COMPSCI 689
Lecture 2: Linear Regression

Brendan O'Connor

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Outline

A definition of machine learning

Mitchell (1997): "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Substitute "training data D" for "experience E."

General Supervised Learning Notation

- Input Space: $\mathcal{X}$
- Output Space: $\mathcal{Y}$
- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- Prediction Function: $f : \mathcal{X} \to \mathcal{Y}$

Outline

The Supervised Learning Problem

The Supervised Learning Problem

Given a *data set* consisting of a collection of input-output tuples $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n) | \mathbf{x}_n \in \mathcal{X}, \mathbf{y}_n \in \mathcal{Y}, 1 \leq n \leq N\}$, select the best prediction function $f : \mathcal{X} \to \mathcal{Y}$.

Note: A data set is not a mathematical set. It is a collection of elements that allows repetition.

Prediction Loss Functions

Prediction Loss Function: A prediction loss function $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a real-valued function that is bounded below (typically at 0), and that satisfies $L(\mathbf{y}, \mathbf{y}) \leq L(\mathbf{y}, \mathbf{y}')$ for all $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$.

Examples:

- Squared Loss: $L_{sqr}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_2^2 = \sum_{k=1}^{K} (\mathbf{y}_k - \mathbf{y}'_k)^2$
- Absolute Loss: $L_{abs}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_1 = \sum_{k=1}^{K} |\mathbf{y}_k - \mathbf{y}'_k|$
- 0/1 Loss: $L_{01}(\mathbf{y}, \mathbf{y}') = [\mathbf{y} \neq \mathbf{y}']$

Given a prediction loss function L, an instance $(\mathbf{x}, \mathbf{y})$, and a prediction function f, we compute the loss of f on $(\mathbf{x}, \mathbf{y})$ as $L(\mathbf{y}, f(\mathbf{x}))$.

Do we now have enough information to select the optimal f given a data set $\mathcal{D}$? Restrict f!

Prediction Function Models

- In general in supervised learning, we do not attempt to identify the best function f from the set of all possible functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.

Prediction Function Models

- In general in supervised learning, we do not attempt to identify the best function f from the set of all possible functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.
- Instead, we specify a specific set of functions $\mathcal{F}$ and select from that set.

Prediction Function Models

- In general in supervised learning, we do not attempt to identify the best function f from the set of all possible functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.
- Instead, we specify a specific set of functions $\mathcal{F}$ and select from that set.
- We will refer to the set $\mathcal{F}$ as a *prediction function model* or just a *model*.

Prediction Function Models

- In general in supervised learning, we do not attempt to identify the best function f from the set of all possible functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.
- Instead, we specify a specific set of functions $\mathcal{F}$ and select from that set.
- We will refer to the set $\mathcal{F}$ as a *prediction function model* or just a *model*.
- The set $\mathcal{F}$ can be finite, but it is more typically uncountably infinite.

Prediction Function Models

- In general in supervised learning, we do not attempt to identify the best function f from the set of all possible functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.
- Instead, we specify a specific set of functions $\mathcal{F}$ and select from that set.
- We will refer to the set $\mathcal{F}$ as a *prediction function model* or just a *model*.
- The set $\mathcal{F}$ can be finite, but it is more typically uncountably infinite.

Prediction Function Models

- In general in supervised learning, we do not attempt to identify the best function f from the set of all possible functions mapping from $\mathcal{X}$ to $\mathcal{Y}$.
- Instead, we specify a specific set of functions $\mathcal{F}$ and select from that set.
- We will refer to the set $\mathcal{F}$ as a *prediction function model* or just a *model*.
- The set $\mathcal{F}$ can be finite, but it is more typically uncountably infinite.

Do we now have enough information to select the optimal f given a data set $\mathcal{D}$?

