Lecture 3: Support Vector Machines

Tuo Zhao

Schools of ISYE and CSE, Georgia Tech
Suppor Vector Machines

Numerous books, papers, and tutorials......Once upon a time, SVM is considered as the ultimate recipe for classification and regression.
Support Vector Machines

SVM have faded away, but it is still an important chapter in any elementary machine learning textbook.
Margins: Intuition
Margins: Intuition

Consider a standard linear classification problem:

- Feature: $X \in \mathbb{R}^d$
- Response: $Y \in \{-1, 1\}$
- Linear Classifier: $\hat{Y} = \text{sign}(X^T\hat{\theta})$
- Our confidence is represented by $|X^T\hat{\theta}|$

Logistic Regression: $P(\hat{Y} = 1) = \frac{1}{1 + \exp(-X^T\hat{\theta})}$. 
Margins: Intuition

Consider a standard linear classification problem:

- **Feature**: $X \in \mathbb{R}^d$
- **Response**: $Y \in \{-1, 1\}$
- **Linear Classifier**: $\hat{Y} = \text{sign}(X^\top \theta)$
- **Our confidence is represented by** $|X^\top \theta|$
Margins: Intuition

Consider a standard linear classification problem:

- **Feature**: $X \in \mathbb{R}^d$
- **Response**: $Y \in \{-1, 1\}$
- **Linear Classifier**: $\hat{Y} = \text{sign}(X^\top \hat{\theta})$
- **Our confidence is represented by** $|X^\top \hat{\theta}|$

Logistic Regression: $\Pr(\hat{Y} = 1) = \frac{1}{1 + \exp(-X^\top \hat{\theta})}$. 
Margins: Intuition

Consider a standard linear classification problem:

- Feature: \( X \in \mathbb{R}^d \)
- Response: \( Y \in \{-1, 1\} \)
- Linear Classifier: \( \hat{Y} = \text{sign}(X^\top \hat{\theta}) \)
- Our confidence is represented by \( |X^\top \hat{\theta}| \)

Logistic Regression: \( \mathbb{P}(\hat{Y} = 1) = \frac{1}{1 + \exp(-X^\top \hat{\theta})} \).
Margin: Intuition

We have three testing samples: A, B, C.

Which one should we have the most confidence in classification?
Geometric Margin
Geometric Margin

- Function margin: \( \gamma_i = y_i (x_i^T \theta) \).
- Scale Invariance: \( \| \theta \|_2 = 1 \).
- Minimum functional margin:
  \[
  \gamma_{\text{min}}(\theta) = \min_i y_i (x_i^T \theta).
  \]
- Maximize the minimum margin:
  \[
  \hat{\theta} = \arg \max_{\theta} \gamma_{\text{min}}(\theta) \quad \text{subject to} \quad \| \theta \|_2 = 1.
  \]
Geometric Margin

- Function margin: $\gamma_i = y_i(x_i^\top \theta)$.
- Scale Invariance: $\|\theta\|_2 = 1$.
- Minimum functional margin:
  $$\gamma_{\min}(\theta) = \min_i y_i(x_i^\top \theta).$$
- Maximize the minimum margin:
  $$\hat{\theta} = \arg\max_{\theta} \gamma_{\min}(\theta) \text{ subject to } \|\theta\|_2 = 1.$$
Geometric Margin

- Function margin: \( \gamma_i = y_i (x_i^\top \theta) \).
- Scale Invariance: \( \|\theta\|_2 = 1 \).
- Minimum functional margin:
  \[
  \gamma_{\text{min}}(\theta) = \min_i y_i (x_i^\top \theta).
  \]
- Maximize the minimum margin:
  \[
  \hat{\theta} = \arg\max_{\theta} \gamma_{\text{min}}(\theta) \quad \text{subject to} \quad \|\theta\|_2 = 1.
  \]
Geometric Margin

- Function margin: $\gamma_i = y_i (x_i^\top \theta)$.
- Scale Invariance: $\|\theta\|_2 = 1$.
- Minimum functional margin:
  \[ \gamma_{\text{min}}(\theta) = \min_i y_i (x_i^\top \theta). \]
- Maximize the minimum margin:
  \[ \hat{\theta} = \arg \max_\theta \gamma_{\text{min}}(\theta) \quad \text{subject to} \quad \|\theta\|_2 = 1. \]
Convexification

- Maximize the minimum margin:

\[
\hat{\theta} = \arg \max_{\theta, \gamma} \gamma \\
\text{subject to } \|\theta\|_2 = 1, \ y_i(x_i^\top \theta) \geq \gamma, \ i = 1, ..., n.
\]

- \(\|\theta\|_2 = 1\) is nonconvex

- Rescale the margin

\[
\hat{\theta} = \arg \max_{\theta, \gamma} \frac{\gamma}{\|\theta\|_2} \\
\text{subject to } y_i(x_i^\top \theta) \geq \gamma, \ i = 1, ..., n.
\]
Convexification

- Maximize the minimum margin:

\[
\hat{\theta} = \arg \max_{\theta, \gamma} \gamma
\]

subject to \( \|\theta\|_2 = 1, y_i(x_i^T \theta) \geq \gamma, \ i = 1, \ldots, n. \)

- \( \|\theta\|_2 = 1 \) is nonconvex

- Rescale the margin

\[
\hat{\theta} = \arg \max_{\theta, \gamma} \frac{\gamma}{\|\theta\|_2}
\]

subject to \( y_i(x_i^T \theta) \geq \gamma, \ i = 1, \ldots, n. \)
Convexification

- Maximize the minimum margin:

\[
\hat{\theta} = \arg \max_{\theta, \gamma} \gamma
\]

subject to \( \|\theta\|_2 = 1, \ y_i(x_i^\top \theta) \geq \gamma, \ i = 1, \ldots, n. \)

- \( \|\theta\|_2 = 1 \) is nonconvex

- Rescale the margin

\[
\hat{\theta} = \arg \max_{\theta, \gamma} \frac{\gamma}{\|\theta\|_2}
\]

subject to \( y_i(x_i^\top \theta) \geq \gamma, \ i = 1, \ldots, n. \)
Convexification

- Maximize the minimum margin:

\[ \hat{\theta} = \arg \max_{\theta, \gamma} \gamma \]
\[ \text{subject to } \|\theta\|_2 = 1, \ y_i(x_i^\top \theta) \geq \gamma, \ i = 1, \ldots, n. \]

- \( \|\theta\|_2 = 1 \) is nonconvex

- Rescale the margin

\[ \hat{\theta} = \arg \max_{\theta, \gamma} \frac{\gamma}{\|\theta\|_2} \]
\[ \text{subject to } y_i(x_i^\top \theta) \geq \gamma, \ i = 1, \ldots, n. \]
Convexification

- The scale of $\gamma$ does not matter: enforce $\gamma = 1$

- Equivalent to minimizing $\|\theta\|_2$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \|\theta\|_2^2$$

subject to $1 - y_i(x_i^T \theta) \leq 0$, $i = 1, \ldots, n$.

