Tutorial 5

Oct. 10-14, 2016

Overview

Review

- Notations
- Linear Algebra
- Calculus

Least Squares

- Regression
- Linear Regression
- Least Squares
- Non-linear basis

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.
- First/last lowercase letters for vector: $w = [0.1,\ 0.2]^{\mathrm{T}}$.

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.
- First/last lowercase letters for vector: $w = [0.1,\ 0.2]^{\mathrm{T}}$.
- First/last uppercase letters for matrices: X, Y, W, A, B.

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.
- First/last lowercase letters for vector: $w = [0.1,\ 0.2]^{\mathrm{T}}$.
- First/last uppercase letters for matrices: X, Y, W, A, B.
- For indices we use i, j, k.

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.
- First/last lowercase letters for vector: $w = [0.1,\ 0.2]^{\mathrm{T}}$.
- First/last uppercase letters for matrices: X, Y, W, A, B.
- For indices we use i, j, k.
- For sizes we use m, n, d, p, k.

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.
- First/last lowercase letters for vector: $w = [0.1,\ 0.2]^{\mathrm{T}}$.
- First/last uppercase letters for matrices: X, Y, W, A, B.
- For indices we use i, j, k.
- For sizes we use m, n, d, p, k.
- For sets we use S, T, U, V.

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.
- First/last lowercase letters for vector: $w = [0.1,\ 0.2]^{\mathrm{T}}$.
- First/last uppercase letters for matrices: X, Y, W, A, B.
- For indices we use i, j, k.
- For sizes we use m, n, d, p, k.
- For sets we use S, T, U, V.
- For functions we use $\mathrm{f}, \mathrm{g}, \mathrm{h}$.

Notations

- Greek letters for scalers: $\alpha = 1, \beta = 3.4, \gamma = \pi$.
- First/last lowercase letters for vector: $w = [0.1,\ 0.2]^{\mathrm{T}}$.
- First/last uppercase letters for matrices: X, Y, W, A, B.
- For indices we use i, j, k.
- For sizes we use m, n, d, p, k.
- For sets we use S, T, U, V.
- For functions we use $\mathrm{f}, \mathrm{g}, \mathrm{h}$.
- $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ are the train and test datasets.

Linear Algebra

Linear Algebra

- Vector dot product (in matrix-form operation):

$$a^{\mathrm{T}}b = \begin{bmatrix} a_1, a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 \cdot b_1 + a_2 \cdot b_2$$

Linear Algebra

- Vector dot product (in matrix-form operation):

$$a^{\mathrm{T}}b = \begin{bmatrix} a_1, a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 \cdot b_1 + a_2 \cdot b_2$$

- **Length** of a vector a is $\|a\| = \sqrt{a^{\mathrm{T}}a} = \sqrt{\sum_{i=1}^{d} a_i^2}$.

Linear Algebra

- Vector dot product (in matrix-form operation):

$$a^{\mathrm{T}}b = \begin{bmatrix} a_1, a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 \cdot b_1 + a_2 \cdot b_2$$

- **Length** of a vector a is $\|a\| = \sqrt{a^{\mathrm{T}}a} = \sqrt{\sum_{i=1}^{d} a_i^2}$.
- We define ℓ_p**-norm** as $\|x\|_p = \sqrt[p]{\sum_{i=1}^{d} |a_i|^p}$.

Linear Algebra

- Vector dot product (in matrix-form operation):

$$a^{\mathrm{T}}b = \begin{bmatrix} a_1, a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 \cdot b_1 + a_2 \cdot b_2$$

- **Length** of a vector a is $\|a\| = \sqrt{a^{\mathrm{T}}a} = \sqrt{\sum_{i=1}^{d} a_i^2}$.
- We define ℓ_p**-norm** as $\|x\|_p = \sqrt[p]{\sum_{i=1}^{d} |a_i|^p}$.
- The length of a vector (as learned in high-school) is an ℓ_2-norm of a vector.

