikipedia.org/wiki/Max%E2%80%93min_inequality](https://en.wikipedia.org/wiki/Max%E2%80%93min_inequality)
When $p^* = d^*$, we can solve the dual problem in lieu of primal problem!
Strong Duality

When \( p^* = d^* \), we can solve the dual problem in lieu of primal problem!

**Sufficient conditions for strong duality:**

- \( f \) and \( g_i \) are convex, \( h_i \) are affine (i.e., linear with offset)
- Inequality constraints are strictly ‘feasible,’ i.e., there exists some \( x \) such that \( g_i(x) < 0 \) for all \( i \)
Strong Duality

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

**Sufficient conditions for strong duality:**

- $f$ and $g_i$ are convex, $h_i$ are affine (i.e., linear with offset)
- Inequality constraints are strictly ‘feasible,’ i.e., there exists some $x$ such that $g_i(x) < 0$ for all $i$
- These conditions are all satisfied by the SVM optimization problem!

\[
\min_{w, b, \xi} \frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n \quad \text{s.t.} \quad y_n[w^T x_n + b] \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad \forall \ n
\]

convex in $w, b, \xi$

affine in $w, b, \xi$

Strictly feasible solution if $\xi_n$ are all sufficiently large.
Implications of Strong Duality

Strong duality implies that there exist $x^*, \alpha^*, \beta^*$ such that:

- $x^*$ is the solution to the primal and $\alpha^*, \beta^*$ is the solution to the dual
- $p^* = d^* = L(x^*, \alpha^*, \beta^*)$
- $x^*, \alpha^*, \beta^*$ satisfy the **KKT conditions** (which in fact are necessary and sufficient for optimality)
Implications of Strong Duality

Strong duality implies that there exist \( x^*, \alpha^*, \beta^* \) such that:

- \( x^* \) is the solution to the primal and \( \alpha^*, \beta^* \) is the solution to the dual
- \( p^* = d^* = L(x^*, \alpha^*, \beta^*) \)
- \( x^*, \alpha^*, \beta^* \) satisfy the KKT conditions (which in fact are necessary and sufficient for optimality)

The Karush-Kuhn-Tucker (KKT) conditions are:

- **Stationarity:** \( \frac{\partial L(x, \alpha^*, \beta^*)}{\partial x} \bigg|_{x^*} = 0 \). \( x^* \) is a local extremum of the Lagrangian \( L \) for fixed \( \alpha^*, \beta^* \).

  - **Feasibility:** \( g_i(x^*) \leq 0 \) and \( h_i(x^*) = 0 \) (primal) and \( \alpha_i^* \geq 0 \) (dual) for all \( i \). All primal and dual constraints are satisfied.

  - **Complementary slackness:** \( \alpha_i^* g_i(x^*) = 0 \) for all \( i \). Either the Lagrange multiplier \( \alpha_i^* \) is 0, or the corresponding constraint \( g_i(x^*) \leq 0 \) is tight (i.e., \( g_i(x^*) = 0 \)).
Consider the example of convex quadratic programming

\[
\begin{align*}
\min & \quad \frac{1}{2}x^2 \\
\text{s.t.} & \quad -x \leq 0 \\
& \quad 2x - 3 \leq 0
\end{align*}
\]
A simple example

Consider the example of convex quadratic programming

\[
\begin{align*}
\min & \quad \frac{1}{2} x^2 \\
\text{s.t.} & \quad -x \leq 0 \\
& \quad 2x - 3 \leq 0
\end{align*}
\]

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

\[
L(x, \alpha) = \frac{1}{2} x^2 + \alpha_1 (-x) + \alpha_2 (2x - 3) = \frac{1}{2} x^2 + (2\alpha_2 - \alpha_1)x - 3\alpha_2
\]

under the constraints that \( \alpha_1 \geq 0 \) and \( \alpha_2 \geq 0 \).
A simple example

Consider the example of convex quadratic programming

\[
\begin{align*}
\min & \quad \frac{1}{2} x^2 \\
\text{s.t.} & \quad -x \leq 0 \\
& \quad 2x - 3 \leq 0
\end{align*}
\]