- Quadratic Programming with Linear Constraints

- Efficient Optimization Solvers
Convexification

- The scale of $\gamma$ does not matter: enforce $\gamma = 1$

- Equivalent to minimizing $\|\theta\|_2$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \|\theta\|_2^2$$

subject to $1 - y_i(x_i^T \theta) \leq 0$, $i = 1, \ldots, n$.

- Quadratic Programming with Linear Constraints

- Efficient Optimization Solvers
Convexification

- The scale of $\gamma$ does not matter: enforce $\gamma = 1$

- Equivalent to minimizing $\|\theta\|_2$

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{2} \|\theta\|_2^2$$

subject to $1 - y_i (x_i^T \theta) \leq 0, \ i = 1, \ldots, n.$

- Quadratic Programming with Linear Constraints

- Efficient Optimization Solvers
Convexification

- The scale of $\gamma$ does not matter: enforce $\gamma = 1$
- Equivalent to minimizing $\|\theta\|_2$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \|\theta\|_2^2$$

subject to $1 - y_i(x_i^T \theta) \leq 0, \ i = 1, \ldots, n.$

- Quadratic Programming with Linear Constraints
- Efficient Optimization Solvers
Optimization Software

Existing Convex Programming Solvers:

- CPLEX: IBM ILOG CPLEX Optimization Studio.
- Programming Language: C++, Java, Python, MATLAB, R
- Modeling Language: AIMMS, AMPL, GAMS, MPL
- Free Academic Licenses.
Optimization Software

**Existing Convex Programming Solvers:**

- **CPLEX:** IBM ILOG CPLEX Optimization Studio.
- **Gurobi:** Zonghao Gu, Edward Rothberg and Robert Bixby.

  - **Programming Language:** C++, Java, Python, MATLAB, R
  - **Modeling Language:** AIMMS, AMPL, GAMS, MPL
  - **Free Academic Licenses.**
Optimization Software

Existing Convex Programming Solvers:

- CPLEX: IBM ILOG CPLEX Optimization Studio.
- Programming Language: C++, Java, Python, MATLAB, R
  - Modeling Language: AIMMS, AMPL, GAMS, MPL
- Free Academic Licenses.
Optimization Software

Existing Convex Programming Solvers:

■ CPLEX: IBM ILOG CPLEX Optimization Studio.
■ Programming Language: C++, Java, Python, MATLAB, R
■ Modeling Language: AIMMS, AMPL, GAMS, MPL
■ Free Academic Licenses.
Optimization Software

Existing Convex Programming Solvers:

- **CPLEX**: IBM ILOG CPLEX Optimization Studio.
- **Gurobi**: Zonghao Gu, Edward Rothberg and Robert Bixby.
- **Programming Language**: C++, Java, Python, MATLAB, R
- **Modeling Language**: AIMMS, AMPL, GAMS, MPL
- **Free Academic Licenses**.
Lagrangian Duality
Lagrangian Multiplier Method

- Dual variables: \( \lambda_i \geq 0, \ i = 1, \ldots, n \)

\[
L(\theta, \lambda) = \frac{1}{2} \| \theta \|_2^2 + \sum_{i=1}^{n} \lambda_i (1 - y_i (x_i^T \theta))
\]

- Min-Max problem:

\[
\min_{\theta} \max_{\lambda \geq 0} L(\theta, \lambda)
\]

- Strong duality:

\[
\min_{\theta} \max_{\lambda \geq 0} L(\theta, \lambda) = \max_{\lambda \geq 0} \min_{\theta} L(\theta, \lambda)
\]
Lagrangian Multiplier Method

- **Dual variables:** $\lambda_i \geq 0, \ i = 1, \ldots, n$

$$L(\theta, \lambda) = \frac{1}{2} \|\theta\|^2 + \sum_{i=1}^{n} \lambda_i (1 - y_i (x_i^\top \theta))$$

- **Min-Max problem:**

$$\min_{\theta} \max_{\lambda \geq 0} L(\theta, \lambda)$$

- **Strong duality:**

$$\min_{\theta} \max_{\lambda \geq 0} L(\theta, \lambda) = \max_{\lambda \geq 0} \min_{\theta} L(\theta, \lambda)$$
Lagrangian Multiplier Method

- Dual variables: $\lambda_i \geq 0$, $i = 1, \ldots, n$

$$L(\theta, \lambda) = \frac{1}{2} \| \theta \|_2^2 + \sum_{i=1}^{n} \lambda_i (1 - y_i (x_i^T \theta))$$

- Min-Max problem:

$$\min_{\theta} \max_{\lambda \geq 0} L(\theta, \lambda)$$

- Strong duality:

$$\min_{\theta} \max_{\lambda \geq 0} L(\theta, \lambda) = \max_{\lambda \geq 0} \min_{\theta} L(\theta, \lambda)$$
Penalty of Constraint Violation

- **Min-Max Problem**

\[
\min_{\theta} \max_{\lambda \geq 0} \frac{1}{2} \|\theta\|_2^2 + \sum_{i=1}^{n} \lambda_i (1 - y_i (x_i^T \theta))
\]

- Given \((1 - y_i (x_i^T \theta)) \leq 0\),

\[
\max_{\lambda_i} \lambda_i (1 - y_i (x_i^T \theta)) = 0
\]

- Given \((1 - y_i (x_i^T \theta)) > 0\),

\[
\max_{\lambda_i} \lambda_i (1 - y_i (x_i^T \theta)) = +\infty
\]
Penalty of Constraint Violation