Linear Algebra

- Vector dot product (in matrix-form operation):

$$a^{\mathrm{T}}b = \begin{bmatrix} a_1, a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 \cdot b_1 + a_2 \cdot b_2$$

- **Length** of a vector a is $\|a\| = \sqrt{a^{\mathrm{T}}a} = \sqrt{\sum_{i=1}^{d} a_i^2}$.
- We define ℓ_p**-norm** as $\|x\|_p = \sqrt[p]{\sum_{i=1}^{d} |a_i|^p}$.
- The length of a vector (as learned in high-school) is an ℓ_2-norm of a vector.
- The ℓ_1-norm is $\|x\|_1 = \sum_{i=1}^{d} |a_i|$.

Linear Algebra

- Vector dot product (in matrix-form operation):

$$a^{\mathrm{T}}b = \begin{bmatrix} a_1, a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 \cdot b_1 + a_2 \cdot b_2$$

- **Length** of a vector a is $\|a\| = \sqrt{a^{\mathrm{T}}a} = \sqrt{\sum_{i=1}^{d} a_i^2}$.
- We define ℓ_p**-norm** as $\|x\|_p = \sqrt[p]{\sum_{i=1}^{d} |a_i|^p}$.
- The length of a vector (as learned in high-school) is an ℓ_2-norm of a vector.
- The ℓ_1-norm is $\|x\|_1 = \sum_{i=1}^{d} |a_i|$.
- You'll commonly see $\|x\|_2^2 = x^{\mathrm{T}}x$.

Linear Algebra

Linear Algebra

- A matrix X is **symmetric** if $X^{\mathrm{T}} = X$.

Linear Algebra

- A matrix X is **symmetric** if $X^{\mathrm{T}} = X$.
- A symmetric matrix X is **positive semi-definite** if for all non-zero vectors z we have $z^{\mathrm{T}} X z \geq 0$.

Linear Algebra

- A matrix X is **symmetric** if $X^{\mathrm{T}} = X$.
- A symmetric matrix X is **positive semi-definite** if for all non-zero vectors z we have $z^{\mathrm{T}} X z \geq 0$.
- $\mathrm{f}(z) = z^{\mathrm{T}} X z$ is a quadratic function of z, furthermore, $\mathrm{f}(\cdot)$ is convex if X is positive semi-definite.

Calculus

- Assuming $\mathrm{f}(x) : \mathbb{R}^d \rightarrow \mathbb{R}$.

Calculus

- Assuming $\mathrm{f}(x) : \mathbb{R}^d \rightarrow \mathbb{R}$.
- The **gradient** vector $\nabla \mathrm{f}(x)$ is a vector of partial derivatives $[\frac{\partial}{\partial x_1}\mathrm{f}, \frac{\partial}{\partial x_2}\mathrm{f}, \ldots, \frac{\partial}{\partial x_d}\mathrm{f}]$.

Calculus

- Assuming $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}$.
- The **gradient** vector $\nabla f(x)$ is a vector of partial derivatives $[\frac{\partial}{\partial x_1} f, \frac{\partial}{\partial x_2} f, \ldots, \frac{\partial}{\partial x_d} f]$.
- The **Hessian** matrix $\nabla^2 f(x)$ is a matrix of second order partial derivatives.

$$\begin{bmatrix} \frac{\partial}{\partial x_1 \partial x_1} f & \cdots & \frac{\partial}{\partial x_1 \partial x_d} f \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_d \partial x_1} f & \cdots & \frac{\partial}{\partial x_d \partial x_d} f \end{bmatrix}$$

Calculus

- Assuming $\mathrm{f}(x) : \mathbb{R}^d \to \mathbb{R}$.
- The **gradient** vector $\nabla \mathrm{f}(x)$ is a vector of partial derivatives $[\frac{\partial}{\partial x_1}\mathrm{f}, \frac{\partial}{\partial x_2}\mathrm{f}, \ldots, \frac{\partial}{\partial x_d}\mathrm{f}]$.
- The **Hessian** matrix $\nabla^2 \mathrm{f}(x)$ is a matrix of second order partial derivatives.

