



Slides adapted from Emily Fox

Introduction to Machine Learning

Machine Learning: Jordan Boyd-Graber

University of Maryland

LOGISTIC REGRESSION FROM TEXT

Reminder: Logistic Regression

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

$$P(Y = 1|X) = \frac{\exp[\beta_0 + \sum_i \beta_i X_i]}{1 + \exp[\beta_0 + \sum_i \beta_i X_i]} \quad (2)$$

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

Logistic Regression: Objective Function

$$\mathcal{L} \equiv \ln p(Y|X, \beta) = \sum_j \ln p(y^{(j)} | x^{(j)}, \beta) \quad (3)$$

$$= \sum_j y^{(j)} \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) - \ln \left[1 + \exp \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) \right] \quad (4)$$

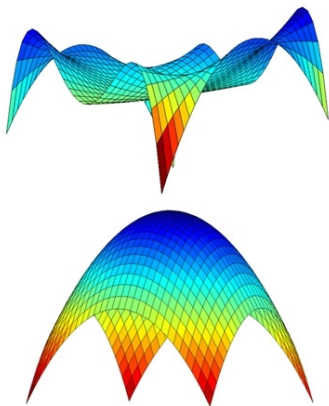
Logistic Regression: Objective Function

$$\mathcal{L} \equiv \ln p(Y|X, \beta) = \sum_j \ln p(y^{(j)} | x^{(j)}, \beta) \quad (3)$$

$$= \sum_j y^{(j)} \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) - \ln \left[1 + \exp \left(\beta_0 + \sum_i \beta_i x_i^{(j)} \right) \right] \quad (4)$$

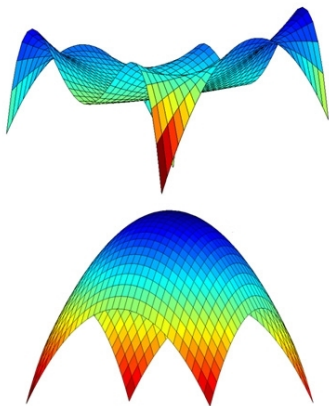
Training data (y, x) are fixed. Objective function is a function of β ... what values of β give a good value.

Convexity



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

Convexity

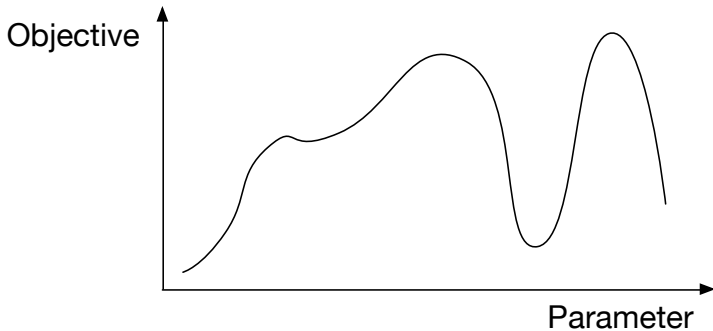


- Convex function
- Doesn't matter where you start, if you "walk up" objective
- Gradient!

Gradient Descent (non-convex)

Goal

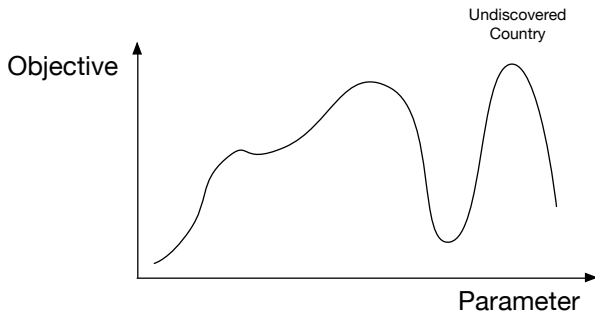
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

