Midterm Information

Midterm will be on Wednesday, 2/26 in-class.

- Closed-book except for one double-sided letter-size handwritten page of notes that you can prepare as you wish.
- We will provide formulas for relevant probability distributions.
- You will not need a calculator. Only pen/pencil and scratch paper are allowed.

Will cover all topics presented through this Wednesday in class (SVM and before).

- (1) point estimation/MLE/MAP, (2) linear regression, (3) naive Bayes, (4) logistic regression, and (5) SVMs.
- This friday's recitation will go over practice exam 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, kernel functions
1. Review of SVM Max Margin Formulation

2. SVM: Hinge Loss Formulation

3. Equivalence of These Two Formulations

4. Lagrange Duality and KKT conditions

5. Dual Derivation of SVMs
Review of 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?

\[ \mathbf{w} \cdot \mathbf{x} + b = 0 \]

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

Let us apply this intuition to build a classifier that MAXIMIZES THE MARGIN between training points and the decision boundary.
Defining the Margin

Margin
Smallest distance between the hyperplane and all training points

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

\[H : w^\top \phi(x) + b = 0\]

\[
\left| \frac{w^\top \phi(x) + b}{\|w\|_2} \right|
\]
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 MARGIN\((\mathbf{w}, b)\) equation, and we have:

\[
\text{MARGIN(} \mathbf{w}, b \text{) } = \frac{\min_n y_n [\mathbf{w}^\top \mathbf{x}_n + b]}{\|\mathbf{w}\|_2} = \frac{1}{\|\mathbf{w}\|_2}
\]

Hence the points closest to the decision boundary are at distance \(\frac{1}{\|\mathbf{w}\|_2}\)!
Assuming separable training data, we thus want to solve:

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

This is equivalent to

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_2^2 \quad \text{subject to} \quad y_n[\mathbf{w}^\top \mathbf{x}_n + b] \geq 1, \quad \forall \quad n$$

Given our geometric intuition, SVM is called a **max margin** (or large margin) classifier. The constraints are called **large margin constraints**.
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?
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. \[ y_n[w^T x_n + b] \geq 1 - \xi_n, \quad \forall \ n \]
\[ \xi_n \geq 0, \quad \forall \ n \]

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,
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?
Margin = 1.5; the decision boundary has $\mathbf{w} = [1, 0]^\top$, and $b = -2.5$. 
Example of SVM

Margin = 1.5; the decision boundary has $\mathbf{w} = [1, 0]^\top$, and $b = -2.5$.

Is this the right scaling of $\mathbf{w}$ and $b$? We need the support vectors to satisfy $y_n(\mathbf{w}^\top \mathbf{x}_n + b) = 1$. 
Margin = 1.5; the decision boundary has $w = [1, 0]^\top$, and $b = -2.5$.

Is this the right scaling of $w$ and $b$? We need the support vectors to satisfy to $y \cdot (w^\top x + b) = 1$.

Not quite. For example, for $x = [1, 0]^\top$, we have

$$y_n(w^\top x_n + b) = (-1)[1 - 2.5] = 1.5.$$
Example of SVM: scaling

Thus, our optimization problem will re-scale $w$ and $b$ to get this equation for the same decision boundary

Margin $= 1.5$; the decision boundary has $w = [2/3, 0]^\top$, and $b = -5/3$.

For example, for $x_n = [1, 0]^\top$, we have

$$y_n(w^\top x_n + b) = (-1)[2/3 - 5/3] = 1.$$
The solution to our optimization problem will be the same to the *reduced* dataset containing all the support vectors.
Example of SVM: support vectors

There can be many more data than the number of support vectors (so we can train on a smaller dataset).
Example of SVM: resilience to outliers

- Still linearly separable, but one of the orange dots is an “outlier”.
Example of SVM: resilience to outliers

- Naively applying the hard-margin SVM will result in a classifier with small margin.
Example of SVM: resilience to outliers

- Naively applying the hard-margin SVM will result in a classifier with small margin.

- So, better to use the soft-margin formulation.
Example of SVM: resilience to outliers

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

- $C = \infty$ corresponds to the hard-margin SVM;
- Due to the flexibility in $C$, SVM is also less sensitive to outliers.
• Similar reasons apply to the case when the data is not linearly separable.

