Regression: Linear, Logistic, and Otherwise

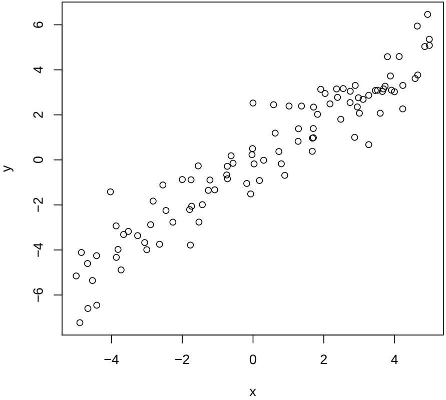
INST 808: Jordan Boyd-Graber

University of Maryland

Fall 2020

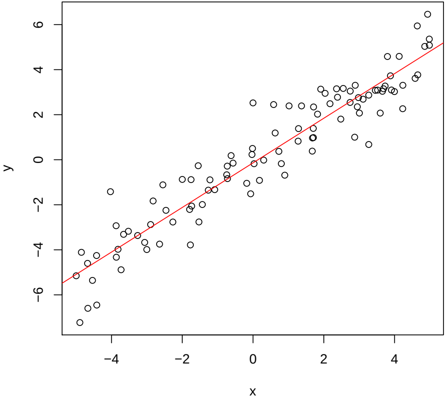
Slides adapted from Lauren Hannah

Linear Regression



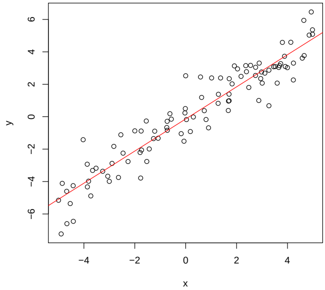
Data are the set of inputs and outputs, $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{n}$

Linear Regression



In *linear regression*, the goal is to predict y from x using a linear function

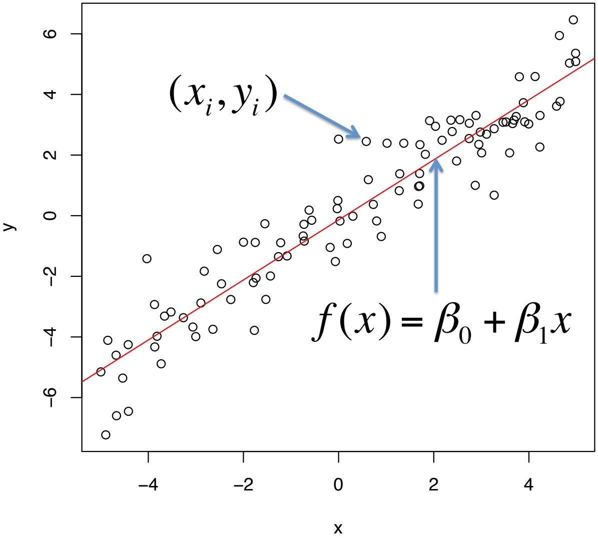
Linear Regression



Examples of linear regression:

- given a child's age and gender, what is his/her height?
- given unemployment, inflation, number of wars, and economic growth, what will the president's approval rating be?
- given a browsing history, how long will a user stay on a page?

Linear Regression



Multiple Covariates

Often, we have a vector of inputs where each represents a different *feature* of the data

$$\mathbf{x} = (x_1, \ldots, x_p)$$

The function fitted to the response is a linear combination of the covariates

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

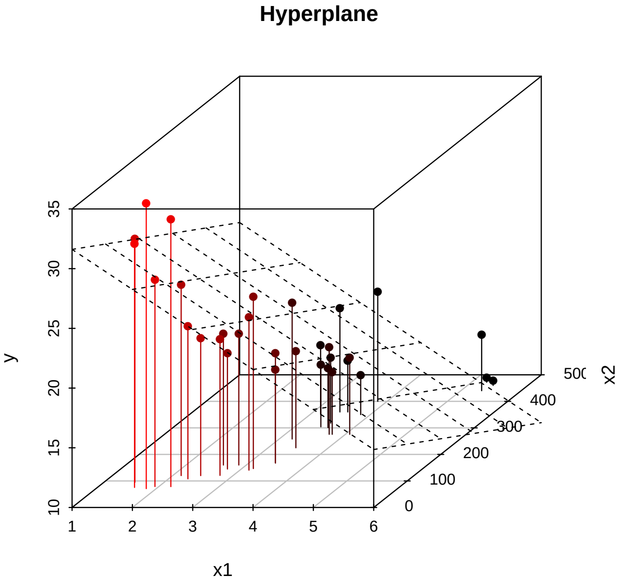
Multiple Covariates

- Often, it is convenient to represent $\mathbf{x}$ as $(1, x_1, \ldots, x_p)$
- In this case $\mathbf{x}$ is a vector, and so is β (we'll represent them in bold face)
- This is the dot product between these two vectors
- This then becomes (this should be familiar!)