$$\begin{bmatrix} \frac{\partial}{\partial x_1 \partial x_1}\mathrm{f} & \cdots & \frac{\partial}{\partial x_1 \partial x_d}\mathrm{f} \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_d \partial x_1}\mathrm{f} & \cdots & \frac{\partial}{\partial x_d \partial x_d}\mathrm{f} \end{bmatrix}$$

- In single-variable calculus: A function $\mathrm{f}(x)$ is convex around x if the second derivate $\mathrm{f}''(x) \geq 0$. In that case x is a local minimum of $\mathrm{f}(x)$.

Calculus

- The **Hessian** matrix $\nabla^2 \mathrm{f}(x)$ is a matrix of second order partial derivatives.

$$\begin{bmatrix} \frac{\partial}{\partial x_1 \partial x_1}\mathrm{f} & \cdots & \frac{\partial}{\partial x_1 \partial x_d}\mathrm{f} \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_d \partial x_1}\mathrm{f} & \cdots & \frac{\partial}{\partial x_d \partial x_d}\mathrm{f} \end{bmatrix}$$

- In single-variable calculus: A function $\mathrm{f}(x)$ is convex around x if the second derivate $\mathrm{f}''(x) \geq 0$. In that case x is a local minimum of $\mathrm{f}(x)$.
- In multivariate calculus: A function $\mathrm{f}(x)$ is convex around x if the Hessian $\nabla^2 \mathrm{f}(x)$ is positive semi-definite. In that case x is a local minimum of $\mathrm{f}(x)$.

Regression

Regression

- **Objective**. Learn a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $f(x)$.

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \to \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$, that is, m sample pairs of (x_i, y_i).

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \to \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$, that is, m sample pairs of (x_i, y_i).
- Given a test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^n$ with n pairs, how can we measure the error of f?

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \to \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$, that is, m sample pairs of (x_i, y_i).
- Given a test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^n$ with n pairs, how can we measure the error of f?
    1. $\sum_{i=1}^n (f(x_i) - y_i)$ is this a good measure?

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \to \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$, that is, m sample pairs of (x_i, y_i).
- Given a test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^n$ with n pairs, how can we measure the error of f?
  1. $\sum_{i=1}^n (f(x_i) - y_i)$ is this a good measure?
  2. $\sum_{i=1}^n |f(x_i) - y_i|$ is this a good measure?

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \rightarrow \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{m}$, that is, m sample pairs of (x_i, y_i).
- Given a test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^{n}$ with n pairs, how can we measure the error of f?
    1. $\sum_{i=1}^{n}(f(x_i) - y_i)$ is this a good measure?
    2. $\sum_{i=1}^{n}|f(x_i) - y_i|$ is this a good measure?
    3. $\frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i|$ is this a good measure?

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \to \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$, that is, m sample pairs of (x_i, y_i).
- Given a test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^n$ with n pairs, how can we measure the error of f?
  1. $\sum_{i=1}^n (f(x_i) - y_i)$ is this a good measure?
  2. $\sum_{i=1}^n |f(x_i) - y_i|$ is this a good measure?
  3. $\frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|$ is this a good measure?
  4. $\frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2$ is this a good measure?

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \to \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$, that is, m sample pairs of (x_i, y_i).
- Given a test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^n$ with n pairs, how can we measure the error of f?
  1. $\sum_{i=1}^n (f(x_i) - y_i)$ is this a good measure?
  2. $\sum_{i=1}^n |f(x_i) - y_i|$ is this a good measure?
  3. $\frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|$ is this a good measure?
  4. $\frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2$ is this a good measure?
  5. Is 3 better or 4? (why)

Regression

- **Objective**. Learn a function $\mathrm{f} : \mathbb{R}^d \to \mathbb{R}$. Given a vector $x \in \mathbb{R}^d$ we make a prediction $y \in \mathbb{R}$ by evaluating the $\mathrm{f}(x)$.
- **Data**. We have a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$, that is, m sample pairs of (x_i, y_i).
- Given a test set $\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^n$ with n pairs, how can we measure the error of f?
  1. $\sum_{i=1}^n (f(x_i) - y_i)$ is this a good measure?
  2. $\sum_{i=1}^n |f(x_i) - y_i|$ is this a good measure?
  3. $\frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|$ is this a good measure?
  4. $\frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2$ is this a good measure?
  5. Is 3 better or 4? (why)
  6. What is a good measure?