- **Min-Max Problem**

\[
\begin{align*}
\min_{\theta} & \quad \max_{\lambda \geq 0} \frac{1}{2} \|\theta\|^2_2 + \sum_{i=1}^{n} \lambda_i (1 - y_i (x_i^T \theta)) \\
\end{align*}
\]

- **Given** \((1 - y_i (x_i^T \theta)) \leq 0\),

\[
\max_{\lambda_i} \lambda_i (1 - y_i (x_i^T \theta)) = 0
\]

- **Given** \((1 - y_i (x_i^T \theta)) > 0\),

\[
\max_{\lambda_i} \lambda_i (1 - y_i (x_i^T \theta)) = +\infty
\]
Penalty of Constraint Violation

- Min-Max Problem

\[
\min_{\theta} \max_{\lambda \geq 0} \frac{1}{2} \|\theta\|^2_2 + \sum_{i=1}^{n} \lambda_i (1 - y_i (x_i^\top \theta))
\]

- Given \((1 - y_i (x_i^\top \theta)) \leq 0\),

\[
\max_{\lambda_i} \lambda_i (1 - y_i (x_i^\top \theta)) = 0
\]

- Given \((1 - y_i (x_i^\top \theta)) > 0\),

\[
\max_{\lambda_i} \lambda_i (1 - y_i (x_i^\top \theta)) = +\infty
\]
Convex Concave Saddle Point Problem
Karush-Kuhn-Tucker Condition

- **KKT Condition:**

  \[
  \left. \frac{\partial L(\theta, \lambda)}{\partial \theta} \right|_{\theta = \hat{\theta}} = 0, \quad \left. \frac{\partial L(\hat{\theta}, \lambda)}{\partial \lambda} \right|_{\lambda = \hat{\lambda}} = 0.
  \]

- **Complementary Slackness**

  \[
  \hat{\lambda}_i = 0 \quad \Rightarrow \quad 1 - y_i x_i^T \hat{\theta} \leq 0,
  \]

  \[
  \hat{\lambda}_i \geq 0 \quad \Rightarrow \quad 1 - y_i x_i^T \hat{\theta} = 0.
  \]

- \( \hat{\lambda}_i > 0 \) — Support vector.
Karush-Kuhn-Tucker Condition

- **KKT Condition:**

\[
\frac{\partial L(\theta, \hat{\lambda})}{\partial \theta} \bigg|_{\theta=\hat{\theta}} = 0, \quad \frac{\partial L(\hat{\theta}, \lambda)}{\partial \lambda} \bigg|_{\lambda=\hat{\lambda}} = 0.
\]

- **Complementary Slackness**

\[
\hat{\lambda}_i = 0 \quad \Rightarrow \quad 1 - y_i x_i^T \hat{\theta} \leq 0,
\]

\[
\hat{\lambda}_i \geq 0 \quad \Rightarrow \quad 1 - y_i x_i^T \hat{\theta} = 0.
\]

- \( \hat{\lambda}_i > 0 \) — Support vector.
Karush-Kuhn-Tucker Condition

- **KKT Condition:**

\[
\frac{\partial L(\theta, \lambda)}{\partial \theta} \bigg|_{\theta = \hat{\theta}} = 0, \quad \frac{\partial L(\hat{\theta}, \lambda)}{\partial \lambda} \bigg|_{\lambda = \hat{\lambda}} = 0.
\]

- **Complementary Slackness**

\[
\hat{\lambda}_i = 0 \quad \Rightarrow \quad 1 - y_i x_i^\top \hat{\theta} \leq 0,
\]

\[
\hat{\lambda}_i \geq 0 \quad \Rightarrow \quad 1 - y_i x_i^\top \hat{\theta} = 0.
\]

\[
\hat{\lambda}_i > 0 \quad \text{— Support vector.}
\]
Dual Problem

■ Solve the minimization problem:

\[
\frac{\partial L(\theta, \lambda)}{\partial \theta} = 0 \Rightarrow \theta = \sum_{i=1}^{n} \lambda_i y_i x_i.
\]

■ Maximization problem:

\[
\max_{\lambda \geq 0} L\left( \sum_{i=1}^{n} \lambda_i y_i x_i, \lambda \right)
= \max_{\lambda \geq 0} \frac{1}{2} \left\| \sum_{i=1}^{n} \lambda_i y_i x_i \right\|^{2} + \sum_{i=1}^{n} \lambda_i \left( 1 - y_i \left( x_i^\top \sum_{k=1}^{n} \lambda_k y_k x_k \right) \right).
\]
Dual Problem

■ Solve the minimization problem:

\[
\frac{\partial L(\theta, \lambda)}{\partial \theta} = 0 \Rightarrow \theta = \sum_{i=1}^{n} \lambda_i y_i x_i.
\]

■ Maximization problem:

\[
\max_{\lambda \geq 0} \left( \sum_{i=1}^{n} \lambda_i y_i x_i, \lambda \right)
\]

\[
= \max_{\lambda \geq 0} \frac{1}{2} \left\| \sum_{i=1}^{n} \lambda_i y_i x_i \right\|_2^2 + \sum_{i=1}^{n} \lambda_i \left(1 - y_i \left(\mathbf{x}_i^\top \sum_{k=1}^{n} \lambda_k y_k \mathbf{x}_k \right) \right).
\]
Simplication

\[ \left\| \sum_{i=1}^{n} \lambda_i y_i x_i \right\|_2^2 = \left( \sum_{i=1}^{n} \lambda_i y_i x_i^\top \right) \left( \sum_{i=1}^{n} \lambda_i y_i x_i \right) \]
\[ = \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k x_i^\top x_k, \]
\[ \sum_{i=1}^{n} \lambda_i \left( 1 - y_i \left( x_i^\top \sum_{i=1}^{n} \lambda_i y_i x_i \right) \right) \]
\[ = \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k x_i^\top x_k. \]
Dual Problem

After simplification, we have

$$\hat{\lambda} = \arg \max_{\lambda \geq 0} \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \mathbf{x}_i^\top \mathbf{x}_k.$$ 

We define \( Q \) as \( Q_{ik} = y_i y_k \mathbf{x}_i^\top \mathbf{x}_k \).