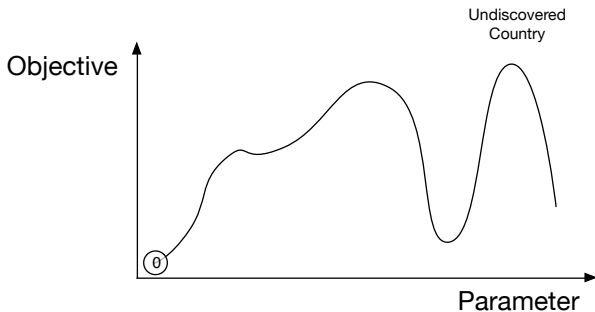
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

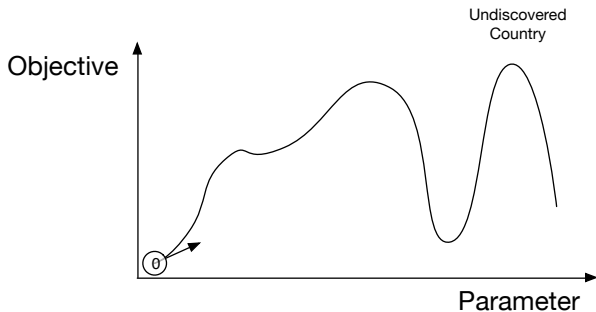
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

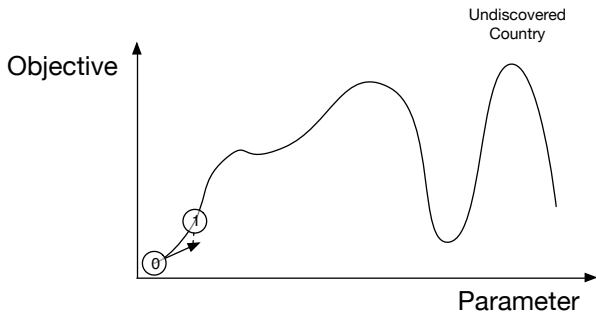
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

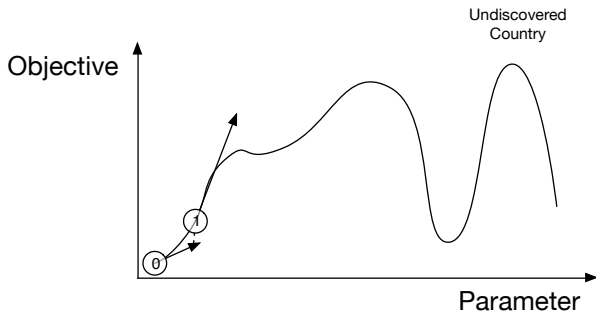
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

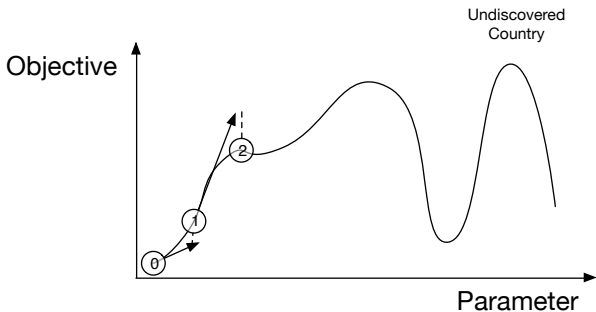
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

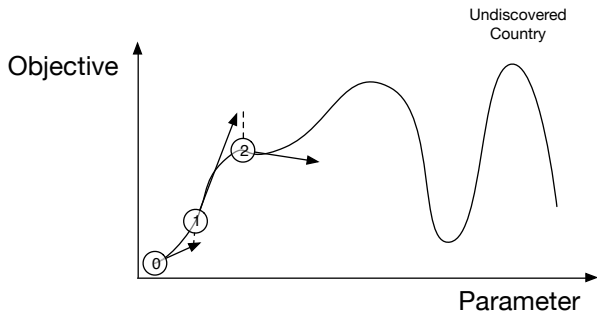
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