Empirical Risk Minimization

Let $\mathcal{F}$ be a set of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$ (e.g., a prediction function model). The principle of Empirical Risk Minimization (ERM) states that we should select the function f from the set $\mathcal{F}$ that *minimizes the average of the prediction loss* $L(\mathbf{y}_n, f(\mathbf{x}_n))$ computed over the data set $\mathcal{D}$, also known as the empirical risk $R(f, \mathcal{D})$:

Empirical Risk Minimization

Let $\mathcal{F}$ be a set of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$ (e.g., a prediction function model). The principle of Empirical Risk Minimization (ERM) states that we should select the function f from the set $\mathcal{F}$ that *minimizes the average of the prediction loss* $L(\mathbf{y}_n, f(\mathbf{x}_n))$ computed over the data set $\mathcal{D}$, also known as the empirical risk $R(f, \mathcal{D})$:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} R(f, \mathcal{D})$$

Empirical Risk Minimization

Let $\mathcal{F}$ be a set of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$ (e.g., a prediction function model). The principle of Empirical Risk Minimization (ERM) states that we should select the function f from the set $\mathcal{F}$ that *minimizes the average of the prediction loss* $L(\mathbf{y}_n, f(\mathbf{x}_n))$ computed over the data set $\mathcal{D}$, also known as the empirical risk $R(f, \mathcal{D})$:

$$\hat{f} = \underset{f \in \mathcal{F}}{\arg\min}\, R(f, \mathcal{D})$$

$$R(f, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} L(\mathbf{y}_n, f(\mathbf{x}_n))$$

$- \log p(y|x) = -\log \sum_i \log p(y_i|x_i) = NLL$

Supervised Learning by ERM

ERM provides our first general framework for supervised learning:

1. The supervised learning task defines $\mathcal{X}$ and $\mathcal{Y}$.

Supervised Learning by ERM

ERM provides our first general framework for supervised learning:

1. The supervised learning task defines $\mathcal{X}$ and $\mathcal{Y}$.
2. We collect or obtain a data set $\mathcal{D}$.

Supervised Learning by ERM

ERM provides our first general framework for supervised learning:

1. The supervised learning task defines $\mathcal{X}$ and $\mathcal{Y}$.
2. We collect or obtain a data set $\mathcal{D}$.
3. We choose a prediction loss function L as the performance measure.

Supervised Learning by ERM

ERM provides our first general framework for supervised learning:

1. The supervised learning task defines $\mathcal{X}$ and $\mathcal{Y}$.
2. We collect or obtain a data set $\mathcal{D}$.
3. We choose a prediction loss function L as the performance measure.
4. We choose the space of prediction functions $\mathcal{F}$.

Supervised Learning by ERM

ERM provides our first general framework for supervised learning:

1. The supervised learning task defines $\mathcal{X}$ and $\mathcal{Y}$.
2. We collect or obtain a data set $\mathcal{D}$.
3. We choose a prediction loss function L as the performance measure.
4. We choose the space of prediction functions $\mathcal{F}$.
5. We select the function $\hat{f}$ from $\mathcal{F}$ that minimizes the empirical risk $R(f, \mathcal{D})$.

Outline

Linear Regression

- Consider the classical regression setting in which $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}$.

Linear Regression

- Consider the classical regression setting in which $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}$.
- In this setting, the data set $\mathcal{D}$ consists of input vectors $\mathbf{x}_n$ and scalar output values y_n.

Linear Regression

- Consider the classical regression setting in which $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}$.
- In this setting, the data set $\mathcal{D}$ consists of input vectors $\mathbf{x}_n$ and scalar output values y_n.
- We will assume that $\mathbf{x}_n$ is a row vector, and thus has shape $(1, D)$.

Linear Regression

- Consider the classical regression setting in which $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}$.
- In this setting, the data set $\mathcal{D}$ consists of input vectors $\mathbf{x}_n$ and scalar output values y_n.
- We will assume that $\mathbf{x}_n$ is a row vector, and thus has shape $(1, D)$.
- In linear regression, we choose as our model $\mathcal{F}$ the space of all linear functions of $\mathbf{x}$.

Linear Regression

- Consider the classical regression setting in which $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}$.
- In this setting, the data set $\mathcal{D}$ consists of input vectors $\mathbf{x}_n$ and scalar output values y_n.
- We will assume that $\mathbf{x}_n$ is a row vector, and thus has shape $(1, D)$.
- In linear regression, we choose as our model $\mathcal{F}$ the space of all linear functions of $\mathbf{x}$.
- The most commonly used prediction loss function in this setting is the squared loss $L_{sqr}(y, y') = (y - y')^2$.