Quadratic Minimization:

$$\hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \lambda^\top Q \lambda - \lambda^\top 1$$

subject to \( \lambda_i \geq 0, \ i = 1, ..., n \)
Let $S$ denote the set of support vectors. Given $x^*$, we predict

$$
\hat{y}^* = \text{sign}(\hat{\theta}^\top x^*) = \text{sign} \left( \sum_{i=1}^{n} \lambda_i y_i x_i^\top x^* \right)
$$

$$
= \text{sign} \left( \sum_{i=1}^{n} \lambda_i y_i x_i^\top x^* \right) = \text{sign} \left( \sum_{i \in S} \lambda_i y_i x_i^\top x^* \right)
$$
SVM with Soft Margin
Linearly Separable and Non-separable

Regularization and the non-separable case

The derivation of the SVM as presented so far assumed that the data is linearly separable. While mapping data to a high dimensional feature space via $\phi$ does generally increase the likelihood that the data is separable, we can't guarantee that it always will be so. Also, in some cases it is not clear that finding a separating hyperplane is exactly what we'd want to do, since that might be susceptible to outliers. For instance, the left figure below shows an optimal margin classifier, and when a single outlier is added in the upper-left region (right figure), it causes the decision boundary to make a dramatic swing, and the resulting classifier has a much smaller margin.

To make the algorithm work for non-linearly separable datasets as well as be less sensitive to outliers, we reformulate our optimization (using $\ell_1$ regularization) as follows:

$$\min_{\gamma, w, b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{m} \xi_i$$

subject to:

$$y(i)(w^T x(i) + b) \geq 1 - \xi_i, i = 1, \ldots, m$$

$$\xi_i \geq 0, i = 1, \ldots, m.$$
Soft Margin

- Slack variables: Given $\xi_i \geq 0$, $i = 1, \ldots, n$,

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \|\theta\|_2^2 + \mu \sum_{i=1}^{n} \xi_i$$

subject to $1 - y_i(x_i^\top \theta) - \xi_i \leq 0$, $\xi_i \geq 0$, $i = 1, \ldots, n$,

where $C > 0$ is a regularization parameter.

- Lagrangian Form: Given $\lambda \geq 0$ and $\beta \geq 0$,

$$L(\theta, \alpha, \beta) = \frac{1}{2} \|\theta\|_2^2 + C \sum_{i=1}^{n} \xi_i$$

$$\quad + \sum_{i=1}^{n} \lambda_i (1 - y_i(x_i^\top \theta) - \xi_i) - \sum_{i=1}^{n} \beta_i \xi_i$$
Soft Margin

- Slack variables: Given $\xi_i \geq 0$, $i = 1, ..., 0$,

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{2} \|\theta\|^2 + \mu \sum_{i=1}^{n} \xi_i$$

subject to $1 - y_i (x_i^\top \theta) - \xi_i \leq 0$, $\xi_i \geq 0$, $i = 1, ..., n$,

where $C > 0$ is a regularization parameter.

- Lagrangian Form: Given $\lambda \geq 0$ and $\beta \geq 0$,

$$L(\theta, \alpha, \beta) = \frac{1}{2} \|\theta\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$+ \sum_{i=1}^{n} \lambda_i (1 - y_i (x_i^\top \theta) - \xi_i) - \sum_{i=1}^{n} \beta_i \xi_i$$
Dual Formulation

- Quadratic Minimization:

\[
\hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \lambda^\top Q \lambda - \lambda^\top 1
\]
subject to \( C \geq \lambda_i \geq 0 \)

- KKT condition ⇒ Complementary Slackness

\[
\hat{\lambda}_i = 0 \quad \Rightarrow \quad y_i x_i^\top \hat{\theta} \geq 1,
\]
\[
C \geq \hat{\lambda}_i > 0 \quad \Rightarrow \quad y_i x_i^\top \hat{\theta} < 1.
\]

- \( \hat{\lambda}_i > 0 \) — Support vector.
Dual Formulation

- **Quadratic Minimization:**

  \[
  \hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \lambda^\top Q \lambda - \lambda^\top 1
  \]

  subject to \( C \geq \lambda_i \geq 0 \)

- **KKT condition ⇒ Complementary Slackness**

  \[
  \hat{\lambda}_i = 0 \quad \Rightarrow \quad y_i x_i^\top \hat{\theta} \geq 1,
  \]

  \[
  C \geq \hat{\lambda}_i > 0 \quad \Rightarrow \quad y_i x_i^\top \hat{\theta} < 1.
  \]

- \( \hat{\lambda}_i > 0 \) — Support vector.
Dual Formulation

- Quadratic Minimization:
  \[ \hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \lambda^\top Q \lambda - \lambda^\top \mathbf{1} \]
  subject to \( C \geq \lambda_i \geq 0 \)

- KKT condition ⇒ Complementary Slackness
  \[ \hat{\lambda}_i = 0 \quad \Rightarrow \quad y_i \mathbf{x}_i^\top \hat{\theta} \geq 1, \]
  \[ C \geq \hat{\lambda}_i > 0 \quad \Rightarrow \quad y_i \mathbf{x}_i^\top \hat{\theta} < 1. \]

- \( \hat{\lambda}_i > 0 \) — Support vector.
Soft Margin
Hinge Loss Function

![Graph showing the Hinge Loss Function with axes labeled $E(z)$ and $z$. The graph includes curves for different values of $z$.]
Unconstrained Formulation

SVM with soft margin is equivalent to

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^d} C \sum_{i=1}^{n} \max\{1 - y_i x_i^\top \theta, 0\} + \frac{1}{2} \|\theta\|_2^2.$$ 

- Logistic loss: $f(z) = \log(1 + \exp(-z))$
- Square Hinge Loss: $f(z) = \max\{1 - z, 0\}^2$
- Huberized Hinge Loss: $f(z) = \begin{cases} -z + \frac{1}{2} & \text{if } z \leq 0, \\ \frac{1}{2}(z - 1)^2 & \text{if } 0 < z \leq 1, \\ 0 & \text{otherwise.} \end{cases}$
- 0-1 Loss: $\ell_i(z) = 1(z \geq 0)$
Huberized Hinge Loss

![Graph showing Huberized Hinge Loss](image)

- **Left**
- **Elbow**
- **Right**

The graph illustrates the behavior of the Huberized Hinge Loss and Hinge Loss functions, highlighting different regions of the loss curve.
Nonsmooth (Stochastic) Optimization

Given a convex function $\mathcal{L}(\theta)$, the subdifferential of $\mathcal{L}(\theta)$ at $\theta$ is a set of vectors denoted by $\partial \mathcal{L}(\theta)$ such that for any $\theta, \theta' \in \mathbb{R}^d$,

$$\mathcal{L}(\theta') \geq \mathcal{L}(\theta) + \zeta^\top (\theta' - \theta) \text{ for all } \zeta \in \partial \mathcal{L}(\theta).$$

- $f(z) = |z|$: $\partial f(z) = [-1, +1]$ at $z = 0$
- $f(z) = \max\{1 - z, 0\}$: $\partial f(z) = [-1, 0]$ at $z = 1$. 