Linear Regression - Least Squares

Linear Regression - Least Squares

- In this setting the function $f(\cdot)$ has a specific form.

Linear Regression - Least Squares

- In this setting the function $f(\cdot)$ has a specific form.
- The function $f(x)$ with a parameter $w \in \mathbb{R}^d$ makes the prediction as $w^{\mathrm{T}}x$.

Linear Regression - Least Squares

- In this setting the function $\mathrm{f}(\cdot)$ has a specific form.
- The function $\mathrm{f}(x)$ with a parameter $w \in \mathbb{R}^d$ makes the prediction as $w^{\mathrm{T}}x$.
- We can write it as $\mathrm{f}(x\,; w) = w^{\mathrm{T}}x$.

Linear Regression - Least Squares

- In this setting the function f(·) has a specific form.
- The function f(x) with a parameter $w \in \mathbb{R}^d$ makes the prediction as $w^{\mathrm{T}}x$.
- We can write it as f$(x\,; w) = w^{\mathrm{T}}x$.
- **Objective**. Given $\mathcal{D}_{\text{train}}$ find a $\hat{w}$ that minimizes the mean squared error on $\mathcal{D}_{\text{train}}$.

Linear Regression - Least Squares

- In this setting the function $f(\cdot)$ has a specific form.
- The function $f(x)$ with a parameter $w \in \mathbb{R}^d$ makes the prediction as $w^T x$.
- We can write it as $f(x\,; w) = w^T x$.
- **Objective**. Given $\mathcal{D}_{\text{train}}$ find a $\hat{w}$ that minimizes the mean squared error on $\mathcal{D}_{\text{train}}$.
  - How can we find $\hat{w}$?

Linear Regression - Least Squares

- In this setting the function $f(\cdot)$ has a specific form.
- The function $f(x)$ with a parameter $w \in \mathbb{R}^d$ makes the prediction as $w^T x$.
- We can write it as $f(x\,; w) = w^T x$.
- **Objective**. Given $\mathcal{D}_{\text{train}}$ find a $\hat{w}$ that minimizes the mean squared error on $\mathcal{D}_{\text{train}}$.
  - How can we find $\hat{w}$?
  - Is $\hat{w}$ also going to minimize the mean squared error on $\mathcal{D}_{\text{test}}$?

Linear Regression - Least Squares

- In this setting the function f(·) has a specific form.
- The function $\mathrm{f}(x)$ with a parameter $w \in \mathbb{R}^d$ makes the prediction as $w^{\mathrm{T}}x$.
- We can write it as $\mathrm{f}(x\,; w) = w^{\mathrm{T}}x$.
- **Objective**. Given $\mathcal{D}_{\text{train}}$ find a $\hat{w}$ that minimizes the mean squared error on $\mathcal{D}_{\text{train}}$.
  - How can we find $\hat{w}$?
  - Is $\hat{w}$ also going to minimize the mean squared error on $\mathcal{D}_{\text{test}}$?
- We can solve this problem analytically :D.