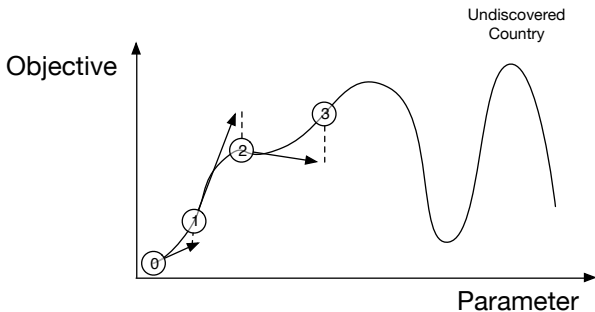
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

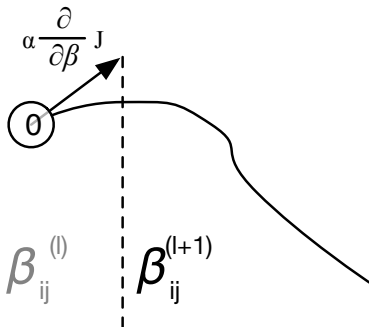
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

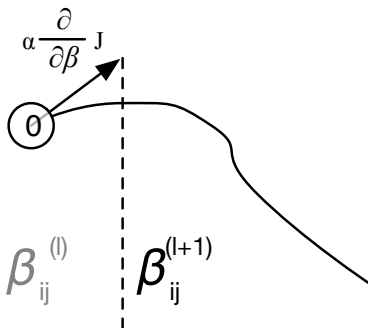
Optimize log likelihood with respect to variables β



Gradient Descent (non-convex)

Goal

Optimize log likelihood with respect to variables β



Luckily, (vanilla) logistic regression is convex

Gradient for Logistic Regression

To ease notation, let's define

$$\pi_i = \frac{\exp \beta^T x_i}{1 + \exp \beta^T x_i} \quad (5)$$

Our objective function is

$$\mathcal{L} = \sum_i \log p(y_i | x_i) = \sum_i \mathcal{L}_i = \sum_i \begin{cases} \log \pi_i & \text{if } y_i = 1 \\ \log(1 - \pi_i) & \text{if } y_i = 0 \end{cases} \quad (6)$$

Taking the Derivative

Apply chain rule:

$$\frac{\partial \mathcal{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathcal{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} \frac{1}{\pi_i} \frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \\ \frac{1}{1-\pi_i} \left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \end{cases} \quad (7)$$

If we plug in the derivative,

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i(1 - \pi_i)x_j, \quad (8)$$

we can merge these two cases

$$\frac{\partial \mathcal{L}_i}{\partial \beta_j} = (y_i - \pi_i)x_j. \quad (9)$$

Gradient for Logistic Regression

Gradient

$$\nabla_{\beta} \mathcal{L}(\vec{\beta}) = \left[\frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_0}, \dots, \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_n} \right] \quad (10)$$

Update

$$\Delta \beta \equiv \eta \nabla_{\beta} \mathcal{L}(\vec{\beta}) \quad (11)$$

$$\beta'_i \leftarrow \beta_i + \eta \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_i} \quad (12)$$

Gradient for Logistic Regression

Gradient

$$\nabla_{\beta} \mathcal{L}(\vec{\beta}) = \left[\frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_0}, \dots, \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_n} \right] \quad (10)$$

Update

$$\Delta \beta \equiv \eta \nabla_{\beta} \mathcal{L}(\vec{\beta}) \quad (11)$$

$$\beta'_i \leftarrow \beta_i + \eta \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_i} \quad (12)$$

Why are we adding? What would we do if we wanted to do **descent**?

Gradient for Logistic Regression

Gradient

$$\nabla_{\beta} \mathcal{L}(\vec{\beta}) = \left[\frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_0}, \dots, \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_n} \right] \quad (10)$$

Update

$$\Delta \beta \equiv \eta \nabla_{\beta} \mathcal{L}(\vec{\beta}) \quad (11)$$

$$\beta'_i \leftarrow \beta_i + \eta \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_i} \quad (12)$$

η : step size, must be greater than zero

Gradient for Logistic Regression

Gradient

$$\nabla_{\beta} \mathcal{L}(\vec{\beta}) = \left[\frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_0}, \dots, \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_n} \right] \quad (10)$$