![Graph showing the subdifferential of two functions](image)
Nonsmooth (Stochastic) Optimization

We consider a nonsmooth optimization problem

$$\min_{\theta} \mathcal{L}(\theta), \text{ where } \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta)$$

- **Subgradient Algorithm:** At the \((t+1)\)-th iteration, we take

  $$\theta^{(t+1)} = \theta^{(t)} - \eta_t \zeta^{(t)}, \text{ where } \zeta^{(t)} \in \partial \mathcal{L}(\theta^{(t)}).$$

- **Stochastic Variant:** At the \((t+1)\)-th iteration, we randomly select \(i\) from \(1, \ldots, n\), and take

  $$\theta^{(t+1)} = \theta^{(t)} - \eta_t \zeta^{(t)}, \text{ where } \zeta^{(t)} \in \partial \ell_{i}(\theta^{(t)}).$$
Nonsmooth (Stochastic) Optimization

We consider a nonsmooth optimization problem

$$\min_{\theta} \mathcal{L}(\theta), \text{ where } \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta)$$

- Subgradient Algorithm: At the \((t+1)\)-th iteration, we take
  $$\theta^{(t+1)} = \theta^{(t)} - \eta_t \zeta^{(t)}, \text{ where } \zeta^{(t)} \in \partial \mathcal{L}(\theta^{(t)}).$$

- Stochastic Variant: At the \((t+1)\)-th iteration, we randomly select \(i\) from 1, ..., \(n\), and take
  $$\theta^{(t+1)} = \theta^{(t)} - \eta_t \zeta^{(t)} , \text{ where } \zeta^{(t)} \in \partial \ell_i(\theta^{(t)}).$$
Stochastic Subgradient Algorithms

How many iterations do we need?

- A sequence of decreasing step size parameters: $\eta_t \propto \frac{1}{t\mu}$
- $\|\zeta\|_2 \leq M$ for any $\zeta \in \partial L(\theta)$
- Given a pre-specified error $\varepsilon$, we need

$$T = O\left(\frac{M^2}{\mu^2 \varepsilon}\right)$$

iterations such that

$$\mathbb{E}\left\|\bar{\theta}^{(T)} - \hat{\theta}\right\|_2^2 \leq \varepsilon,$$

where $\bar{\theta}^{(T)} = \frac{2 \sum_{t=0}^{T} (t + 1) \theta^{(t)}}{(T + 1)(T + 2)}$.

- Slower than gradient descent for the square hinge loss SVM.
SVM with Kernel
Nonlinear SVM

- Feature Mapping:
  \[ \phi(x) = (\phi_1(x), \ldots, \phi_m(x))^\top \]

- SVM with Feature Mapping:
  \[ \hat{\theta} = \arg \min_{\theta \in \mathbb{R}^m} C \sum_{i=1}^{n} \max \{ 1 - y_i \phi(x_i)^\top \theta, 0 \} + \frac{1}{2} \| \theta \|^2 \]

- Dual Problem:
  \[ \hat{\lambda} = \arg \min_{C \geq \lambda \geq 0} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \phi(x_i)^\top \phi(x_k) - \lambda^\top 1 \]
Nonlinear SVM

- Feature Mapping:
  \[ \phi(x) = (\phi_1(x), \ldots, \phi_m(x))^\top \]

- SVM with Feature Mapping:
  \[ \hat{\theta} = \arg \min_{\theta \in \mathbb{R}^m} C \sum_{i=1}^{n} \max\{1 - y_i \phi(x_i)^\top \theta, 0\} + \frac{1}{2} \|\theta\|_2^2. \]

- Dual Problem:
  \[ \hat{\lambda} = \arg \min_{\lambda \geq 0} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \phi(x_i)^\top \phi(x_k) - \lambda^\top 1 \]
Nonlinear SVM

- Feature Mapping:
  \[ \phi(x) = (\phi_1(x), ..., \phi_m(x))^\top \]

- SVM with Feature Mapping:
  \[ \hat{\theta} = \arg\min_{\theta \in \mathbb{R}^m} C \sum_{i=1}^{n} \max\{1 - y_i \phi(x_i)^\top \theta, 0\} + \frac{1}{2} \|\theta\|_2^2. \]

- Dual Problem:
  \[ \hat{\lambda} = \arg\min_{C \geq \lambda \geq 0} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \phi(x_i)^\top \phi(x_k) - \lambda^\top 1 \]
Feature Mapping

\[
\phi : (x_1, x_2) \rightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)
\]

\[
\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 = 1 \rightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = 1
\]
Kernel Trick

- **Kernel Function:**
  \[ \mathcal{K}(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \]

- **Dual Problem:**
  \[
  \hat{\lambda} = \arg\min_{\lambda \geq 0} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k \mathcal{K}(x_i, x_k) - \lambda^\top 1
  \]

- **Gaussian/RBF Kernel – infinite dimensional mapping:**
  \[ \mathcal{K}(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|_2^2) \]

- **Polynomial Kernel – finite dimensional mapping:**
  \[ \mathcal{K}(x_i, x_j) = (1 + x_i^\top x_j)^r \]
Kernel Trick

- Kernel Function:
  \[ K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \]

- Dual Problem:
  \[
  \hat{\lambda} = \arg \min_{C \geq \lambda \geq 0} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k K(x_i, x_k) - \lambda^\top 1
  \]

- Gaussian/RBF Kernel – infinite dimensional mapping:
  \[ K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|_2^2) \]

- Polynomial Kernel – finite dimensional mapping:
  \[ K(x_i, x_j) = (1 + x_i^\top x_j)^r \]
Kernel Trick