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

\[
L(x, \alpha) = \frac{1}{2} x^2 + \alpha_1 (-x) + \alpha_2 (2x - 3) = \frac{1}{2} x^2 + (2\alpha_2 - \alpha_1) x - 3\alpha_2
\]

under the constraints that \( \alpha_1 \geq 0 \) and \( \alpha_2 \geq 0 \). Its dual problem is

\[
\max_{\alpha_1 \geq 0, \alpha_2 \geq 0} \min_x L(x, \alpha) = \max_{\alpha_1 \geq 0, \alpha_2 \geq 0} \min_x \left( \frac{1}{2} x^2 + (2\alpha_2 - \alpha_1) x - 3\alpha_2 \right)
\]
We now solve $\min_x L(x, \alpha)$. The optimal $x$ is attained by

$$\frac{\partial}{\partial x} \left( \frac{1}{2} x^2 + (2\alpha_2 - \alpha_1)x - 3\alpha_2 \right) = 0 \rightarrow x = -(2\alpha_2 - \alpha_1)$$
We now solve $\min_x L(x, \alpha)$. The optimal $x$ is attained by

$$\frac{\partial}{\partial x} \left( \frac{1}{2} x^2 + (2\alpha_2 - \alpha_1)x - 3\alpha_2 \right) = 0 \rightarrow x = -(2\alpha_2 - \alpha_1)$$

We next substitute the solution back into the Lagrangian:

$$g(\alpha) = \min_x \frac{1}{2} x^2 + (2\alpha_2 - \alpha_1)x - 3\alpha_2 = -\frac{1}{2} (2\alpha_2 - \alpha_1)^2 - 3\alpha_2$$
We now solve $\min_x L(x, \alpha)$. The optimal $x$ is attained by

$$\frac{\partial}{\partial x} \left( \frac{1}{2}x^2 + (2\alpha_2 - \alpha_1)x - 3\alpha_2 \right) = 0 \rightarrow x = -(2\alpha_2 - \alpha_1)$$

We next substitute the solution back into the Lagrangian:

$$g(\alpha) = \min_x \frac{1}{2}x^2 + (2\alpha_2 - \alpha_1)x - 3\alpha_2 = -\frac{1}{2}(2\alpha_2 - \alpha_1)^2 - 3\alpha_2$$

Our dual problem can now be simplified:

$$\max_{\alpha_1 \geq 0, \alpha_2 \geq 0} -\frac{1}{2}(2\alpha_2 - \alpha_1)^2 - 3\alpha_2$$

We will solve the dual next.
Note that,

\[ g(\alpha) = -\frac{1}{2}(2\alpha_2 - \alpha_1)^2 - 3\alpha_2 \leq 0 \]

for all \( \alpha_1 \geq 0, \alpha_2 \geq 0 \). Thus, to maximize the function, the optimal solution is

\( \alpha_1^* = 0, \quad \alpha_2^* = 0 \)

This brings us back the optimal solution of \( x \)

\[ x^* = -(2\alpha_2^* - \alpha_1^*) = 0 \]

Namely, we have arrived at the same solution as the one we guessed from the primal formulation.
Dual Derivation of SVMs
We will next derive the dual formulation for SVMs.

**Recipe**

- Formulate the generalized 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**

\[
\begin{align*}
\min_{w, b, \xi} & \quad \frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n \\
\text{s.t.} & \quad y_n [w^\top x_n + b] \geq 1 - \xi_n, \quad \forall \ n \\
& \quad \xi_n \geq 0, \quad \forall \ n
\end{align*}
\]

The constraints are equivalent to \(-\xi_n \leq 0\) and \(1 - y_n [w^\top x_n + b] - \xi \leq 0\).

**Lagrangian**

\[
L(w, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) = C \sum_n \xi_n + \frac{1}{2} \|w\|_2^2 - \sum_n \lambda_n \xi_n \\
+ \sum_n \alpha_n \{1 - y_n [w^\top x_n + b] - \xi_n\}
\]

under the constraints that \(\alpha_n \geq 0\) and \(\lambda_n \geq 0\).
Minimizing the Lagrangian

Taking derivatives with respect to the primal variables

\[
\frac{\partial L}{\partial w} = \frac{\partial}{\partial w} \left( \frac{1}{2} \|w\|_2^2 - \sum_n \alpha_n y_n w^T x_n \right) = w - \sum_n y_n \alpha_n x_n = 0
\]

\[
\frac{\partial L}{\partial b} = \frac{\partial}{\partial b} - \sum_n \alpha_n y_n b = - \sum_n \alpha_n y_n = 0
\]

\[
\frac{\partial L}{\partial \xi_n} = \frac{\partial}{\partial \xi_n} (C - \lambda_n - \alpha_n) \xi_n = C - \lambda_n - \alpha_n = 0
\]

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

\[
w = \sum_n y_n \alpha_n x_n
\]

\[
\sum_n \alpha_n y_n = 0
\]

\[
C - \lambda_n - \alpha_n = 0
\]
Rearrange the Lagrangian

\[ L(\cdot) = C \sum_n \xi_n + \frac{1}{2} \| w \|^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \{1 - y_n [w^T x_n + b] - \xi_n\} \]

where \( \alpha_n \geq 0 \) and \( \lambda_n \geq 0 \). We now know that \( w = \sum_n y_n \alpha_n x_n \).