$$f(\mathbf{x}) = \sum_{j=1}^{p} \beta_j x_j \tag{1}$$

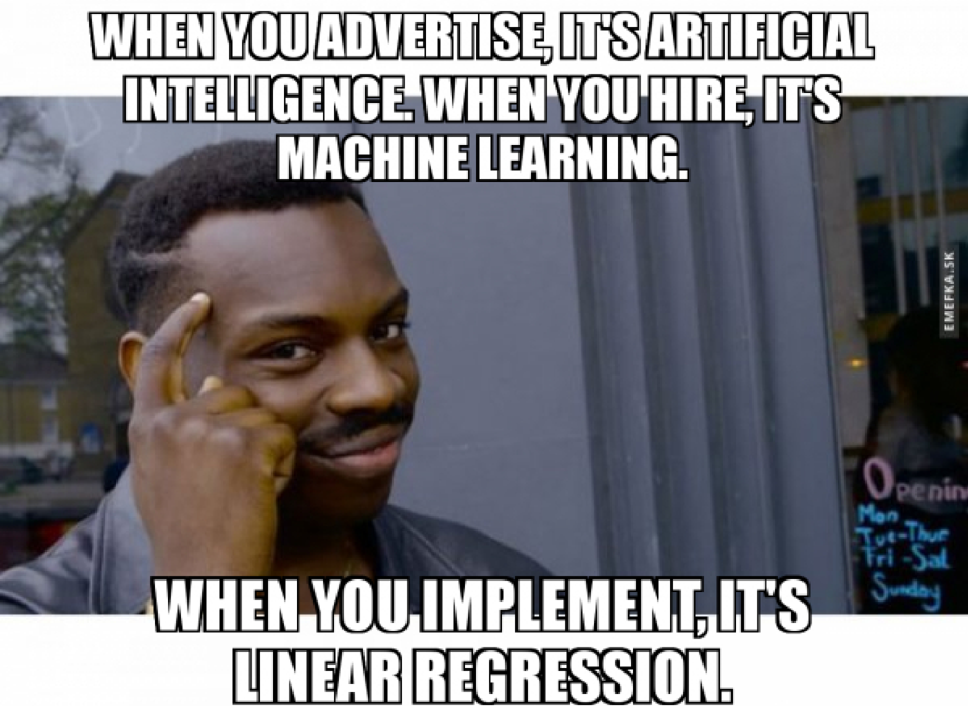
(2)

Hyperplanes: Linear Functions in Multiple Dimensions



Covariates

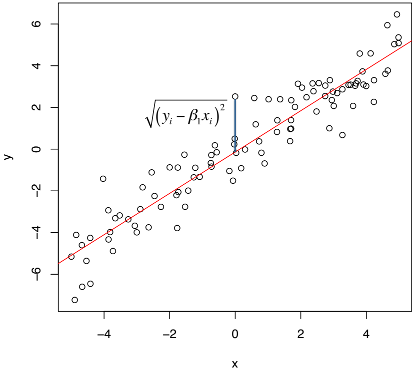
- Do not need to be raw value of $x_1, x_2, \ldots$
- Can be any feature or function of the data:
  - Transformations like $x_2 = \log(x_1)$ or $x_2 = \cos(x_1)$
  - Basis expansions like $x_2 = x_1^2$, $x_3 = x_1^3$, $x_4 = x_1^4$, etc
  - Indicators of events like $x_2 = 1_{\{-1 \leq x_1 \leq 1\}}$
  - Interactions between variables like $x_3 = x_1 x_2$
- Because of its simplicity and flexibility, it is one of the most widely implemented regression techniques


WHEN YOU ADVERTISE, IT'S ARTIFICIAL INTELLIGENCE. WHEN YOU HIRE, IT'S MACHINE LEARNING.
WHEN YOU IMPLEMENT, IT'S LINEAR REGRESSION.
EMEFKA.SK

Training, Validation, and Testing

- **Training Data**: Data with x and y, build your model on this
- **Validation Data**: Data with x and y, see how well your model did (you'll do this many times)
- **Test Data**:As far as you're concerned, data with only x

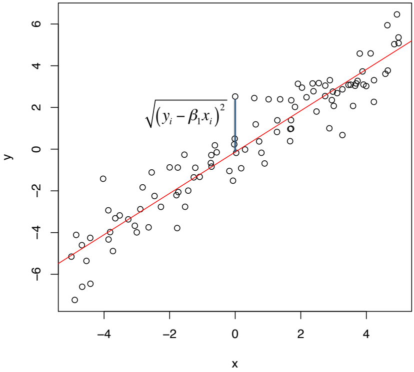
Fitting a Linear Regression



Idea: minimize the Euclidean distance between data and fitted line

$$RSS(\beta) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \vec{\beta}\cdot\mathbf{x}_i\right)^2$$

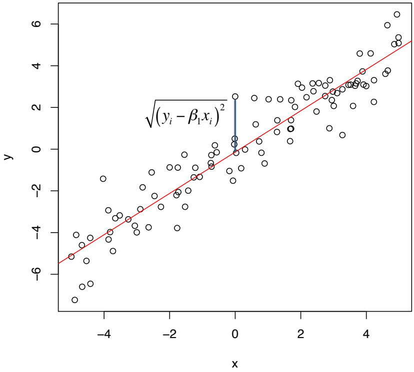
Fitting a Linear Regression



Idea: minimize the Euclidean distance between data and fitted line

$$RSS(\beta) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \vec{\beta}\cdot\mathbf{x}_i\right)^2$$

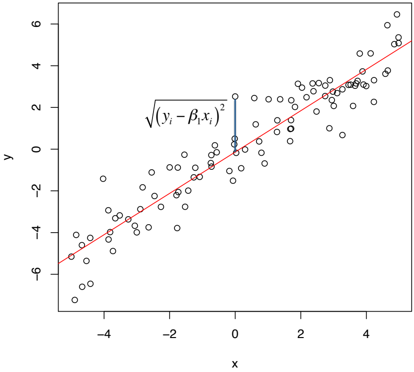
Fitting a Linear Regression



Idea: minimize the Euclidean distance between data and fitted line

$$RSS(\beta) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \vec{\beta}\cdot\mathbf{x}_i\right)^2$$

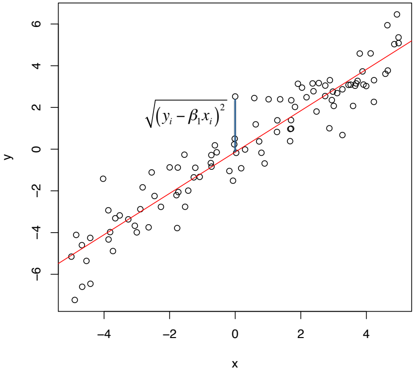
Fitting a Linear Regression



Idea: minimize the Euclidean distance between data and fitted line

$$RSS(\beta) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \vec{\beta}\cdot\mathbf{x}_i\right)^2$$