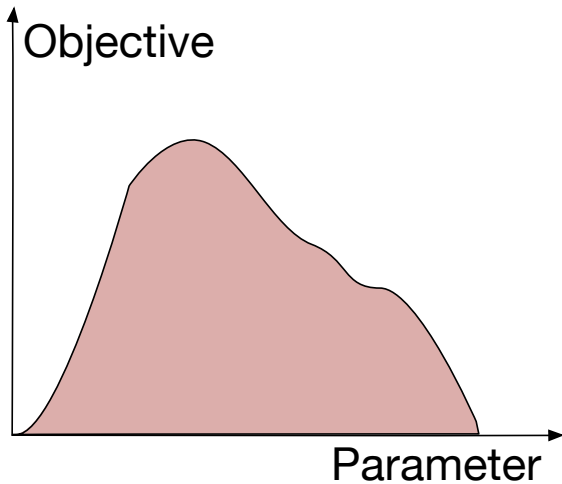
Update

$$\Delta \beta \equiv \eta \nabla_{\beta} \mathcal{L}(\vec{\beta}) \quad (11)$$

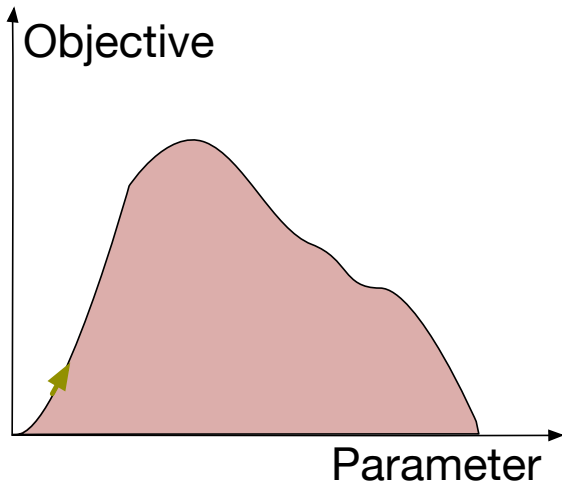
$$\beta'_i \leftarrow \beta_i + \eta \frac{\partial \mathcal{L}(\vec{\beta})}{\partial \beta_i} \quad (12)$$

