FA17 10-701 Homework 2 Recitation 1

Logan Brooks   Matthew Oresky   Guoquan (GQ) Zhao

Perceptron learning algorithm

Q1: perceptrons: why $y \in \{-1, +1\}$ instead of $y \in \{0, 1\}$?

Perceptron learning algorithm

Q1: perceptrons: why $y \in \{-1, +1\}$ instead of $y \in \{0, 1\}$?

- Convenience: so $\hat{y} = \text{sign}(\mathbf{w}^T \mathbf{x})$.

Perceptron learning algorithm

Q1: perceptrons: why $y \in \{-1, +1\}$ instead of $y \in \{0, 1\}$?

- Convenience: so $\hat{y} = \mathrm{sign}(\mathbf{w}^T \mathbf{x})$.

Q2: what does $\mathbf{w}^T \mathbf{x} = 0$ mean?

Perceptron learning algorithm

Q1: perceptrons: why $y \in \{-1, +1\}$ instead of $y \in \{0, 1\}$?

- Convenience: so $\hat{y} = \mathrm{sign}(\mathbf{w}^T \mathbf{x})$.

Q2: what does $\mathbf{w}^T \mathbf{x} = 0$ mean?

- $\mathbf{x}$ is on the decision boundary.

Q3: what "should" we do if $\mathbf{w}^T \mathbf{x} = 0$?

Perceptron learning algorithm

Q1: perceptrons: why $y \in \{-1, +1\}$ instead of $y \in \{0, 1\}$?

- Convenience: so $\hat{y} = \text{sign}(\mathbf{w}^T\mathbf{x})$.

Q2: what does $\mathbf{w}^T\mathbf{x} = 0$ mean?

- $\mathbf{x}$ is on the decision boundary.

Q3: what "should" we do if $\mathbf{w}^T\mathbf{x} = 0$?

- Training: update $\mathbf{w}$ for both $y = -1$ and $y = +1$
  - $\hat{y} = 0$ is one way

Perceptron learning algorithm

Q1: perceptrons: why $y \in \{-1, +1\}$ instead of $y \in \{0, 1\}$?

- Convenience: so $\hat{y} = \text{sign}(\mathbf{w}^T\mathbf{x})$.

Q2: what does $\mathbf{w}^T\mathbf{x} = 0$ mean?

- $\mathbf{x}$ is on the decision boundary.

Q3: what "should" we do if $\mathbf{w}^T\mathbf{x} = 0$?

- Training: update $\mathbf{w}$ for both $y = -1$ and $y = +1$
  - $\hat{y} = 0$ is one way
- Application: make a valid prediction
  - arbitrarily assign $\hat{y} = -1$ or $\hat{y} = +1$

Perceptron learning algorithm

Q1: perceptrons: why $y \in \{-1, +1\}$ instead of $y \in \{0, 1\}$?

- Convenience: so $\hat{y} = \text{sign}(\mathbf{w}^T\mathbf{x})$.

Q2: what does $\mathbf{w}^T\mathbf{x} = 0$ mean?

- $\mathbf{x}$ is on the decision boundary.

Q3: what "should" we do if $\mathbf{w}^T\mathbf{x} = 0$?

- Training: update $\mathbf{w}$ for both $y = -1$ and $y = +1$
  - $\hat{y} = 0$ is one way
- Application: make a valid prediction
  - arbitrarily assign $\hat{y} = -1$ or $\hat{y} = +1$

Perceptron Update Rule

A simple learning algorithm - PLA

The perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\intercal}\mathbf{x})$$

Given the training set:

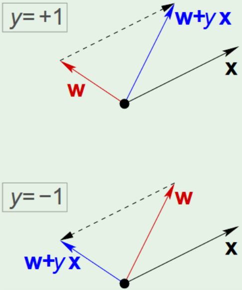
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

pick a misclassified point:

$$\text{sign}(\mathbf{w}^{\intercal}\mathbf{x}_n) \neq y_n$$

and update the weight vector:

$$\boxed{\mathbf{w} \leftarrow \mathbf{w} + y_n\mathbf{x}_n}$$



Perceptron learning algorithm & linear separability

For the perceptron algorithm on linearly separable training data:

- Does the algorithm terminate?
- Is $\hat{\mathbf{w}}$ guaranteed unique?
- Describe the outputted decision boundary.

Perceptron learning algorithm & linear separability

For the perceptron algorithm on linearly separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
- Describe the outputted decision boundary.

Perceptron learning algorithm & linear separability

For the perceptron algorithm on linearly separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
  - No.
- Describe the outputted decision boundary.

Perceptron learning algorithm & linear separability

For the perceptron algorithm on linearly separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
  - No.
- Describe the outputted decision boundary.
  - Linear separator (separating hyperplane) for the two classes in the training data

Perceptron learning algorithm & linear separability

For the perceptron algorithm on linearly **in**separable training data:

- Does the algorithm terminate?
- Is $\hat{\mathbf{w}}$ guaranteed unique?
- Describe the outputted decision boundary.

Perceptron learning algorithm & linear separability

For the perceptron algorithm on linearly **in**separable training data:

- Does the algorithm terminate?
  - No.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
- Describe the outputted decision boundary.