The Linear Regression Model

- To apply ERM to the linear regression model, we need a mathematical description of the set $\mathcal{F}$ of all linear functions of $\mathbf{x}$.

The Linear Regression Model

- To apply ERM to the linear regression model, we need a mathematical description of the set $\mathcal{F}$ of all linear functions of $\mathbf{x}$.
- First, define the parameter space $\Theta = \{[\mathbf{w}; b] | \mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}\}$.

The Linear Regression Model

- To apply ERM to the linear regression model, we need a mathematical description of the set $\mathcal{F}$ of all linear functions of $\mathbf{x}$.
- First, define the parameter space $\Theta = \{[\mathbf{w}; b] | \mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}\}$.
- An element $\theta \in \Theta$ is a vector $\theta = [\mathbf{w}; b]$, referred to as the *model parameters*.

The Linear Regression Model

- To apply ERM to the linear regression model, we need a mathematical description of the set $\mathcal{F}$ of all linear functions of $\mathbf{x}$.
- First, define the parameter space $\Theta = \{[\mathbf{w}; b] | \mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}\}$.
- An element $\theta \in \Theta$ is a vector $\theta = [\mathbf{w}; b]$, referred to as the *model parameters*.
- $\mathbf{w}$ is a column vector with shape $(D, 1)$ called the *weights* or *coefficients* and b is a real scalar ~~called~~ the *bias* or *intercept*.

The Linear Regression Model

- To apply ERM to the linear regression model, we need a mathematical description of the set $\mathcal{F}$ of all linear functions of $\mathbf{x}$.
- First, define the parameter space $\Theta = \{[\mathbf{w}; b] | \mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}\}$.
- An element $\theta \in \Theta$ is a vector $\theta = [\mathbf{w}; b]$, referred to as the *model parameters*.
- $\mathbf{w}$ is a column vector with shape $(D, 1)$ called the *weights* or *coefficients* and b is a real scalar called the *bias* or *intercept*.
- Now define the set of parametric functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b | \theta \in \Theta\}$.

The Linear Regression Model

- To apply ERM to the linear regression model, we need a mathematical description of the set $\mathcal{F}$ of all linear functions of $\mathbf{x}$.
- First, define the parameter space $\Theta = \{[\mathbf{w}; b] | \mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}\}$.
- An element $\theta \in \Theta$ is a vector $\theta = [\mathbf{w}; b]$, referred to as the *model parameters*.
- $\mathbf{w}$ is a column vector with shape $(D, 1)$ called the *weights* or *coefficients* and b is a real scalar called the *bias* or *intercept*.
- Now define the set of parametric functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b | \theta \in \Theta\}$.

$\Theta = \mathbb{R}^D \times \mathbb{R} = \mathbb{R}^{D+1}$

- This parametric space of functions is the linear regression prediction function model.

ERM for Linear Regression

Given the choice of the squared prediction loss $L_{sqr}(y, y') = (y - y')^2$ and the space of prediction functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b | \theta \in \Theta\}$, we can now apply ERM to define the optimal prediction function:

ERM for Linear Regression

Given the choice of the squared prediction loss $L_{sqr}(y, y') = (y - y')^2$ and the space of prediction functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b | \theta \in \Theta\}$, we can now apply ERM to define the optimal prediction function:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} R(f, \mathcal{D}) \quad \rightarrow \quad \hat{\theta} = \arg\min_{\theta \in \Theta} R(f_\theta, \mathcal{D})$$

ERM for Linear Regression

Given the choice of the squared prediction loss $L_{sqr}(y, y') = (y - y')^2$ and the space of prediction functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b | \theta \in \Theta\}$, we can now apply ERM to define the optimal prediction function:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} R(f, \mathcal{D}) \quad \rightarrow \quad \hat{\theta} = \arg\min_{\theta \in \Theta} R(f_\theta, \mathcal{D})$$

$$R(f, \mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - f(\mathbf{x}_n))^2 \quad \rightarrow \quad R(f_\theta, \mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - (\mathbf{x}_n\mathbf{w} + b))^2$$