• The value of $C$ determines how much the boundary will shift: trade-off of accuracy and robustness (sensitivity to outliers).
1. Review of SVM Max Margin Formulation

2. SVM: Hinge Loss Formulation

3. Equivalence of These Two Formulations

4. Lagrange Duality and KKT conditions

5. Dual Derivation of SVMs
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
**Definition**

Assume $y \in \{-1, 1\}$ and the decision rule is $h(x) = \text{SIGN}(\mathbf{w}^\top \mathbf{x})$ with $f(x) = \mathbf{w}^\top \mathbf{x} + b$,

$$
\ell_{\text{Hinge}}(f(x), y) = \begin{cases} 
0 & \text{if } yf(x) \geq 1 \\
1 - yf(x) & \text{otherwise}
\end{cases}
$$

$$
\max (\mathbf{w}^\top \mathbf{x}_n + b + 1, 0) \quad \text{max } (1 - \mathbf{w}^\top \mathbf{x}_n - b, 0)
$$

If $y = -1$

If $y = 1$
Hinge loss

**Definition**

Assume $y \in \{-1, 1\}$ and the decision rule is $h(x) = \text{SIGN}(f(x))$ with $f(x) = \mathbf{w}^\top x + b$,

$$
\ell_{\text{Hinge}}(f(x), y) = \begin{cases} 
0 & \text{if } yf(x) \geq 1 \\
1 - yf(x) & \text{otherwise}
\end{cases}
$$

**Intuition**

- No penalty if raw output, $f(x)$, 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**

$$
\ell_{\text{Hinge}}(f(x), y) = \max(0, 1 - yf(x)) = (1 - yf(x))_+
$$
Optimization Problem of support vector machines (SVM)

Minimizing the total hinge loss on all the training data

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

hinge loss for sample $n$

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

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

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

- Can solve using gradient descent to get the optimal $w$ and $b$
Minimizing the total hinge loss on all the training data

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

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 \(w^\top x_n + b\)
Optimization Problem of support vector machines (SVM)

Minimizing the total hinge loss on all the training data

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

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
1. Review of SVM Max Margin Formulation

2. SVM: Hinge Loss Formulation

3. Equivalence of These Two Formulations

4. Lagrange Duality and KKT conditions

5. Dual Derivation of SVMs
Equivalence of These Two Formulations
Recovering our previous SVM formulation

Rewrite the geometric formulation as the hinge loss formulation:

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

Here’s the geometric formulation again:

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

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

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

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

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

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

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

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:

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

Here's the geometric formulation again:

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

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

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

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

$$\min_{w,b} C \sum_{n} \max(0, 1 - y_n[w^\top x_n + b]) + \frac{1}{2} ||w||_2^2$$
Advantages of 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.
1. Review of SVM Max Margin Formulation

2. SVM: Hinge Loss Formulation

3. Equivalence of These Two Formulations

4. Lagrange Duality and KKT conditions

5. Dual Derivation of SVMs
Lagrange Duality and KKT conditions
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.
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.
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.
Constrained Optimization

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

This is the ‘primal’ problem.

Set of \( x \) that satisfy the constraints
Constrained Optimization

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

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

[Equation content]

[Explanation content]
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:

\[
L(x, \alpha, \beta) = f(x) + \sum_{i} \alpha_i g_i(x) + \sum_{j} \beta_j h_j(x)
\]
Constrained Optimization

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

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

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

Consider the following function:

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

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

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

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

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 \( \theta_P(x) = f(x) \).
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 \( p^* \)
Constrained Optimization

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

This is the ‘primal’ problem.

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

\[
\max_{\alpha > 0, \beta} L(x, \alpha, \beta)
\]
is equal to \( f(x) \) for the feasible \( x \) and infinity everywhere else

![Graph showing the set of x that satisfy the constraints]
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’

• [Link](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 $i$
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!