- **Kernel Function:**
  \[ K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \]

- **Dual Problem:**
  \[ \hat{\lambda} = \arg \min_{C1 \geq \lambda \geq 0} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k K(x_i, x_k) - \lambda^\top 1 \]

- **Gaussian/RBF Kernel – infinite dimensional mapping:**
  \[ K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|_2^2) \]

- **Polynomial Kernel – finite dimensional mapping:**
  \[ K(x_i, x_j) = (1 + x_i^\top x_j)^r \]
Kernel Trick

- **Kernel Function:**
  \[ K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \]

- **Dual Problem:**
  \[ \hat{\lambda} = \arg\min_{C \geq \lambda \geq 0} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \lambda_i \lambda_k y_i y_k K(x_i, x_k) - \lambda^\top 1 \]

- **Gaussian/RBF Kernel – infinite dimensional mapping:**
  \[ K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \]

- **Polynomial Kernel – finite dimensional mapping:**
  \[ K(x_i, x_j) = (1 + x_i^\top x_j)^r \]
More on Kernels

- Reproducing Kernel Hilbert Space
- Representer Theorem
- Mercer Theorem

\[ K \succeq 0, \text{ where } K_{ij} = \mathcal{K}(x_i, x_j), \]

- Nonparametric SVM:

\[
\hat{f} = \arg \min_{f \in \mathcal{H}} C \sum_{i=1}^{n} \max\{1 - y_i f(x_i), 0\} + \frac{1}{2} \|f\|_{\mathcal{H}}^2.
\]
More on Kernels

- Reproducing Kernel Hilbert Space
- Representer Theorem
- Mercer Theorem

\[ K \succ 0, \quad \text{where } K_{ij} = \mathcal{K}(x_i, x_j), \]

- Nonparametric SVM:

\[ \hat{f} = \arg \min_{f \in \mathcal{H}} C \sum_{i=1}^{n} \max \{1 - y_i f(x_i), 0\} + \frac{1}{2} \|f\|_{\mathcal{H}}^2. \]
More on Kernels

- Reproducing Kernel Hilbert Space
- Representer Theorem
- Mercer Theorem

\[ K \succ 0, \quad \text{where } K_{ij} = \mathcal{K}(x_i, x_j), \]

- Nonparametric SVM:

\[
\hat{f} = \arg \min_{f \in \mathcal{H}} C \sum_{i=1}^{n} \max\{1 - y_i f(x_i), 0\} + \frac{1}{2} \|f\|_{\mathcal{H}}^2.
\]
More on Kernels

- Reproducing Kernel Hilbert Space
- Representer Theorem
- Mercer Theorem
  \[ K \succeq 0, \text{ where } K_{ij} = \mathcal{K}(x_i, x_j), \]
- Nonparametric SVM:
  \[
  \hat{f} = \arg \min_{f \in \mathcal{H}} C \sum_{i=1}^{n} \max\{1 - y_if(x_i), 0\} + \frac{1}{2} \|f\|_{\mathcal{H}}^2.
  \]
Randomized Coordinate Minimization

- Solve the dual problem:

\[ \hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \lambda^\top Q \lambda - \lambda^\top \mathbf{1} \]

subject to \( C \geq \lambda_i \geq 0 \)

- Minimize \( \lambda_j \) with \( \lambda_1, ..., \lambda_{j-1}, \lambda_{j+1}, \lambda_n \) fixed.

- At the \((t + 1)\)-th iteration, we select \( i \in \{1, ..., n\} \) and take

\[ \lambda_i^{(t+1)} = \arg \min_{\lambda_i} \frac{1}{2} Q_{ii} \lambda_i^2 + \sum_{j \neq i} Q_{ij} \lambda_i \lambda_j^{(t)} - \lambda_i \]

subject to \( C \geq \lambda_i \geq 0 \)

and \( \lambda_j^{(t+1)} = \lambda_j^{(t)} \) for all \( j \neq i \).
Randomized Coordinate Minimization

- Solve the dual problem:

\[
\hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \lambda^T Q \lambda - \lambda^T \mathbf{1}
\]
subject to \( C \geq \lambda_i \geq 0 \)

- Minimize \( \lambda_j \) with \( \lambda_1, ..., \lambda_{j-1}, \lambda_{j+1}, \lambda_n \) fixed.

- At the \((t+1)\)-th iteration, we select \( i \in \{1, ..., n\} \) and take

\[
\lambda_i^{(t+1)} = \arg \min_{\lambda_i} \frac{1}{2} Q_{ii} \lambda_i^2 + \sum_{j \neq i} Q_{ij} \lambda_i \lambda_j^{(t)} - \lambda_i
\]
subject to \( C \geq \lambda_i \geq 0 \)

and \( \lambda_j^{(t+1)} = \lambda_j^{(t)} \) for all \( j \neq i \).
Randomized Coordinate Minimization

- Solve the dual problem:
  \[
  \hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \lambda^\top Q \lambda - \lambda^\top 1 \\
  \text{subject to } C \geq \lambda_i \geq 0
  \]

- Minimize \( \lambda_j \) with \( \lambda_1, \ldots, \lambda_{j-1}, \lambda_{j+1}, \lambda_n \) fixed.

- At the \((t + 1)\)-th iteration, we select \( i \in \{1, \ldots, n\} \) and take
  \[
  \lambda_i^{(t+1)} = \arg \min_{\lambda_i} \frac{1}{2} Q_{ii} \lambda_i^2 + \sum_{j \neq i} Q_{ij} \lambda_i \lambda_j^{(t)} - \lambda_i \\
  \text{subject to } C \geq \lambda_i \geq 0
  \]
  and \( \lambda_j^{(t+1)} = \lambda_j^{(t)} \) for all \( j \neq i \).
Computational Cost per Iteration

- An analytical updating formula:
  \[
  \lambda_i^{(t+1)} = \arg\min_{C \geq \lambda_i \geq 0} \frac{1}{2} Q_{ii} \lambda_i^2 + \lambda_i \sum_{j \neq i} Q_{ij} \lambda_j^{(t)} - \lambda_i
  \]

\[
= \begin{cases} 
  C & \text{if } \tilde{\lambda}^{(t+1)} \geq C, \\
  0 & \text{if } \tilde{\lambda}^{(t+1)} \leq 0, \\
  \tilde{\lambda}^{(t+1)} & \text{otherwise,}
\end{cases}
\]

where \( \tilde{\lambda}^{(t+1)} = Q_{ii}^{-1} \left( \sum_{j \neq i} Q_{ij} \lambda_j^{(t)} - 1 \right) \).