ERM for Linear Regression

Given the choice of the squared prediction loss $L_{sqr}(y, y') = (y - y')^2$ and the space of prediction functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b | \theta \in \Theta\}$, we can now apply ERM to define the optimal prediction function:

$$\hat{\theta} = \underset{\theta \in \Theta}{\arg\min}\, R(f_\theta, \mathcal{D})$$

$$R(f_\theta, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{y}_n - (\mathbf{x}_n \mathbf{w} + b))^2$$

Question: How do we actually find the model parameters θ that minimize the empirical risk defined above?

Optimization Theory for ERM

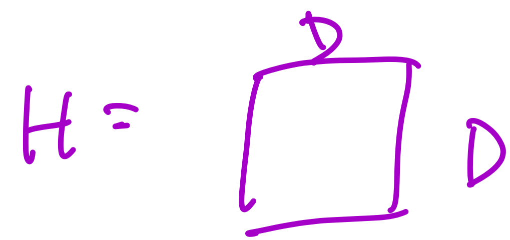
Key optimization definitions for minimizing empirical risk:

- **Gradient**: The gradient $\nabla R(f_\theta, \mathcal{D})$ of the empirical risk is the vector of partial derivatives of $R(f_\theta, \mathcal{D})$ with respect to each of the model parameters: $[\nabla R(f_\theta, \mathcal{D})]_i = \frac{\partial}{\partial \theta_i} R(f_\theta, \mathcal{D})$. $\forall i \in 1..D$

Optimization Theory for ERM

Key optimization definitions for minimizing empirical risk:

- **Gradient**: The gradient $\nabla R(f_\theta, \mathcal{D})$ of the empirical risk is the vector of partial derivatives of $R(f_\theta, \mathcal{D})$ with respect to each of the model parameters: $[\nabla R(f_\theta, \mathcal{D})]_i = \frac{\partial}{\partial \theta_i} R(f_\theta, \mathcal{D})$.
- **Hessian**: The hessian $\nabla^2 R(f_\theta, \mathcal{D})$ of the empirical risk $R(f_\theta, \mathcal{D})$ is the matrix of mixed partial derivatives of $R(f_\theta, \mathcal{D})$ with respect to each pair of model parameters:
$[\nabla^2 R(f_\theta, \mathcal{D})]_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} R(f_\theta, \mathcal{D})$.



Optimization Theory for ERM

Key optimization definitions for minimizing empirical risk:

- **Gradient**: The gradient $\nabla R(f_\theta, \mathcal{D})$ of the empirical risk is the vector of partial derivatives of $R(f_\theta, \mathcal{D})$ with respect to each of the model parameters: $[\nabla R(f_\theta, \mathcal{D})]_i = \frac{\partial}{\partial \theta_i} R(f_\theta, \mathcal{D})$.
- **Hessian**: The hessian $\nabla^2 R(f_\theta, \mathcal{D})$ of the empirical risk $R(f_\theta, \mathcal{D})$ is the matrix of mixed partial derivatives of $R(f_\theta, \mathcal{D})$ with respect to each pair of model parameters: $[\nabla^2 R(f_\theta, \mathcal{D})]_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} R(f_\theta, \mathcal{D})$.
- **Local Minimizer**: θ is a local minimizer of $R(f_\theta, \mathcal{D})$ if and only if $\nabla R(f_\theta, \mathcal{D}) = 0$ and the Hessian of $R(f_\theta, \mathcal{D})$ at θ is positive semi-definite.