Linear Regression - Least Squares

- In this setting the function $\mathrm{f}(\cdot)$ has a specific form.
- The function $\mathrm{f}(x)$ with a parameter $w \in \mathbb{R}^d$ makes the prediction as $w^{\mathrm{T}}x$.
- We can write it as $\mathrm{f}(x\,; w) = w^{\mathrm{T}}x$.
- **Objective**. Given $\mathcal{D}_{\text{train}}$ find a $\hat{w}$ that minimizes the mean squared error on $\mathcal{D}_{\text{train}}$.
  - How can we find $\hat{w}$?
  - Is $\hat{w}$ also going to minimize the mean squared error on $\mathcal{D}_{\text{test}}$?
- We can solve this problem analytically :D.
  - You really have to appreciate this – an analytical solution rarely pops out in typical machine learning problems.

Solving Least Squares

Solving Least Squares

- First let's rewrite the mean squared error as a function of w.

Solving Least Squares

- First let's rewrite the mean squared error as a function of w.

$$\mathcal{L}(w) = \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2$$
$$= \frac{1}{m}\sum_{i=1}^{m}(w^{\mathrm{T}}x_i - y_i)^2$$

Solving Least Squares

- First let's rewrite the mean squared error as a function of w.

$$\begin{aligned}\mathcal{L}(w) &= \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}(w^{\mathrm{T}}x_i - y_i)^2\end{aligned}$$

$$\mathcal{L}(w) = \frac{1}{m}(Xw - Y)^{\mathrm{T}}(Xw - Y) = \frac{1}{m}\|Xw - Y\|_2^2$$

Solving Least Squares

- First let's rewrite the mean squared error as a function of w.

$$\mathcal{L}(w) = \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2$$
$$= \frac{1}{m}\sum_{i=1}^{m}(w^{\mathrm{T}}x_i - y_i)^2$$

$$\mathcal{L}(w) = \frac{1}{m}(Xw - Y)^{\mathrm{T}}(Xw - Y) = \frac{1}{m}\|Xw - Y\|_2^2$$

- Notice that $\mathcal{L}(w)$ is a quadratic and a convex function of w (why convex?).

Solving Least Squares

- First let's rewrite the mean squared error as a function of w.

$$\begin{aligned}\mathcal{L}(w) &= \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}(w^{\mathrm{T}}x_i - y_i)^2\end{aligned}$$

$$\mathcal{L}(w) = \frac{1}{m}(Xw - Y)^{\mathrm{T}}(Xw - Y) = \frac{1}{m}\|Xw - Y\|_2^2$$

- Notice that $\mathcal{L}(w)$ is a quadratic and a convex function of w (why convex?).
- Thus, the w that sets $\nabla\mathcal{L}(w) = 0$ is a minimum of the function $\mathcal{L}(w)$.

Solving Least Squares

Solving Least Squares

- The $\nabla \mathcal{L}(w)$ is:

Solving Least Squares

- The $\nabla\mathcal{L}(w)$ is:

$$\nabla\mathcal{L}(w) = \frac{2}{m} X^{\mathrm{T}}(Xw - Y)$$

Solving Least Squares

- The $\nabla\mathcal{L}(w)$ is:

$$\nabla\mathcal{L}(w) = \frac{2}{m}X^{\mathrm{T}}(Xw - Y)$$

$$\nabla\mathcal{L}(w) = 0 \Rightarrow w = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y$$

Solving Least Squares

- The $\nabla\mathcal{L}(w)$ is:

$$\nabla\mathcal{L}(w) = \frac{2}{m}X^{\mathrm{T}}(Xw - Y)$$

$$\nabla\mathcal{L}(w) = 0 \Rightarrow w = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y$$

- Is $(X^{\mathrm{T}}X)$ necessarily invertible? If not, what should we do?

Solving Least Squares

- The $\nabla \mathcal{L}(w)$ is:

$$\nabla \mathcal{L}(w) = \frac{2}{m} X^{\mathrm{T}} (Xw - Y)$$

$$\nabla \mathcal{L}(w) = 0 \Rightarrow w = (X^{\mathrm{T}} X)^{-1} X^{\mathrm{T}} Y$$

- Is $(X^{\mathrm{T}} X)$ necessarily invertible? If not, what should we do?
- What's the time consuming part of this solution?