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

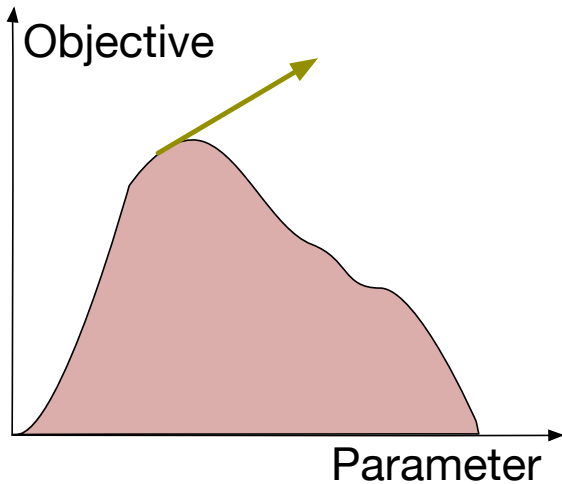
Choosing Step Size



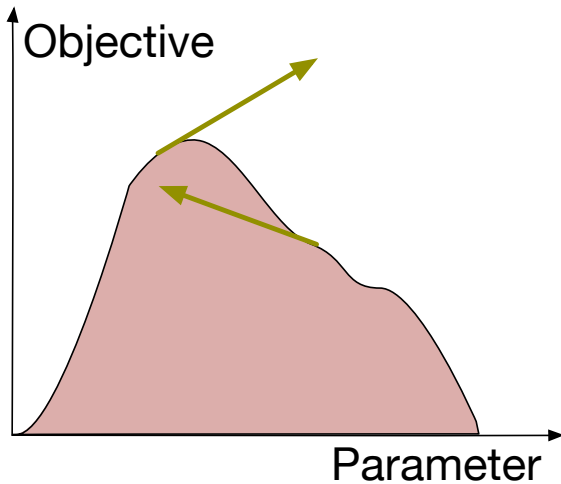
Choosing Step Size



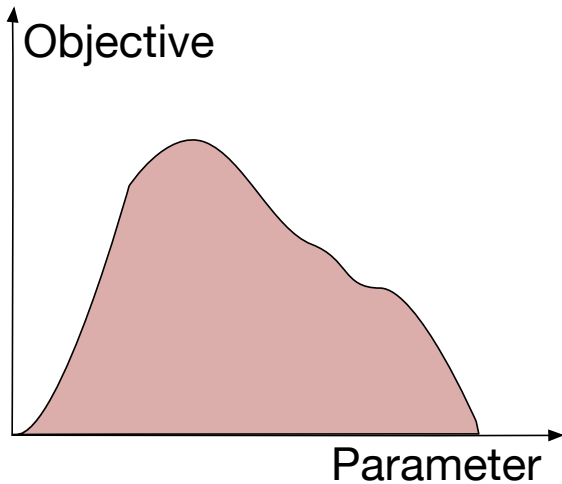
Choosing Step Size



Choosing Step Size



Choosing Step Size



Remaining issues

- When to stop?
- What if β keeps getting bigger?

Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg \max_{\beta} \ln [p(y^{(j)} | x^{(j)}, \beta)] \quad (13)$$

Regularized

$$\beta^* = \arg \max_{\beta} \ln [p(y^{(j)} | x^{(j)}, \beta)] - \mu \sum_i \beta_i^2 \quad (14)$$

Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg \max_{\beta} \ln [p(y^{(j)} | x^{(j)}, \beta)] \quad (13)$$

Regularized

$$\beta^* = \arg \max_{\beta} \ln [p(y^{(j)} | x^{(j)}, \beta)] - \mu \sum_i \beta_i^2 \quad (14)$$

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

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

$$\mathcal{L}(\beta) \equiv \mathbb{E}_x [\nabla \mathcal{L}(\beta, x)] \quad (15)$$

- Average over all observations

Approximating the Gradient

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

$$\mathcal{L}(\beta) \equiv \mathbb{E}_x [\nabla \mathcal{L}(\beta, x)] \quad (15)$$

- Average over all observations
- What if we compute an update just from one observation?

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** x_i chosen at random from the dataset,

$$\beta_j \leftarrow \beta_j' + \eta \left(-\mu \beta_j' + x_{ij} [y_i - \pi_i] \right) \quad (16)$$

Stochastic Gradient for Logistic Regression

Given a **single observation** x_i chosen at random from the dataset,

$$\beta_j \leftarrow \beta'_j + \eta \left(-\mu \beta'_j + x_{ij} [y_i - \pi_i] \right) \quad (16)$$

Examples in class.

Stochastic Gradient for Regularized Regression

$$\mathcal{L} = \log p(y|x; \beta) - \mu \sum_j \beta_j^2 \quad (17)$$

Stochastic Gradient for Regularized Regression

$$\mathcal{L} = \log p(y | x; \beta) - \mu \sum_j \beta_j^2 \quad (17)$$

Taking the derivative (with respect to example x_i)

$$\frac{\partial \mathcal{L}}{\partial \beta_j} = (y_i - \pi_i) x_j - 2\mu \beta_j \quad (18)$$

Algorithm

1. Initialize a vector B to be all zeros
2. For $t = 1, \dots, T$
 - For each example \vec{x}_i, y_i and feature j :
 - Compute $\pi_i \equiv \Pr(y_i = 1 | \vec{x}_i)$
 - Set $\beta[j] = \beta[j]' + \lambda(y_i - \pi_i)x_i$
3. Output the parameters β_1, \dots, β_d .

Proofs about Stochastic Gradient

- Depends on convexity of objective and how close ϵ you want to get to actual answer
- Best bounds depend on changing η over time and **per dimension** (not all features created equal)

In class

- Your questions!
- Working through simple example
- Prepared for logistic regression homework