$x^T H x \geq 0 \quad \forall x$

Optimization Theory for ERM

Key optimization definitions for minimizing empirical risk:

- **Gradient**: The gradient $\nabla R(f_\theta, \mathcal{D})$ of the empirical risk is the vector of partial derivatives of $R(f_\theta, \mathcal{D})$ with respect to each of the model parameters: $[\nabla R(f_\theta, \mathcal{D})]_i = \frac{\partial}{\partial \theta_i} R(f_\theta, \mathcal{D})$.
- **Hessian**: The hessian $\nabla^2 R(f_\theta, \mathcal{D})$ of the empirical risk $R(f_\theta, \mathcal{D})$ is the matrix of mixed partial derivatives of $R(f_\theta, \mathcal{D})$ with respect to each pair of model parameters: $[\nabla^2 R(f_\theta, \mathcal{D})]_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} R(f_\theta, \mathcal{D})$.
- **Local Minimizer**: θ is a local minimizer of $R(f_\theta, \mathcal{D})$ if and only if $\nabla R(f_\theta, \mathcal{D}) = 0$ and the Hessian of $R(f_\theta, \mathcal{D})$ at θ is positive semi-definite.
- **Global Minimizer**: θ is a global minimizer of $R(f_\theta, \mathcal{D})$ if θ is a local minimizer of $R(f_\theta, \mathcal{D})$ and $R(f_\theta, \mathcal{D}) \leq R(f_{\theta'}, \mathcal{D})$ for all $\theta' \in \Theta$.

Closed-Form Optimization Recipe for ERM

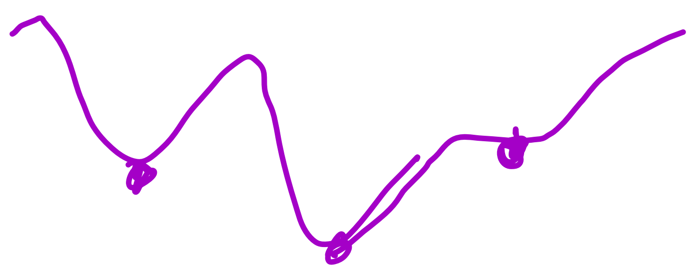
1. Derive the gradient $\nabla R(f_\theta, \mathcal{D})$.

Closed-Form Optimization Recipe for ERM

1. Derive the gradient $\nabla R(f_\theta, \mathcal{D})$.
2. Solve the gradient equation $\nabla R(f_\theta, \mathcal{D}) = 0$, obtaining all solutions.

Closed-Form Optimization Recipe for ERM

1. Derive the gradient $\nabla R(f_\theta, \mathcal{D})$.
2. Solve the gradient equation $\nabla R(f_\theta, \mathcal{D}) = 0$, obtaining all solutions.
3. Determine which solutions of the gradient equation are local minimizers by checking the hessian condition.



Closed-Form Optimization Recipe for ERM

1. Derive the gradient $\nabla R(f_\theta, \mathcal{D})$.
2. Solve the gradient equation $\nabla R(f_\theta, \mathcal{D}) = 0$, obtaining all solutions.
3. Determine which solutions of the gradient equation are local minimizers by checking the hessian condition.
4. Check the value $R(f_\theta, \mathcal{D})$ at each local minimizer to determine which are global minimizers.

Helpful Results

Some helpful results for optimizing the linear regression model.

- **Bias Absorption:** The prediction function $\mathbf{x}\mathbf{w} + b$ can be expressed as a single inner product by defining $\tilde{\mathbf{x}} = [\mathbf{x}, 1]$ and $\tilde{\theta} = [\mathbf{w}; b]$. We then have $\tilde{\mathbf{x}}\tilde{\theta} = \mathbf{x}\mathbf{w} + b$. For simplicity, we will assume bias absorption and write the prediction function as $\mathbf{x}\theta$.

Helpful Results

Some helpful results for optimizing the linear regression model.

- **Bias Absorption:** The prediction function $\mathbf{x}\mathbf{w} + b$ can be expressed as a single inner product by defining $\tilde{\mathbf{x}} = [\mathbf{x}, 1]$ and $\tilde{\theta} = [\mathbf{w}; b]$. We then have $\tilde{\mathbf{x}}\tilde{\theta} = \mathbf{x}\mathbf{w} + b$. For simplicity, we will assume bias absorption and write the prediction function as $\mathbf{x}\theta$.
- **Matrix form of the Risk:** The empirical risk function is easier to work with in matrix form. Define $\mathbf{X}$ to be the $N \times D$ matrix of inputs and $\mathbf{Y}$ to be the $N \times 1$ matrix of outputs. Then:

$$\frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \mathbf{x}_n\theta)^2 = \frac{1}{N}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta)$$

$Y - X\theta$

$Y - X\theta =$

$y_i - \theta x_i$

Helpful Results

Some helpful results for optimizing the linear regression model.