Perceptron learning algorithm & linear separability

For the perceptron algorithm on linearly **in**separable training data:

- Does the algorithm terminate?
  - No.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
  - N/A
- Describe the outputted decision boundary.
  - N/A

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly separable training data:

- Does the algorithm terminate?
- Is $\hat{\mathbf{w}}$ guaranteed unique?
- Describe the outputted regression function.

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?

- Describe the outputted regression function.

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
  - No.
- Describe the outputted regression function.

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
  - No.
- Describe the outputted regression function.
  - Looks close to a perceptron.
    - Training $y_i = 1 \implies \hat{y}_i$ extremely close to 1
    - Training $y_i = 0 \implies \hat{y}_i$ extremely close to 0

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly **in**separable training data:

- Does the algorithm terminate?
- Is $\hat{\mathbf{w}}$ guaranteed unique?
- Describe the outputted regression function.

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly **in**separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?

- Describe the outputted regression function.

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly **in**separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
  - No.
- Describe the outputted regression function.

Logistic regression & linear separability

For (unregularized) logistic regression using gradient ascent on linearly **in**separable training data:

- Does the algorithm terminate?
  - Yes.
- Is $\hat{\mathbf{w}}$ guaranteed unique?
  - No.
- Describe the outputted regression function.
  - It depends. Output $\hat{y}_i$'s potentially far from 0 and 1.

Linear model: test error

Setup:

- Someone gives us a $\hat{\mathbf{w}}$ for a linear model.
- Let trueError $= \mathbb{E}[(y - \mathbf{w}^T \mathbf{x})^2]$ for a new $(\mathbf{x}, y)$.
- Estimate trueError with mean squared error testError_m on m i.i.d. test points

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

m	$\mathbb{E}[\text{testError}_m]$	Bias_m	Var_m
tiny			
large			
limit as $m \to \infty$			

Linear model: test error

Setup:

- Someone gives us a $\hat{\mathbf{w}}$ for a linear model.
- Let trueError $= \mathbb{E}[(y - \mathbf{w}^T\mathbf{x})^2]$ for a new $(\mathbf{x}, y)$.
- Estimate trueError with mean squared error testError$_m$ on m i.i.d. test points

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

m	$\mathbb{E}[\text{testError}_m]$	Bias$_m$	Var$_m$
tiny	$=$ trueError		
large	$=$ trueError		
limit as $m \to \infty$	$=$ trueError		

Linear model: test error

Setup:

- Someone gives us a $\hat{\mathbf{w}}$ for a linear model.
- Let trueError $= \mathbb{E}[(y - \mathbf{w}^T\mathbf{x})^2]$ for a new $(\mathbf{x}, y)$.
- Estimate trueError with mean squared error testError$_m$ on m i.i.d. test points

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

m	$\mathbb{E}[\text{testError}_m]$	Bias$_m$	Var$_m$
tiny	$=$ trueError	$= 0$	
large	$=$ trueError	$= 0$	
limit as $m \to \infty$	$=$ trueError	$= 0$	

Linear model: test error

Setup:

- Someone gives us a $\hat{\mathbf{w}}$ for a linear model.
- Let trueError $= \mathbb{E}[(y - \mathbf{w}^T\mathbf{x})^2]$ for a new $(\mathbf{x}, y)$.
- Estimate trueError with mean squared error testError$_m$ on m i.i.d. test points

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

m	$\mathbb{E}[\text{testError}_m]$	Bias$_m$	Var$_m$
tiny	$=$ trueError	$= 0$	$\gg 0$
large	$=$ trueError	$= 0$	> 0
limit as $m \to \infty$	$=$ trueError	$= 0$	$= 0$

Usual linear regression formula: training & test error

Let $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Let trueError denote its expected squared test error.

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

n	$\frac{1}{n}\|\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}\|_2^2$	trueError
$< p$		
$= p$		
$> p$		
limit as $n \to \infty$		

Usual linear regression formula: training & test error

Let $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Let trueError denote its expected squared test error.

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

n	$\frac{1}{n}\lVert\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}\rVert_2^2$	trueError
$< p$	ERR	ERR
$= p$		
$> p$		
limit as $n \to \infty$		

Usual linear regression formula: training & test error

Let $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Let trueError denote its expected squared test error.

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

n	$\frac{1}{n}\|\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}\|_2^2$	trueError
$< p$	ERR	ERR
$= p$	$= 0$	$\gg 0$
$> p$		
limit as $n \to \infty$		

Usual linear regression formula: training & test error

Let $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Let trueError denote its expected squared test error.

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

n	$\frac{1}{n}\|\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}\|_2^2$	trueError
$< p$	ERR	ERR
$= p$	$= 0$	$\gg 0$
$> p$	> 0	> 0
limit as $n \to \infty$		

Usual linear regression formula: training & test error

Let $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Let trueError denote its expected squared test error.

Fill in the table with $\{\ll, <, =, >, \gg\} \times \{0, \text{trueError}\}$:

n	$\frac{1}{n}\|\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}\|_2^2$	trueError
$< p$	ERR	ERR
$= p$	$= 0$	$\gg 0$
$> p$	> 0	> 0
limit as $n \to \infty$	> 0	> 0