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

Strictly feasible solution if \( \xi_n \) are all sufficiently large.
How to find the solution $p^* = d^*$? Use KKT Conditions

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)
How to find the solution $p^* = d^*$? Use KKT Conditions

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:** $\frac{\partial L(x, \alpha^*, \beta^*)}{\partial x}|_{x^*} = 0$. $x^*$ 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

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

Can you guess the solution?
Consider the example of convex quadratic programming

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

Can you guess the solution?
\textbf{ANSWER}: \( x^* = 0 \)
A simple example

Consider the example of convex quadratic programming

$$\min \quad \frac{1}{2} x^2$$

s.t. \quad -x \leq 0

$$2x - 3 \leq 0$$
A simple example

Consider the example of convex quadratic programming

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

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

\[
\max_{\alpha_1 \geq 0, \alpha_2 \geq 0} \min_x L(x, \alpha) = \max_{\alpha_1 \geq 0, \alpha_2 \geq 0} \min_x \frac{1}{2}x^2 + (2\alpha_2 - \alpha_1)x - 3\alpha_2
\]
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 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.
Solving the dual

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.
The Diet problem

To satisfy daily nutritional requirements we need at least

- 200 units of carbs
- 50 units of protein
- 40 units of vitamins

Primal problem: How do we minimize the cost of satisfying these requirements?

<table>
<thead>
<tr>
<th>Nutrients</th>
<th>Food 1 ($2)</th>
<th>Food 2 ($5)</th>
<th>Food 3 ($15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbs</td>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Protein</td>
<td>1</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Vitamins</td>
<td>1</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>
The Diet problem

<table>
<thead>
<tr>
<th>Nutrients</th>
<th>Food 1 ($2)</th>
<th>Food 2 ($5)</th>
<th>Food 3 ($15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbs</td>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Protein</td>
<td>1</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Vitamins</td>
<td>1</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{min} \quad & 2x_1 + 5x_2 + 15x_3 \\
\text{s.t.} \quad & -20x_1 - x_2 - x_3 \leq -200 \\
& -x_1 - 30x_2 - 40x_3 \leq -50 \\
& -x_1 - 10x_2 - 5x_3 \leq -40
\end{align*}
\]

Primal Solution: \[x_1 \approx 9.84, \quad x_2 \approx 3, \quad x_3 = 0\]

Dual Solution: \[\alpha_1 = 0.07, \quad \alpha_2 = 0, \quad \alpha_3 = 0.5\]

\[\alpha_2 = 0\] means that protein requirement is easy to satisfy
The Diet problem

<table>
<thead>
<tr>
<th>Nutrients</th>
<th>Food 1 ($2)</th>
<th>Food 2 ($5)</th>
<th>Food 3 ($15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbs</td>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Protein</td>
<td>1</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Vitamins</td>
<td>1</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{min} & \quad 2x_1 + 5x_2 + 15x_3 \\
\text{s.t.} & \quad -20x_1 - x_2 - x_3 \leq -200 \\
& \quad -x_1 - 30x_2 - 40x_3 \leq -50 \\
& \quad -x_1 - 10x_2 - 5x_3 \leq -40
\end{align*}
\]

Primal Solution: \( x_1 \approx 9.84, \ x_2 \approx 3, \ x_3 = 0 \)
Dual Solution: \( \alpha_1 = 0.07, \ \alpha_2 = 0, \ \alpha_3 = 0.5 \)
### The Diet problem

<table>
<thead>
<tr>
<th>Nutrients</th>
<th>Food 1 ($2)</th>
<th>Food 2 ($5)</th>
<th>Food 3 ($15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbs</td>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Protein</td>
<td>1</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Vitamins</td>
<td>1</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{min} & \quad 2x_1 + 5x_2 + 15x_3 \\
\text{s.t.} & \quad -20x_1 - x_2 - x_3 \leq -200 \\
& \quad -x_1 - 30x_2 - 40x_3 \leq -50 \\
& \quad -x_1 - 10x_2 - 5x_3 \leq -40
\end{align*}
\]

Primal Solution: \(x_1 \approx 9.84, x_2 \approx 3, x_3 = 0\)

Dual Solution: \(\alpha_1 = 0.07, \alpha_2 = 0, \alpha_3 = 0.5\)

\(\alpha_2 = 0\) means that protein requirement is easy to satisfy
The Diet problem: Dual

<table>
<thead>
<tr>
<th>Nutrients</th>
<th>Food 1 ($2)</th>
<th>Food 2 ($5)</th>
<th>Food 3 ($15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbs</td>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Protein</td>
<td>1</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Vitamins</td>
<td>1</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