\[ g(\{\alpha_n\}, \{\lambda_n\}) = L(w, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) \]

\[ = \sum_n (C - \alpha_n - \lambda_n) \xi_n + \frac{1}{2} \| \sum_n y_n \alpha_n x_n \|^2 + \sum_n \alpha_n \]

gather terms with \( \xi_n \)

\[ + \sum_n \alpha_n y_n \left( \sum_m y_m \alpha_m x_m \right)^T x_n - \left( \sum_n \alpha_n y_n \right) b \]

substitute for \( w \)

again substitute for \( w \)
Rearrange the Lagrangian

\[ L(\cdot) = C \sum_n \xi_n + \frac{1}{2} \|w\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \{1 - y_n [w^T x_n + b] - \xi_n\} \]

where \( \alpha_n \geq 0 \) and \( \lambda_n \geq 0 \). We now know that \( w = \sum_n y_n \alpha_n x_n \).

\[ g(\{\alpha_n\},\{\lambda_n\}) = L(w, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) \]

\[
= \sum_n (C - \alpha_n - \lambda_n) \xi_n + \frac{1}{2} \left\| \sum_n y_n \alpha_n x_n \right\|_2^2 + \sum_n \alpha_n
\]

- gather terms with \( \xi_n \)

- substitute for \( w \)

- \[
- \sum_n \alpha_n y_n \left( \sum_m y_m \alpha_m x_m \right)^T x_n - \left( \sum_n \alpha_n y_n \right) b
\]

- again substitute for \( w \)

Then, set \( \sum_n \alpha_n y_n = 0 \) and \( C - \lambda_n - \alpha_n = 0 \) and simplify to get..
The dual problem

Maximizing the dual under the constraints

\[
\max_{\alpha} \quad g(\{\alpha_n\}, \{\lambda_n\}) = \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n x_m^T x_n \\
\text{s.t.} \quad \alpha_n \geq 0, \quad \forall \ n \\
\sum_n \alpha_n y_n = 0 \\
C - \lambda_n - \alpha_n = 0, \quad \forall \ n \\
\lambda_n \geq 0, \quad \forall \ n
\]

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

\[
C - \lambda_n - \alpha_n = 0, \ \lambda_n \geq 0 \iff \lambda_n = C - \alpha_n \geq 0 \iff \alpha_n \leq C
\]
Dual formulation of SVM

Dual is also a convex quadratic program

$$\max_{\alpha} \sum_{n} \alpha_n - \frac{1}{2} \sum_{m,n} y_my_n\alpha_m\alpha_n x_m^\top x_n$$

s.t. $0 \leq \alpha_n \leq C, \quad \forall \ n$

$$\sum_{n} \alpha_n y_n = 0$$

- There are $N$ dual variables $\alpha_n$, one for each data point
- Independent of the size $d$ of $x$: **SVM scales better for high-dimensional feature.**
- May seem like a lot of optimization variables when $N$ is large, but many of the $\alpha_n$'s become zero. $\alpha_n$ is non-zero only if the $n^{th}$ point is a support vector
Why do many $\alpha_n$’s become zero?

$$\max_{\alpha} \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n x_m \top x_n$$

s.t. $0 \leq \alpha_n \leq C$, $\forall \ n$

$$\sum_n \alpha_n y_n = 0$$

• KKT complementary slackness conditions tell us:

(1) $\lambda_n \xi_n = 0$

(2) $\alpha_n \{1 - \xi_n - y_n [w \top x_n + b]\} = 0$

• (2) tells us that $\alpha_n > 0$ iff $1 - \xi_n = y_n [w \top x_n + b]$

  • If $\xi_n = 0$, then support vector is on the margin
  • Otherwise, $\xi_n > 0$ means that the point is an outlier
Visualizing the support vectors

\[ \mathcal{H} : \mathbf{w}^T \phi(\mathbf{x}) + b = 0 \]

Support vectors \((\alpha_n > 0)\) are highlighted by the dotted orange lines.

- \(\xi_n = 0\) and \(0 < \alpha_n < C\) when \(y_n[\mathbf{w}^T \mathbf{x}_n + b] = 1\).
- \(\xi_n > 0\) and \(\alpha_n = C\) if \(y_n[\mathbf{w}^T \mathbf{x}_n + b] < 1\).
Once we solve for $\alpha_n$’s, how to get $w$ and $b$?

**Recovering $w$**

\[
\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{n} \alpha_n y_n x_n
\]

Only depends on support vectors, i.e., points with $\alpha_n > 0$!
Once we solve for $\alpha_n$’s, how to get $w$ and $b$?

**Recovering $w$**

\[
\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_n \alpha_n y_n x_n
\]

Only depends on support vectors, i.e., points with $\alpha_n > 0$!