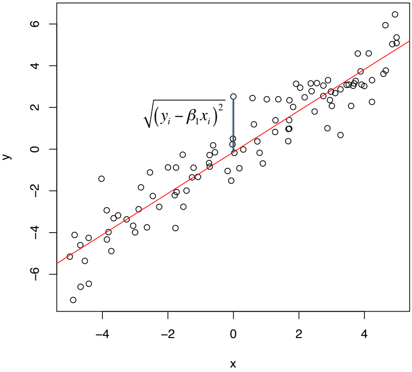
Fitting a Linear Regression



Idea: minimize the Euclidean distance between data and fitted line

$$RSS(\beta) = \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \vec{\beta} \cdot \mathbf{x}_i \right)^2$$

Fitting a Linear Regression



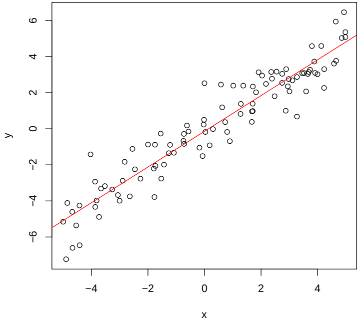
Idea: minimize the Euclidean distance between data and fitted line

$$RSS(\beta) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \vec{\beta}\cdot\mathbf{x}_i\right)^2$$

How to Find β

- Use calculus to find the value of β that minimizes the RSS
- The optimal value is

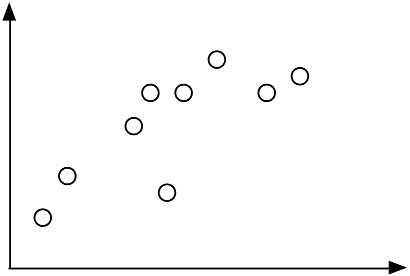
$$\hat{\beta} = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} x_i^2}$$



Prediction

- After finding $\hat{\beta}$, we would like to predict an output value for a new set of covariates
- We just find the point on the line that corresponds to the new input:

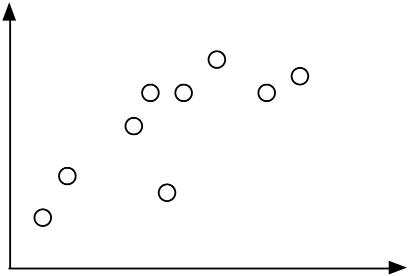
$$\hat{y} = \beta_0 + \beta_1 x \tag{3}$$



Prediction

- After finding $\hat{\beta}$, we would like to predict an output value for a new set of covariates
- We just find the point on the line that corresponds to the new input:

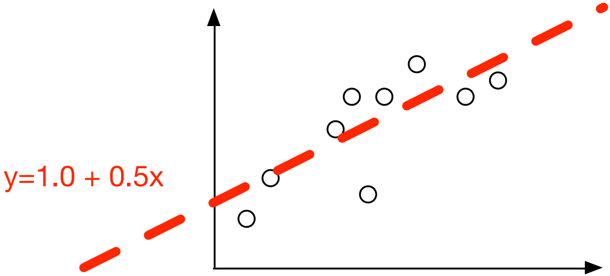
$$\hat{y} = \beta_0 + \beta_1 x \tag{3}$$



Prediction

- After finding $\hat{\beta}$, we would like to predict an output value for a new set of covariates
- We just find the point on the line that corresponds to the new input:

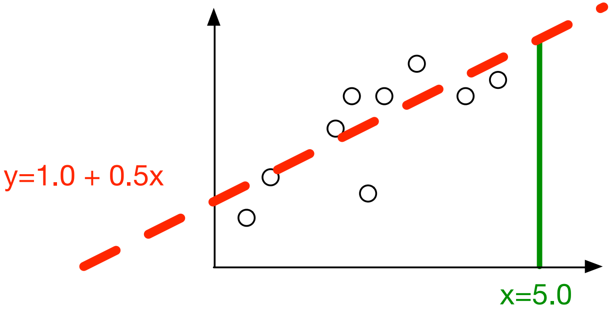
$$\hat{y} = 1.0 + 0.5x \tag{3}$$



Prediction

- After finding $\hat{\beta}$, we would like to predict an output value for a new set of covariates
- We just find the point on the line that corresponds to the new input:

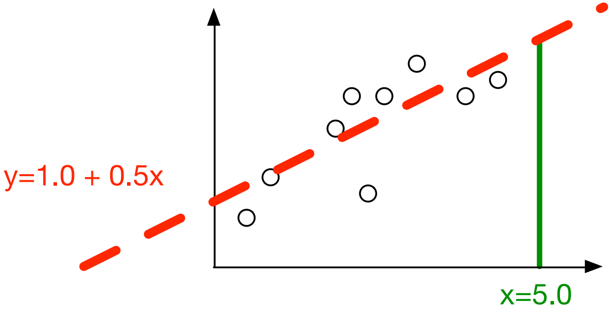
$$\hat{y} = 1.0 + 0.5 * 5 \quad (3)$$



Prediction

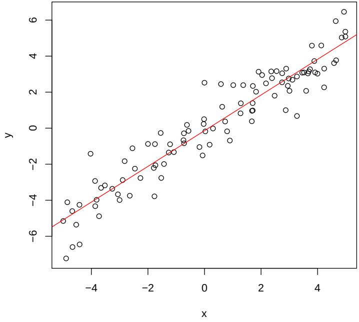
- After finding $\hat{\beta}$, we would like to predict an output value for a new set of covariates
- We just find the point on the line that corresponds to the new input:

$$\hat{y} = 3.5 \tag{3}$$



Probabilistic Interpretation

- Our analysis so far has not included any probabilities
- Linear regression does have a *probabilisitc* (probability model-based) interpretation