- **Bias Absorption:** The prediction function $\mathbf{x}\mathbf{w} + b$ can be expressed as a single inner product by defining $\tilde{\mathbf{x}} = [\mathbf{x}, 1]$ and $\tilde{\theta} = [\mathbf{w}; b]$. We then have $\tilde{\mathbf{x}}\tilde{\theta} = \mathbf{x}\mathbf{w} + b$. For simplicity, we will assume bias absorption and write the prediction function as $\mathbf{x}\theta$.
- **Matrix form of the Risk:** The empirical risk function is easier to work with in matrix form. Define $\mathbf{X}$ to be the $N \times D$ matrix of inputs and $\mathbf{Y}$ to be the $N \times 1$ matrix of outputs. Then:

$$\frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \mathbf{x}_n\theta)^2 = \frac{1}{N}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta)$$

- **Matrix Calculus**: We will need two basic matrix calculus results: $\nabla \mathbf{c}^T\theta = \mathbf{c}$ where $\mathbf{c} \in \mathbb{R}^D$ is a $D \times 1$ vector, and $\underline{\nabla\theta^T\mathbf{A}\theta} = \underline{2\mathbf{A}\theta}$ for $\mathbf{A}$ a $D \times D$ real symmetric matrix.

Step 1: Derive Gradient

$$\nabla R(f_\theta, \mathcal{D}) = \nabla \frac{1}{N}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta)$$

Step 1: Derive Gradient

$$\nabla R(f_\theta, \mathcal{D}) = \nabla \frac{1}{N}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta)$$

$$= \frac{1}{N}\nabla(\mathbf{Y}^T - \theta^T\mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\theta)$$

Step 1: Derive Gradient

$$\nabla R(f_\theta, \mathcal{D}) = \nabla \frac{1}{N}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta)$$

$$= \frac{1}{N}\nabla(\mathbf{Y}^T - \theta^T\mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\theta)$$

$$= \frac{1}{N}\nabla(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\theta - \theta^T\mathbf{X}^T\mathbf{Y} + \theta^T\mathbf{X}^T\mathbf{X}\theta)$$

Step 1: Derive Gradient

$$
\begin{aligned}
\nabla R(f_\theta, \mathcal{D}) &= \nabla \frac{1}{N}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta) \\
&= \frac{1}{N}\nabla(\mathbf{Y}^T - \theta^T\mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\theta) \\
&= \frac{1}{N}\nabla(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\theta - \theta^T\mathbf{X}^T\mathbf{Y} + \theta^T\mathbf{X}^T\mathbf{X}\theta) \\
&= \frac{1}{N}\nabla(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\theta + \theta^T\mathbf{X}^T\mathbf{X}\theta)
\end{aligned}
$$

Step 1: Derive Gradient

$$\nabla R(f_\theta, \mathcal{D}) = \nabla \frac{1}{N}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta)$$

$$= \frac{1}{N}\nabla(\mathbf{Y}^T - \theta^T\mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\theta)$$

$$= \frac{1}{N}\nabla(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\theta - \theta^T\mathbf{X}^T\mathbf{Y} + \theta^T\mathbf{X}^T\mathbf{X}\theta)$$

$$= \frac{1}{N}\nabla(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\theta + \theta^T\mathbf{X}^T\mathbf{X}\theta)$$

$$= \frac{1}{N}(-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\theta)$$

Step 2: Solve Gradient Equation

$$\nabla R(f_\theta, \mathcal{D}) = 0$$

Step 2: Solve Gradient Equation

$$\nabla R(f_\theta, \mathcal{D}) = 0$$

$$\Rightarrow \frac{1}{N}(-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\theta) = 0$$