**Recovering $b$**

If $0 < \alpha_n < C$ and $y_n \in \{-1, 1\}$:

\[
y_n \left[ w^\top x_n + b \right] = 1
\]

\[
b = y_n - w^\top x_n
\]

\[
b = y_n - \sum_m \alpha_m y_m x_m^\top x_n
\]
We've seen that the geometric formulation of SVM is equivalent to minimizing the empirical hinge loss. This explains why SVM:

1. **Is less sensitive to outliers.**
2. **Maximizes distance of training data from the boundary**
3. Generalizes well to many nonlinear models.
4. **Only requires a subset of the training points.**
5. **Scales better with high-dimensional data.**
Advantages of SVM

We've seen that the geometric formulation of SVM is equivalent to minimizing the empirical hinge loss. This explains why SVM:

1. **Is less sensitive to outliers.**
2. **Maximizes distance of training data from the boundary**
3. Generalizes well to many nonlinear models.
4. **Only requires a subset of the training points.**
5. **Scales better with high-dimensional data.**

The last thing left to consider is non-linear decision boundaries, or kernel SVMs.
Kernel SVM
Non-linear basis functions in SVM

- What if the data is not linearly separable?
- We can transform the feature vector $\mathbf{x}$ using non-linear basis functions. For example,

$$
\phi(x) = \begin{bmatrix}
1 \\
x_1 \\
x_2 \\
x_1x_2 \\
x_1^2 \\
x_2^2
\end{bmatrix}
$$

- Replace $\mathbf{x}$ by $\phi(\mathbf{x})$ in both the primal and dual SVM formulations.
Primal

\[
\begin{align*}
\min_{w, b, \xi} & \quad \frac{1}{2} \|w\|^2_2 + C \sum_n \xi_n \\
\text{s.t.} & \quad y_n [w^\top \phi(x_n) + b] \geq 1 - \xi_n, \quad \forall \ n \\
& \quad \xi_n \geq 0, \quad \forall \ n
\end{align*}
\]

Dual

\[
\begin{align*}
\max_{\alpha} & \quad \sum_n \alpha_n - \frac{1}{2} \sum_{m, n} y_m y_n \alpha_m \alpha_n \phi(x_m)^\top \phi(x_n) \\
\text{s.t.} & \quad \alpha_n \geq 0, \quad \forall \ n \\
\text{s.t.} & \quad 0 \leq \alpha_n \leq C, \quad \forall \ n \\
& \quad \sum_n \alpha_n y_n = 0
\end{align*}
\]
Dual Kernel SVM

We replace the inner products $\phi(x_m)^\top \phi(x_n)$ with a kernel function

\[
\max_\alpha \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(x_m, x_n)
\]

s.t. $0 \leq \alpha_n \leq C, \forall n$

\[
\sum_n \alpha_n y_n = 0
\]
We replace the inner products $\phi(x_m)^\top \phi(x_n)$ with a kernel function

$$\max_{\alpha} \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(x_m, x_n)$$

s.t. $0 \leq \alpha_n \leq C, \quad \forall \ n$

$$\sum_n \alpha_n y_n = 0$$

- $k(x_m, x_n)$ roughly measures the similarity of $x_m$ and $x_n$. If they are similar and signs of $y_m$ and $y_n$ match, they are likely to be away from the boundary. The dual problem will try to set of the corresponding $\alpha$'s to zero.

- $k(x_m, x_n)$ is a kernel function if it is symmetric and positive-definite ($k(x, x) > 0$ for all $x > 0$).
We replace the inner products $\phi(x_m)^T \phi(x_n)$ with a kernel function

$$\max_{\alpha} \sum \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(x_m, x_n)$$

s.t. $0 \leq \alpha_n \leq C$, $\forall n$

$$\sum \alpha_n y_n = 0$$

- The dimension of $k(x_m, x_n)$ is 1 and it is independent of the dimension of the feature vector $\phi(x)$.
- We do not need to know the exact form of $\phi(x)$. E.g., if the kernel is the radial basis function $k(x_m, x_n) = \exp \left( - \|x_m - x_n\|^2 \right)$, it’s not obvious what $\phi(x)$ should be.
- This lets us define much more flexible nonlinearities.
Learning $w$ and $b$:

\[ w = \sum_n \alpha_n y_n \phi(x_n) \]

\[ b = y_n - w^\top \phi(x_n) = y_n - \sum_m \alpha_m y_m k(x_m, x_n) \]

Test Prediction:

\[ h(x) = \text{SIGN} \left( \sum_n y_n \alpha_n k(x_n, x) + b \right) \]

At test time it suffices to know the kernel function! So we really do not need to know $\phi$. 
You should know:

- How to derive the SVM dual.
- How to use the “kernel trick.”
- How to compute an SVM prediction.