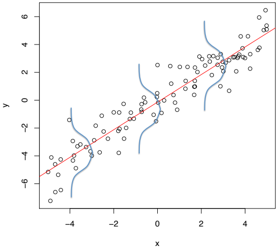


Probabilistic Interpretation

- Linear regression assumes that response values have a Gaussian distribution around the linear mean function,

$$Y_i \,|\, \mathbf{x}_i, \beta \sim N(\mathbf{x}_i \beta, \sigma^2)$$

- This is a *discriminative model*, where inputs x are not modeled



- Minimizing RSS is equivalent to maximizing conditional likelihood

Courses, Lectures, Exercises and More

http://boydgraber.org

Regression: Linear, Logistic, and Otherwise

INST 808: Jordan Boyd-Graber

University of Maryland

Fall 2020

Slides adapted from Hinrich Schütze and Lauren Hannah

What are we talking about?

- Statistical classification: $p(y|x)$
- Classification uses: ad placement, spam detection
- Building block of other machine learning methods

Logistic Regression: Definition

- Weight vector β_i
- Observations X_i
- "Bias" β_0 (like intercept in linear regression)

$$P(Y=0|X) = \frac{1}{1+\exp\left[\beta_0 + \sum_i \beta_i X_i\right]} \tag{4}$$

$$P(Y=1|X) = \frac{\exp\left[\beta_0 + \sum_i \beta_i X_i\right]}{1+\exp\left[\beta_0 + \sum_i \beta_i X_i\right]} \tag{5}$$

- For shorthand, we'll say that

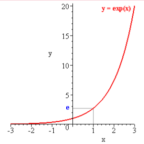
$$P(Y=0|X) = \sigma(-(\beta_0 + \sum_i \beta_i X_i)) \tag{6}$$

$$P(Y=1|X) = 1 - \sigma(-(\beta_0 + \sum_i \beta_i X_i)) \tag{7}$$

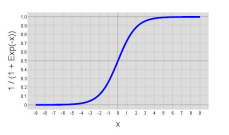
- Where $\sigma(z) = \frac{1}{1+exp[-z]}$

What's this "exp" doing?

Exponential



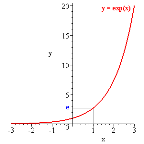
Logistic



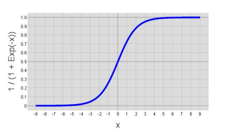
- $\exp[x]$ is shorthand for e^x
- e is a special number, about 2.71828
  - e^x is the limit of compound interest formula as compounds become infinitely small
  - It's the function whose derivative is itself
- The "logistic" function is $\sigma(z) = \frac{1}{1+e^{-z}}$
- Looks like an "S"
- Always between 0 and 1.

What's this "exp" doing?

Exponential



Logistic



- $\exp[x]$ is shorthand for e^x
- e is a special number, about 2.71828
  - e^x is the limit of compound interest formula as compounds become infinitely small
  - It's the function whose derivative is itself
- The "logistic" function is $\sigma(z) = \frac{1}{1+e^{-z}}$
- Looks like an "S"
- Always between 0 and 1.
  - Allows us to model probabilities
  - Different from **linear** regression

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	−1.0
"work"	β_3	−0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 1: Empty Document?

$X = \{\}$

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	-1.0
"work"	β_3	-0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 1: Empty Document?

$X = \{\}$

- $P(Y = 0) = \frac{1}{1+\exp[0.1]} =$
- $P(Y = 1) = \frac{\exp[0.1]}{1+\exp[0.1]} =$

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	−1.0
"work"	β_3	−0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 1: Empty Document?

$X = \{\}$

- $P(Y = 0) = \frac{1}{1+\exp[0.1]} = 0.48$
- $P(Y = 1) = \frac{\exp[0.1]}{1+\exp[0.1]} = 0.52$
- Bias β_0 encodes the prior probability of a class

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	-1.0
"work"	β_3	-0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 2

$X = \{\text{Mother}, \text{Nigeria}\}$

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	−1.0
"work"	β_3	−0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 2

$X = \{\text{Mother}, \text{Nigeria}\}$

- $P(Y = 0) = \frac{1}{1+\exp[0.1-1.0+3.0]} =$
- $P(Y = 1) = \frac{\exp[0.1-1.0+3.0]}{1+\exp[0.1-1.0+3.0]} =$
- Include bias, and sum the other weights

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	−1.0
"work"	β_3	−0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 2

$X = \{\text{Mother}, \text{Nigeria}\}$

- $P(Y = 0) = \frac{1}{1 + \exp[0.1 - 1.0 + 3.0]} = 0.11$
- $P(Y = 1) = \frac{\exp[0.1 - 1.0 + 3.0]}{1 + \exp[0.1 - 1.0 + 3.0]} = 0.88$
- Include bias, and sum the other weights

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
“viagra”	β_1	2.0
“mother”	β_2	−1.0
“work”	β_3	−0.5
“nigeria”	β_4	3.0

- What does $Y = 1$ mean?

Example 3

$X = \{\text{Mother}, \text{Work}, \text{Viagra}, \text{Mother}\}$

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
“viagra”	β_1	2.0
“mother”	β_2	−1.0
“work”	β_3	−0.5
“nigeria”	β_4	3.0

- What does $Y = 1$ mean?