Step 2: Solve Gradient Equation

$$
\begin{aligned}
&\nabla R(f_\theta, \mathcal{D}) = 0 \\
&\Rightarrow \frac{1}{N}(-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\theta) = 0 \\
&\Rightarrow \frac{2}{N}\mathbf{X}^T\mathbf{X}\theta = \frac{2}{N}\mathbf{X}^T\mathbf{Y}
\end{aligned}
$$

Step 2: Solve Gradient Equation

$$
\begin{aligned}
&\nabla R(f_\theta, \mathcal{D}) = 0 \\
&\Rightarrow \frac{1}{N}(-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\theta) = 0 \\
&\Rightarrow \frac{2}{N}\mathbf{X}^T\mathbf{X}\theta = \frac{2}{N}\mathbf{X}^T\mathbf{Y} \\
&\Rightarrow \mathbf{X}^T\mathbf{X}\theta = \mathbf{X}^T\mathbf{Y}
\end{aligned}
$$

Step 2: Solve Gradient Equation

$$
\begin{aligned}
&\nabla R(f_\theta, \mathcal{D}) = 0 \\
&\Rightarrow \frac{1}{N}(-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\theta) = 0 \\
&\Rightarrow \frac{2}{N}\mathbf{X}^T\mathbf{X}\theta = \frac{2}{N}\mathbf{X}^T\mathbf{Y} \\
&\Rightarrow \mathbf{X}^T\mathbf{X}\theta = \mathbf{X}^T\mathbf{Y} \\
&\Rightarrow \hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}
\end{aligned}
$$

Step 2: Solve Gradient Equation

$$
\begin{aligned}
&\nabla R(f_\theta, \mathcal{D}) = 0 \\
&\Rightarrow \frac{1}{N}(-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\theta) = 0 \\
&\Rightarrow \frac{2}{N}\mathbf{X}^T\mathbf{X}\theta = \frac{2}{N}\mathbf{X}^T\mathbf{Y} \\
&\Rightarrow \mathbf{X}^T\mathbf{X}\theta = \mathbf{X}^T\mathbf{Y} \\
&\Rightarrow \hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}
\end{aligned}
$$

Note: This solution is only well-defined if $\mathbf{X}^T\mathbf{X}$ is invertible! It will be invertible if it is strictly positive definite.

Step 3: Check Hessian

- It's easy to see that $\nabla^2 R(f_\theta, \mathcal{D}) = 2\mathbf{X}^T\mathbf{X}$.

Step 3: Check Hessian

- It's easy to see that $\nabla^2 R(f_\theta, \mathcal{D}) = 2\mathbf{X}^T\mathbf{X}$.
- Thus, the solution $\hat{\theta}$ is a local minimizer if it is well-defined.

Step 3: Check Hessian

- It's easy to see that $\nabla^2 R(f_\theta, \mathcal{D}) = 2\mathbf{X}^T\mathbf{X}$.
- Thus, the solution $\hat{\theta}$ is a local minimizer if it is well-defined.
- Since there is at most one local minimizer $\hat{\theta}$, $\hat{\theta}$ is the global minimizer so long as it is well defined.

Step 3: Check Hessian

- It's easy to see that $\nabla^2 R(f_\theta, \mathcal{D}) = 2\mathbf{X}^T\mathbf{X}$.
- Thus, the solution $\hat{\theta}$ is a local minimizer if it is well-defined.
- Since there is at most one local minimizer $\hat{\theta}$, $\hat{\theta}$ is the global minimizer so long as it is well defined.
- Therefore, $\hat{\theta}$ is the solution to the ERM learning problem for the linear regression model with squared loss when it is well defined.

Making Predictions

- To make a prediction for a new data point $\mathbf{x}_*$, we compute:

Making Predictions

- To make a prediction for a new data point $\mathbf{x}_*$, we compute:

Making Predictions

- To make a prediction for a new data point $\mathbf{x}_*$, we compute:

$$\hat{y} = f_{\hat{\theta}}(\mathbf{x}_*) = \mathbf{x}_*\hat{\theta}$$