- What is the computational cost per iteration? \( O(n) \)
- Better than gradient descent? \( O(n^2) \)
Computational Cost per Iteration

- An analytical updating formula:
  \[
  \lambda_i^{(t+1)} = \arg \min_{C \geq \lambda_i \geq 0} \frac{1}{2} Q_{ii} \lambda_i^2 + \lambda_i \sum_{j \neq i} Q_{ij} \lambda_j^{(t)} - \lambda_i
  \]
  \[
  = \begin{cases} 
  C & \text{if } \tilde{\lambda}^{(t+1)} \geq C, \\
  0 & \text{if } \tilde{\lambda}^{(t+1)} \leq 0, \\
  \tilde{\lambda}^{(t+1)} & \text{otherwise}, 
  \end{cases}
  \]
  \[
  \text{where } \tilde{\lambda}^{(t+1)} = Q_{ii}^{-1} \left( \sum_{j \neq i} Q_{ij} \lambda_j^{(t)} - 1 \right).
  \]

- What is the computational cost per iteration? \(O(n)\)

- Better than gradient descent? \(O(n^2)\)
Computational Cost per Iteration

- An analytical updating formula:
  \[
  \lambda_i^{(t+1)} = \arg \min_{C \geq \lambda_i \geq 0} \frac{1}{2} Q_{ii} \lambda_i^2 + \lambda_i \sum_{j \neq i} Q_{ij} \lambda_j^{(t)} - \lambda_i 
  \]
  
  \[
  = \begin{cases} 
  C & \text{if } \tilde{\lambda}^{(t+1)} \geq C, \\
  0 & \text{if } \tilde{\lambda}^{(t+1)} \leq 0, \\
  \tilde{\lambda}^{(t+1)} & \text{otherwise}, 
  \end{cases}
  \]

  where \( \tilde{\lambda}^{(t+1)} = Q_{ii}^{-1} \left( \sum_{j \neq i} Q_{ij} \lambda_j^{(t)} - 1 \right) \).

- What is the computational cost per iteration? \( O(n) \)

- Better than gradient descent? \( O(n^2) \)
Geometric Illustration

Optimal Solution
Coordinate Strong Smoothness

- There exists a constant $L_j$ such that for any $\lambda$ and $\lambda'$ with $\lambda_i = \lambda'_i$ for $i \neq j$, we have
  \[
  \mathcal{F}(\lambda') - \mathcal{F}(\lambda) - \nabla_j \mathcal{F}(\lambda)(\lambda'_j - \lambda_j) \leq \frac{L_j}{2} (\lambda'_j - \lambda_j)^2.
  \]

- There exists a constant $\mu$ such that for any $\lambda$ and $\lambda'$, we have
  \[
  \mathcal{F}(\lambda') - \mathcal{F}(\lambda) - \nabla \mathcal{F}(\lambda)^\top (\lambda' - \lambda) \geq \frac{\mu}{2} \| \lambda' - \lambda \|_2^2.
  \]

- Coordinate v.s. Global Strong Smoothness:
  \[
  \frac{L}{n\mu} \leq \frac{\max_j L_j}{\mu} \leq \frac{L}{\mu}.
  \]
Coordinate Strong Smoothness

- There exists a constant $L_j$ such that for any $\lambda$ and $\lambda'$ with $\lambda_i = \lambda'_i$ for $i \neq j$, we have

$$F(\lambda') - F(\lambda) - \nabla_j F(\lambda)(\lambda'_j - \lambda_j) \leq \frac{L_j}{2} (\lambda'_j - \lambda_j)^2.$$

- There exists a constant $\mu$ such that for any $\lambda$ and $\lambda'$, we have

$$F(\lambda') - F(\lambda) - \nabla F(\lambda)^\top (\lambda' - \lambda) \geq \frac{\mu}{2} \|\lambda' - \lambda\|_2^2.$$

- Coordinate v.s. Global Strong Smoothness:

$$\frac{L}{n\mu} \leq \frac{\max_j L_j}{\mu} \leq \frac{L}{\mu}.$$
Coordinate Strong Smoothness

There exists a constant $L_j$ such that for any $\lambda$ and $\lambda'$ with $\lambda_i = \lambda'_i$ for $i \neq j$, we have

$$F(\lambda') - F(\lambda) - \nabla_j F(\lambda)(\lambda'_j - \lambda_j) \leq \frac{L_j}{2} (\lambda'_j - \lambda_j)^2.$$ 

There exists a constant $\mu$ such that for any $\lambda$ and $\lambda'$, we have

$$F(\lambda') - F(\lambda) - \nabla F(\lambda)^\top (\lambda' - \lambda) \geq \frac{\mu}{2} \| \lambda' - \lambda \|^2_2.$$ 

Coordinate v.s. Global Strong Smoothness:

$$\frac{L}{n\mu} \leq \frac{\max_j L_j}{\mu} \leq \frac{L}{\mu}.$$
Iteration Complexity

How many iterations do we need?

- Given a pre-specified error $\varepsilon$, we need
  \[ T = O \left( \frac{nL_{\text{max}}}{\mu} \log \left( \frac{1}{\varepsilon} \right) \right) \]
  iterations such that
  \[ \mathbb{E} \left\| \lambda^{(T)} - \hat{\lambda} \right\|_2^2 \leq \varepsilon. \]

- Faster than the gradient descent algorithm? Yes!
Iteration Complexity

How many iterations do we need?

- Given a pre-specified error $\varepsilon$, we need
  \[ T = O \left( \frac{nL_{\max}}{\mu} \log \left( \frac{1}{\varepsilon} \right) \right) \]
  iterations such that
  \[ \mathbb{E} \left\| \lambda^{(T)} - \hat{\lambda} \right\|_2^2 \leq \varepsilon. \]

- Faster than the gradient descent algorithm? Yes!
Randomized v.s. Cyclic v.s. Shuffled

- Cyclic Order: 1, 2, ..., d, 1, 2, ..., d, ...
- Shuffled and Random Shuffled Order
- Randomized Order

Figure 1: Relative error $f(x_k)/f^\star - f(x_0)/f^\star$ v.s. iterations, for 5 methods C-CD, cyclic CGD with small stepsize $1/\max(A)$, randomly permuted CD, randomized CD and GD. Minimize $f(x) = x^T A x$, $n=100$, $A = A_c$ with $c=0$.