Example 3

$X = \{\text{Mother}, \text{Work}, \text{Viagra}, \text{Mother}\}$

- $P(Y = 0) = \frac{1}{1+\exp[0.1-1.0-0.5+2.0-1.0]} =$
- $P(Y = 1) = \frac{\exp[0.1-1.0-0.5+2.0-1.0]}{1+\exp[0.1-1.0-0.5+2.0-1.0]} =$
- Multiply feature presence by weight

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	-1.0
"work"	β_3	-0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 3

$X = \{\text{Mother}, \text{Work}, \text{Viagra}, \text{Mother}\}$

- $P(Y = 0) = \frac{1}{1+\exp[0.1-1.0-0.5+2.0-1.0]} = 0.60$
- $P(Y = 1) = \frac{\exp[0.1-1.0-0.5+2.0-1.0]}{1+\exp[0.1-1.0-0.5+2.0-1.0]} = 0.30$
- Multiply feature presence by weight

How is Logistic Regression Used?

- Given a set of weights $\vec{\beta}$, we know how to compute the conditional likelihood $P(y|\beta, x)$
- Find the set of weights $\vec{\beta}$ that maximize the conditional likelihood on training data (next week)
- **Intuition**: higher weights mean that this feature implies that this feature is a good this is the class you want for this observation

How is Logistic Regression Used?

- Given a set of weights $\vec{\beta}$, we know how to compute the conditional likelihood $P(y|\beta, x)$
- Find the set of weights $\vec{\beta}$ that maximize the conditional likelihood on training data (next week)
- **Intuition**: higher weights mean that this feature implies that this feature is a good this is the class you want for this observation
- Naïve Bayes is a special case of logistic regression that uses Bayes rule and conditional probabilities to set these weights

$$\arg\max_{c_j \in \mathbb{C}} \left[\ln \hat{P}(c_j) + \sum_{1 \le i \le n_d} \ln \hat{P}(w_i|c_j)\right]$$

How is Logistic Regression Used?

- Given a set of weights $\vec{\beta}$, we know how to compute the conditional likelihood $P(y|\beta, x)$
- Find the set of weights $\vec{\beta}$ that maximize the conditional likelihood on training data (next week)
- **Intuition**: higher weights mean that this feature implies that this feature is a good this is the class you want for this observation
- Naïve Bayes is a special case of logistic regression that uses Bayes rule and conditional probabilities to set these weights

$$\arg\max_{c_j \in \mathbb{C}} \left[\ln \hat{P}(c_j) + \sum_{1 \leq i \leq n_d} \ln \hat{P}(w_i|c_j)\right]$$

How is Logistic Regression Used?

- Given a set of weights $\vec{\beta}$, we know how to compute the conditional likelihood $P(y|\beta, x)$
- Find the set of weights $\vec{\beta}$ that maximize the conditional likelihood on training data (next week)
- **Intuition**: higher weights mean that this feature implies that this feature is a good this is the class you want for this observation
- Naïve Bayes is a special case of logistic regression that uses Bayes rule and conditional probabilities to set these weights

$$\arg\max_{c_j \in \mathbb{C}} \left[\ln \hat{P}(c_j) + \sum_{1 \le i \le n_d} \ln \hat{P}(w_i|c_j)\right]$$

Contrasting Naïve Bayes and Logistic Regression

- Naïve Bayes easier
- Naïve Bayes better on smaller datasets
- Logistic regression better on medium-sized datasets
- On huge datasets, it doesn't really matter (data always win)
  - Optional reading by Ng and Jordan has proofs and experiments
- Logistic regression allows arbitrary features (biggest difference!)

Contrasting Naïve Bayes and Logistic Regression

- Naïve Bayes easier
- Naïve Bayes better on smaller datasets
- Logistic regression better on medium-sized datasets
- On huge datasets, it doesn't really matter (data always win)
  - Optional reading by Ng and Jordan has proofs and experiments
- Logistic regression allows arbitrary features (biggest difference!)
- Don't need to memorize (or work through) previous slide—just understand that naïve Bayes is a special case of logistic regression

Next time . . .

- How to learn the best setting of weights
- Regularizing logistic regression to encourage sparse vectors
- Extracting features

Courses, Lectures, Exercises and More

http://boydgraber.org

Regression: Linear, Logistic, and Otherwise

INST 808: Jordan Boyd-Graber

University of Maryland

Fall 2020

Slides adapted from Emily Fox

Reminder: Logistic Regression

$$P(Y=0|X) = \frac{1}{1+\exp\left[\beta_0 + \sum_i \beta_i X_i\right]} \tag{8}$$

$$P(Y=1|X) = \frac{\exp\left[\beta_0 + \sum_i \beta_i X_i\right]}{1+\exp\left[\beta_0 + \sum_i \beta_i X_i\right]} \tag{9}$$

- Discriminative prediction: $p(y|x)$
- Classification uses: ad placement, spam detection
- What we didn't talk about is how to learn β from data

Objective for Logistic Regression

To ease notation, let's define

$$\pi_i = \frac{\exp \beta^T x_i}{1 + \exp \beta^T x_i} \tag{10}$$

Our objective function is

$$\mathscr{L} = \sum_i \log p(y_i \mid x_i) = \sum_i \mathscr{L}_i = \sum_i \begin{cases} \log(1 - \pi_i) & \text{if } y_i = 0 \\ \log \pi_i & \text{if } y_i = 1 \end{cases} \tag{11}$$

Objective for Logistic Regression

To ease notation, let's define

$$\pi_i = \frac{\exp \beta^T x_i}{1 + \exp \beta^T x_i} \tag{10}$$

Our objective function is

$$\mathscr{L} = \sum_i \log p(y_i \,|\, x_i) = \sum_i \mathscr{L}_i = \sum_i \begin{cases} \log(1 - \pi_i) & \text{if } y_i = 0 \\ \log \pi_i & \text{if } y_i = 1 \end{cases} \tag{11}$$

Objective for Logistic Regression

To ease notation, let's define

$$\pi_i = \frac{\exp \beta^T x_i}{1 + \exp \beta^T x_i} \tag{10}$$

Our objective function is

$$\mathscr{L} = \sum_i \log p(y_i \,|\, x_i) = \sum_i \mathscr{L}_i = \sum_i \begin{cases} \log(1 - \pi_i) & \text{if } y_i = 0 \\ \log \pi_i & \text{if } y_i = 1 \end{cases} \tag{11}$$