Notes on LS

Notes on LS

- What if we'd like to have an intercept (or bias) $\mathrm{f}(x) = w_1^{\mathrm{T}} x + w_0$?

Notes on LS

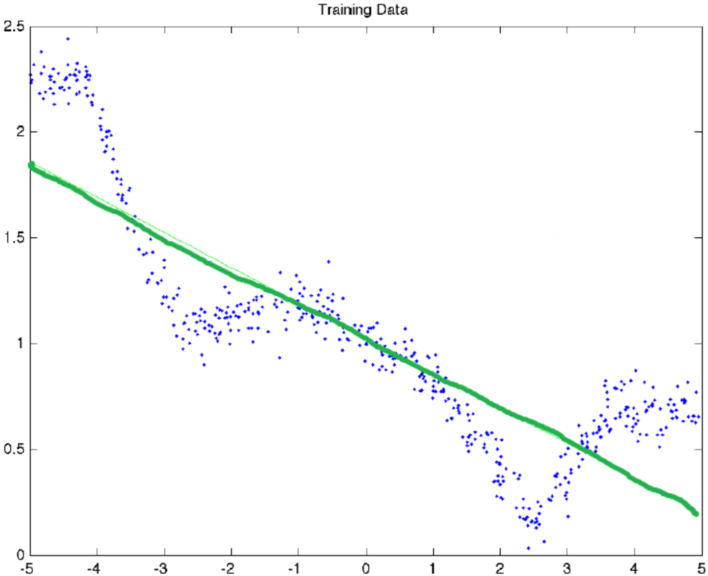
- What if we'd like to have an intercept (or bias) $\mathrm{f}(x) = w_1^{\mathrm{T}} x + w_0$?

$$X = \begin{bmatrix} 1 & x_1^{\mathrm{T}} \\ \vdots & \\ 1 & x_m^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_1^{(d)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_m^{(1)} & \cdots & x_m^{(d)} \end{bmatrix}$$

Solving Least Squares in Matlab

```
function [model] = simpleLeastSquares(X,y)

% Add bias variable
[N,D] = size(X);
X = [ones(N,1) X];

% Solve least squares problem
w = (X'*X)\X'*y;

model.w = w;
model.predict = @predict;

end

function [yhat] = predict(model,Xtest)
[T,D] = size(Xtest);
w = model.w;
Xtest = [ones(T,1) Xtest];
yhat = Xtest*w;
end
```

Solving Least Squares in Matlab



Notes on LS

- What if we'd like to have an intercept (or bias) $\mathrm{f}(x) = w_1^{\mathrm{T}} x + w_0$?

Notes on LS

- What if we'd like to have an intercept (or bias) $\mathrm{f}(x) = w_1^{\mathrm{T}} x + w_0$?
- What if it is not a (hyper)-plane or a line?

Notes on LS

- What if we'd like to have an intercept (or bias) $\mathrm{f}(x) = w_1^{\mathrm{T}} x + w_0$?
- What if it is not a (hyper)-plane or a line?

$$Xpoly = \begin{bmatrix} 1 & x_1 & (x_1)^2 & (x_1)^3 \\ 1 & x_2 & (x_2)^2 & (x_2)^3 \\ \vdots & & & \\ 1 & x_n & (x_n)^2 & (x_N)^3 \end{bmatrix}$$

Solving Least Squares in Matlab

```
function [model] = leastSquaresBasis(x,y,degree)

Xpoly = polyBasis(x,degree);

% Solve least squares problem
w = (Xpoly'*Xpoly)\Xpoly'*y;

model.w = w;
model.degree = degree;
model.predict = @predict;

end

function [yhat] = predict(model,Xtest)
Xpoly = polyBasis(Xtest,model.degree);
yhat = Xpoly*model.w;
end

function [Xpoly] = polyBasis(x,m)
n = length(x);
Xpoly = zeros(n,m+1);
for i = 0:m
    Xpoly(:,i+1) = x.^i;
end
end
```

Solving Least Squares in Matlab