Next, we discuss numerical experiments for randomly generated $A$; for simplicity, we will normalize the diagonal entries of $A$ to be 1. Since different random distributions of $A$ will lead to different comparison results of various algorithms, we test many distributions and try to understand for which distributions C-CD performs well/poorly. To guarantee that $A$ is positive semidefinite, we generate a random matrix $U$ and let $A = U^T U$. We generate the entries of $U$ i.i.d. from a certain random distribution (which implies that the columns of $U$ are independent), such as $N(0,1)$ (standard Gaussian distribution), $\text{Unif}[0,1]$ (uniform $[0,1]$ distribution), log-normal distribution, etc. It turns out for most distributions C-CD is slower than R-CD and RP-CD, but for standard Gaussian distribution C-CD is better than R-CD and RP-CD.

Inspired by the numerical experiments for the example (24), we suspect that the performance of C-CD depends on how large the off-diagonal entries of $A$ are (with fixed diagonal entries). When $U_{ij} \sim N(0,1)$, $U^T U$ has small off-diagonal entries since $\mathbb{E}(A_{ij}) = 0$, $i \neq j$. When $U_{ij} \sim \text{Unif}[0,1]$, $U^T U$ has large off-diagonal entries since $\mathbb{E}((U^T U)_{ij}) = n/4$. The difference of these two scenarios lie in the fact that $\text{Unif}[0,1]$ has non-zero mean while $N(0,1)$ has zero mean. It is then interesting to compare the case of drawing entries from $\text{Unif}[0,1]$ with the case of drawing entries from $\text{Unif}[0.5,0.5]$. We will do some sort of A/B testing for each distribution: compare the zero-mean case with the non-zero mean case. To quantify the "off-diagonals over diagonals ratio", we define

$$i = \sum_{j \neq i} |A_{ij}| / A_{ii}, \quad \forall i = 1, \ldots, n; \quad \text{avg} = \frac{1}{n} \sum_i i.$$ 

We also define

$$\tau = L = \frac{\log_{10}(\text{error})}{\log_{10}(\text{error})}.$$

As we have normalized $A_{ii}$ to be 1, we can simply write

$$i = \sum_{j \neq i} |A_{ij}|, \quad \tau = \max(A) \approx 1 + \text{avg}.$$ 

In many examples we find $L$ to be close to $1 + \text{avg}$ especially when both of them are large.

Worst Case v.s. Average Case
Randomized v.s. Cyclic v.s. Shuffled

- Cyclic Order: 1, 2, ..., d, 1, 2, ..., d, ...
- Shuffled and Random Shuffled Order

Randomized Order

Worst Case v.s. Average Case
Randomized v.s. Cyclic v.s. Shuffled

- Cyclic Order: 1, 2, ..., d, 1, 2, ..., d, ...
- Shuffled and Random Shuffled Order
- Randomized Order

Worst Case v.s. Average Case
Randomized v.s. Cyclic v.s. Shuffled

- Cyclic Order: 1, 2, ..., d, 1, 2, ..., d, ...
- Shuffled and Random Shuffled Order
- Randomized Order

Worst Case v.s. Average Case
Bias Variance Tradeoff
Bias Variance Tradeoff

Given a regression model:

\[ Y = f^*(X) + \epsilon, \]

where \( \mathbb{E}\epsilon = 0 \) and \( \mathbb{E}\epsilon^2 = \sigma^2 \)

Bias Variance Decomposition:

\[
\mathbb{E}_{\epsilon,X}(Y - \hat{f}(X))^2 = \mathbb{E}_{\epsilon,X}(Y - f^*(X) + f^*(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X) + \mathbb{E}_{\epsilon|X}\hat{f}(X) - \hat{f}(X))^2 \\
= \sigma^2 + \mathbb{E}_X(f^*(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X))^2 + \mathbb{E}_X(\hat{f}(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X))^2. 
\]

Bias ↑ + Variance ↓ or Bias ↓ + Variance ↑.
Bias Variance Tradeoff

- Given a regression model:
  \[ Y = f^*(X) + \epsilon, \]
  where \( \mathbb{E}[\epsilon] = 0 \) and \( \mathbb{E}[\epsilon^2] = \sigma^2 \)

- Bias Variance Decomposition:
  \[
  \mathbb{E}_{\epsilon,X}(Y - \hat{f}(X))^2 \\
  = \mathbb{E}_{\epsilon,X}(Y - f^*(X) + f^*(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X) + \mathbb{E}_{\epsilon|X}\hat{f}(X) - \hat{f}(X))^2 \\
  = \sigma^2 + \mathbb{E}_X(f^*(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X))^2 + \mathbb{E}_X(f(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X))^2. 
  \]

- Bias \( \uparrow \) + Variance \( \downarrow \) or Bias \( \downarrow \) + Variance \( \uparrow \).
Bias Variance Tradeoff

- Given a regression model:
  \[ Y = f^*(X) + \epsilon, \]
  where \( \mathbb{E}\epsilon = 0 \) and \( \mathbb{E}\epsilon^2 = \sigma^2 \)

- Bias Variance Decomposition:
  \[
  \mathbb{E}_{\epsilon,X}(Y - \hat{f}(X))^2 \\
  = \mathbb{E}_{\epsilon,X}(Y - f^*(X) + f^*(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X) + \mathbb{E}_{\epsilon|X}\hat{f}(X) - \hat{f}(X))^2 \\
  = \sigma^2 + \mathbb{E}_X(f^*(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X))^2 + \mathbb{E}_X(\hat{f}(X) - \mathbb{E}_{\epsilon|X}\hat{f}(X))^2.
  \]

- Bias \uparrow + Variance \downarrow or Bias \downarrow + Variance \uparrow.
Bias Variance Tradeoff

![Diagram showing the relationship between error, variance, and bias squared with respect to model complexity. The optimum model complexity is where total error is minimized.]
Overfitting
Optimal Model Complexity

Simultaneously control bias and variance!

Next Lecture: Model Selection and Regularization