Chain Rule to the Rescue

Chain Rule

If

$$f(x) = u(v(x)), \tag{12}$$

then

$$\frac{d}{dx} f = \frac{du}{dv}\frac{dv}{dx} \tag{13}$$

- We know derivatives of individual functions, but not when they're put together
- Chain rule lets us compute overall derivatives anyway
- Derivative for logistic function

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i(1 - \pi_i)x_j, \tag{14}$$

Chain Rule to the Rescue

Chain Rule

If

$$f(x) = u(v(x)), \tag{12}$$

then

$$\frac{d}{dx} f = \frac{du}{dv} \frac{dv}{dx} \tag{13}$$

- We know derivatives of individual functions, but not when they're put together
- Chain rule lets us compute overall derivatives anyway
- Derivative for logistic function

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i (1 - \pi_i) x_j, \tag{14}$$

Chain Rule to the Rescue

Chain Rule

If

$$f(x) = u(v(x)), \tag{12}$$

then

$$\frac{d}{dx} f = \frac{du}{dv} \frac{dv}{dx} \tag{13}$$

- We know derivatives of individual functions, but not when they're put together
- Chain rule lets us compute overall derivatives anyway
- Derivative for logistic function

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i (1 - \pi_i) x_j, \tag{14}$$

Chain Rule to the Rescue

Chain Rule

If

$$f(x) = u(v(x)), \tag{12}$$

then

$$\frac{d}{dx} f = \frac{du}{dv} \frac{dv}{dx} \tag{13}$$

- We know derivatives of individual functions, but not when they're put together
- Chain rule lets us compute overall derivatives anyway
- Derivative for logistic function

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i (1 - \pi_i) x_j, \tag{14}$$

Chain Rule for Logistic Regression

Objective function:

$$\mathscr{L} = \sum_i \log p(y_i \,|\, x_i) = \sum_i \mathscr{L}_i = \sum_i \begin{cases} \log(1-\pi_i) & \text{if } y_i = 0 \\ \log \pi_i & \text{if } y_i = 1 \end{cases} \tag{15}$$

- In this case the objective function

$$f(x) = u(v(x)) = \log(\pi_i) \tag{16}$$

- Logarithm has nice derivative

$$\frac{d \log(v)}{dv} = \frac{1}{v} \tag{17}$$

- So does logistic function

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i(1-\pi_i)x_j. \tag{18}$$

Chain Rule for Logistic Regression

Objective function:

$$\mathscr{L} = \sum_i \log p(y_i \,|\, x_i) = \sum_i \mathscr{L}_i = \sum_i \begin{cases} \log(1-\pi_i) & \text{if } y_i = 0 \\ \log \pi_i & \text{if } y_i = 1 \end{cases} \tag{15}$$

- In this case the objective function

$$f(x) = u(v(x)) = \log(\pi_i) \tag{16}$$

- Logarithm has nice derivative

$$\frac{d \log(v)}{dv} = \frac{1}{v} \tag{17}$$

- So does logistic function

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i(1-\pi_i)x_j. \tag{18}$$

Chain Rule for Logistic Regression

Objective function:

$$\mathscr{L} = \sum_i \log p(y_i \,|\, x_i) = \sum_i \mathscr{L}_i = \sum_i \begin{cases} \log(1-\pi_i) & \text{if } y_i = 0 \\ \log \pi_i & \text{if } y_i = 1 \end{cases} \tag{15}$$

- In this case the objective function

$$f(x) = u(v(x)) = \log(\pi_i) \tag{16}$$

- Logarithm has nice derivative

$$\frac{d \log(v)}{dv} = \frac{1}{v} \tag{17}$$

- So does logistic function

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i(1-\pi_i)x_j. \tag{18}$$

Chain Rule for Logistic Regression

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathscr{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} \frac{1}{1-\pi_i}\left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \\ \frac{1}{\pi_i}\frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \end{cases} \tag{19}$$

$y == 0$

(20)

$y == 1$

(21)

Chain Rule for Logistic Regression

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathscr{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} \frac{1}{1-\pi_i}\left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \\ \frac{1}{\pi_i}\frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \end{cases} \tag{19}$$

$y == 0$

$$\frac{-\pi_i(1-\pi_i)}{1-\pi_i} x_j \tag{20}$$

$$\tag{21}$$

$y == 1$

$$\tag{22}$$

Chain Rule for Logistic Regression

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathscr{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} \frac{1}{1-\pi_i}\left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \\ \frac{1}{\pi_i}\frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \end{cases} \tag{19}$$

$y == 0$

$$\frac{-\pi_i(1-\pi_i)}{1-\pi_i}x_j \tag{20}$$

$$-\pi_i x_j \tag{21}$$

$y == 1$

$$\tag{22}$$

Chain Rule for Logistic Regression

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathscr{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} \frac{1}{1-\pi_i}\left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \\ \frac{1}{\pi_i}\frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \end{cases} \tag{19}$$

$y == 0$

$$\frac{-\pi_i(1-\pi_i)}{1-\pi_i} x_j \tag{20}$$

$$-\pi_i x_j \tag{21}$$

$y == 1$

$$\frac{1}{\pi_i}\pi_i(1-\pi_i)x_j \tag{22}$$

(23)

Chain Rule for Logistic Regression

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathscr{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} \frac{1}{1-\pi_i}\left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \\ \frac{1}{\pi_i}\frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \end{cases} \tag{19}$$

$y == 0$

$$\frac{-\pi_i(1-\pi_i)}{1-\pi_i} x_j \tag{20}$$

$$-\pi_i x_j \tag{21}$$

$y == 1$

$$\frac{1}{\pi_i}\pi_i(1-\pi_i)x_j \tag{22}$$

$$(1-\pi_i)x_j \tag{23}$$

Chain Rule for Logistic Regression

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathscr{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} \frac{1}{1-\pi_i}\left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \\ \frac{1}{\pi_i}\frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \end{cases} \tag{19}$$