A pharmacist wants to create a diet pill to satisfy the requirements and maximize profit. \( \alpha_1, \alpha_2, \alpha_3 \) are the shadow prices of the nutrients.

\[
\begin{align*}
\max_{\alpha_1, \alpha_2, \alpha_3} & \quad 200\alpha_1 + 50\alpha_2 + 40\alpha_3 \\
\text{s.t.} & \quad 20\alpha_1 + \alpha_2 + \alpha_3 \leq 2 \\
& \quad \alpha_1 + 30\alpha_2 + 10\alpha_3 \leq 5 \\
& \quad \alpha_1 + 40\alpha_2 + 5\alpha_3 \leq 15 \\
& \quad \alpha_1, \alpha_2, \alpha_3 \geq 0
\end{align*}
\]
A pharmacist wants to create a diet pill to satisfy the requirements and maximize profit. $\alpha_1$, $\alpha_2$, $\alpha_3$ are the shadow prices of the nutrients.

$$\max_{\alpha_1, \alpha_2, \alpha_3} \quad 200\alpha_1 + 50\alpha_2 + 40\alpha_3$$

s.t. $20\alpha_1 + \alpha_2 + \alpha_3 \leq 2$

$\alpha_1 + 30\alpha_2 + 10\alpha_3 \leq 5$

$\alpha_1 + 40\alpha_2 + 5\alpha_3 \leq 15$

$\alpha_1, \alpha_2, \alpha_3 \geq 0$

Primal Solution: $x_1 \approx 9.84, x_2 \approx 3, x_3 = 0$

Dual Solution: $\alpha_1 = 0.07, \alpha_2 = 0, \alpha_3 = 0.5$
Recap

- When working with constrained optimization problems with inequality constraints, we can write down primal and dual problems.
- The dual solution is always a lower bound on the primal solution (weak duality) and it is always convex.
- The duality gap equals 0 under certain conditions (strong duality), and in such cases we can either solve the primal or dual problem.
- Strong duality (and thus the KKT conditions) hold for the SVM problem.
Dual Derivation of SVMs
Derivation of the dual

We will next derive the dual formulation for SVMs.

Recipe

1. Formulate the generalized Lagrangian function that incorporates the constraints and introduces dual variables
2. Minimize the Lagrangian function over the primal variables
3. Substitute the primal variables for dual variables in the Lagrangian
4. Maximize the Lagrangian with respect to dual variables
5. Recover the solution (for the primal variables) from the dual variables
Deriving the dual for SVM

**Primal SVM**

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

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

**Lagrangian**

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

\[
+ \sum_n \alpha_n \{1 - y_n [w^\top x_n + b] - \xi_n\}
\]

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

Taking derivatives with respect to the primal variables

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

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

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

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

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

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

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

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

\[ \text{gather terms with } \xi_n \]

\[ \text{substitute for } w \]

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

\[ \text{again substitute for } w \]
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 \( w = \sum_n y_n \alpha_n x_n \).

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

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

\[ \text{gather terms with } \xi_n \]

\[ \text{substitute for } w \]

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

\[ \text{again substitute for } w \]

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: \(\alpha_n \leq C\):

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

Dual is also a convex quadratic program

$$\max_{\alpha} \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \mathbf{x}_m \mathbf{x}_n^\top$$

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

$$\sum_n \alpha_n y_n = 0$$

- There are $N$ dual variables $\alpha_n$, one for each data point.
- Independent of the size $d$ of $\mathbf{x}$: 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:

\begin{align*}
(1) \quad & \lambda_n \xi_n = 0 \\
(2) \quad & \alpha_n \{1 - \xi_n - y_n [w^\top x_n + b]\} = 0
\end{align*}

- (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

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

- \(\xi_n = 0\) and \(0 < \alpha_n < C\) when \(y_n[w^T x_n + b] = 1\).
- \(\xi_n > 0\) and \(\alpha_n = C\) if \(y_n[w^T x_n + b] < 1\).
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 $y_n \in \{-1, 1\}$:

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

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

\[
b = y_n - \sum_m \alpha_m y_m x_m^\top x_n
\]
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.
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.