$y == 0$

$$\frac{-\pi_i(1-\pi_i)}{1-\pi_i} x_j \tag{20}$$

$$-\pi_i x_j \tag{21}$$

$y == 1$

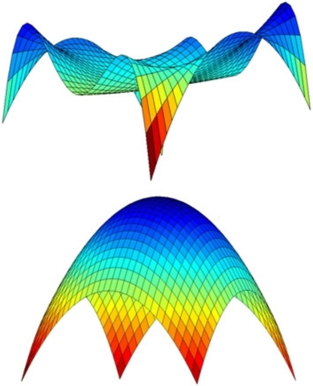
$$\frac{1}{\pi_i}\pi_i(1-\pi_i)x_j \tag{22}$$

$$(1-\pi_i)x_j \tag{23}$$

Merge these two cases

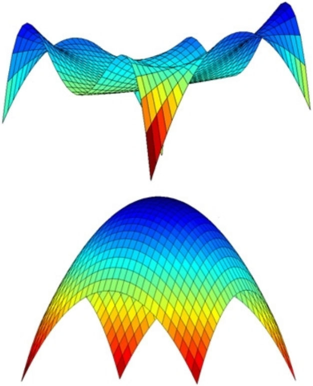
$$\frac{\partial \mathscr{L}_i}{\partial \beta_j} = (y_i - \pi_i)x_j. \tag{24}$$

Convexity



- Convex function
- Doesn't matter where you start, if you "walk up" objective

Convexity



- Convex function
- Doesn't matter where you start, if you "walk up" objective
- Gradient!

Gradient for Logistic Regression

Gradient

$$\nabla_\beta \mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \ldots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n} \right] \tag{25}$$

Update

$$\Delta\beta \equiv \lambda \nabla_\beta \mathscr{L}(\vec{\beta}) \tag{26}$$

$$\beta_i' \leftarrow \beta_i + \lambda \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{27}$$

Gradient for Logistic Regression

Gradient

$$\nabla_\beta \mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \ldots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n}\right] \tag{25}$$

Update

$$\Delta\beta \equiv \lambda \nabla_\beta \mathscr{L}(\vec{\beta}) \tag{26}$$

$$\beta_i' \leftarrow \beta_i + \lambda \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{27}$$

We're doing gradient ascent here, flip sign for descent

Gradient for Logistic Regression

Gradient

$$\nabla_{\beta}\mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \ldots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n}\right] \tag{25}$$

Update

$$\Delta\beta \equiv \lambda \nabla_{\beta}\mathscr{L}(\vec{\beta}) \tag{26}$$

$$\beta_i' \leftarrow \beta_i + \lambda \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{27}$$

λ: step size, must be greater than zero

Gradient for Logistic Regression

Gradient

$$\nabla_\beta \mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \ldots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n} \right] \tag{25}$$

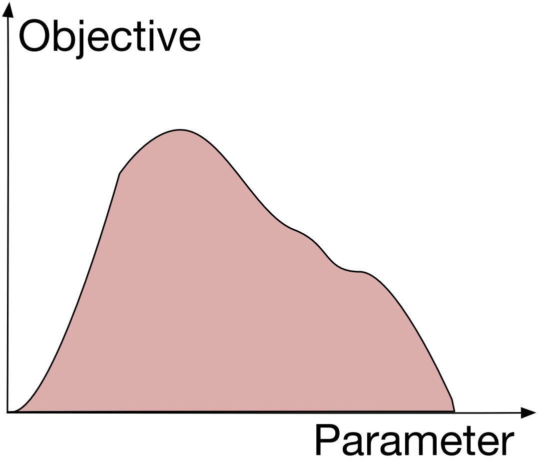
Update

$$\Delta \beta \equiv \lambda \nabla_\beta \mathscr{L}(\vec{\beta}) \tag{26}$$

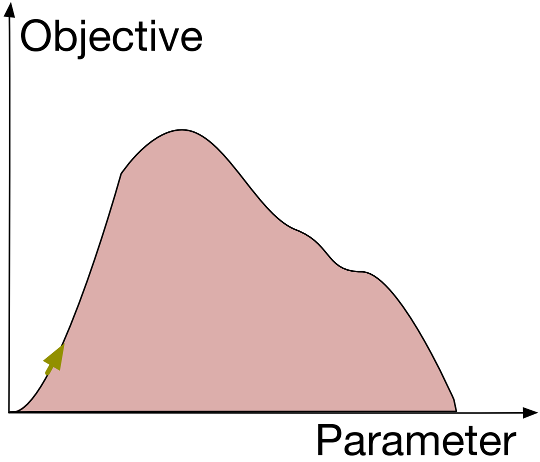
$$\beta_i' \leftarrow \beta_i + \lambda \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{27}$$

NB: Conjugate gradient is usually better, but harder to implement

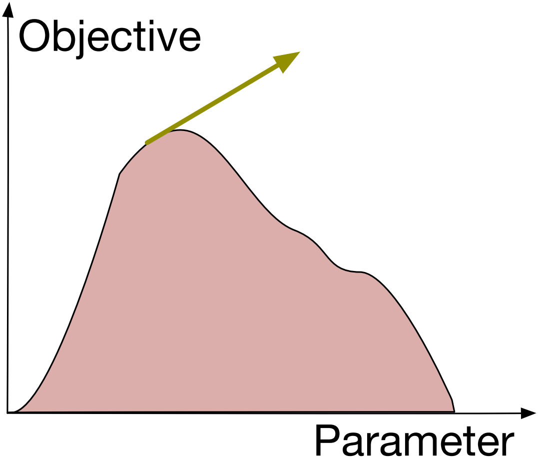
Choosing Step Size



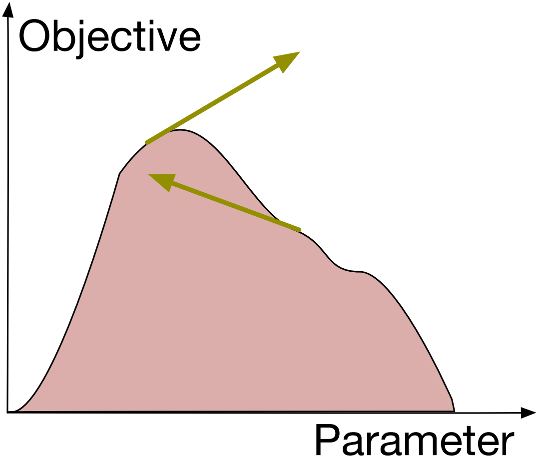
Choosing Step Size



Choosing Step Size



Choosing Step Size



Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg\max_{\beta} \ln\left[p(y^{(j)} \,|\, x^{(j)}, \beta)\right] \tag{28}$$

Regularized

$$\beta^* = \arg\max_{\beta} \ln\left[p(y^{(j)} \,|\, x^{(j)}, \beta)\right] - \mu \sum_i \beta_i^2 \tag{29}$$

Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg\max_{\beta} \ln\left[p(y^{(j)} \mid x^{(j)}, \beta)\right] \tag{28}$$

Regularized

$$\beta^* = \arg\max_{\beta} \ln\left[p(y^{(j)} \mid x^{(j)}, \beta)\right] - \mu \sum_i \beta_i^2 \tag{29}$$

μ is "regularization" parameter that trades off between likelihood and having small parameters

Stochastic Gradient for Regularized Regression

$$\mathscr{L} = \log p(y \mid x; \beta) - \mu \sum_j \beta_j^2 \quad (30)$$

Stochastic Gradient for Regularized Regression

$$\mathcal{L} = \log p(y \mid x; \beta) - \mu \sum_j \beta_j^2 \tag{30}$$

Taking the derivative (with respect to example x_i)

$$\frac{\partial \mathcal{L}}{\partial \beta_j} = (y_i - \pi_i) x_j - 2\mu\beta_j \tag{31}$$

Approximating the Gradient

- Our datasets are big (to fit into memory)
- …or data are changing / streaming

Approximating the Gradient

- Our datasets are big (to fit into memory)
- ...or data are changing / streaming
- Hard to compute true gradient

$$\mathscr{L}(\beta)_t = \mathscr{L}(\beta)_t + \lambda \mathbb{E}_x [\nabla \mathscr{L}(\beta, x)] \tag{32}$$

- Average over all observations (**batch**)

Approximating the Gradient

- Our datasets are big (to fit into memory)
- ...or data are changing / streaming
- Hard to compute true gradient

$$\mathscr{L}(\beta)_t = \mathscr{L}(\beta)_t + \lambda \mathbb{E}_x \left[\nabla \mathscr{L}(\beta, x) \right] \tag{32}$$

- Average over all observations (**batch**)
- What if we compute an update just from a few or even one observation? (**mini-batch**)

Getting to Union Station

Pretend it's a pre-smartphone world and you want to get to Union Station

Stochastic Gradient for Logistic Regression

Given a **single observation** (mini-batch k=1) x_i chosen at random from the dataset,

$$\beta_j \leftarrow \beta_j' + \lambda\left(x_{ij}\left[y_i - \pi_i\right]\right) \tag{33}$$

Stochastic Gradient for Logistic Regression

Given a **single observation** (mini-batch k=1) x_i chosen at random from the dataset,

$$\beta_j \leftarrow \beta_j' + \lambda\left(x_{ij}\left[y_i - \pi_i\right]\right) \tag{33}$$

Stochastic Gradient for Logistic Regression

Given a **single observation** (mini-batch k=1) x_i chosen at random from the dataset,

$$\beta_j \leftarrow \beta_j' + \lambda\left(x_{ij}\left[y_i - \pi_i\right]\right) \tag{33}$$

Stochastic Gradient for Logistic Regression

Given a **single observation** (mini-batch k=1) x_i chosen at random from the dataset,

$$\beta_j \leftarrow \beta_j' + \lambda\left(x_{ij}\left[y_i - \pi_i\right]\right) - \mu\beta_j'\right) \tag{33}$$

Algorithm (Unregularized)

1. Initialize a vector $\vec{\beta}$ to be all zeros
2. For $t = 1, \ldots, T$
   - For each example $\vec{x}_i, y_i$ and feature j:
     - Compute $\pi_i \equiv \Pr(y_i = 1 \mid \vec{x}_i)$
     - Set $\beta[j] = \beta[j]' + \lambda(y_i - \pi_i)x_i$
3. Output the parameters $\beta_1, \ldots, \beta_d$.

Algorithm (Unregularized)

1. Initialize a vector $\vec{\beta}$ to be all zeros
2. For $t = 1, \ldots, T$
   - For each example $\vec{x}_i, y_i$ and feature j:
     - Compute $\pi_i \equiv \Pr(y_i = 1 \,|\, \vec{x}_i)$
     - Set $\beta[j] = \beta[j]' + \lambda(y_i - \pi_i)x_i$
3. Output the parameters $\beta_1, \ldots, \beta_d$.

Algorithm (Unregularized)

1. Initialize a vector $\vec{\beta}$ to be all zeros
2. For $t = 1, \dots, T$
   - For each example $\vec{x}_i, y_i$ and feature j:
     - Compute $\pi_i \equiv \Pr(y_i = 1 \mid \vec{x}_i)$
     - Set $\beta[j] = \beta[j]' + \lambda(y_i - \pi_i)x_i$
3. Output the parameters $\beta_1, \dots, \beta_d$.

Courses, Lectures, Exercises and More

http://boydgraber.org