Machine Learning: Chenhao Tan

University of Colorado Boulder

LECTURE 9

Slides adapted from Jordan Boyd-Graber, Chris Ketelsen

Logistics

- HW2 available on Github, due in 9 days
- Office hour logistics
- View your class as a community, Piazza

Learning objectives

- Understand gradient descent
- Understand structural risk minimization

Outline

Objective function

Gradient Descent

Empirical Risk Minimization

Outline

Reminder: Logistic Regression

$$P(Y=0 \mid \boldsymbol{x}) = \frac{1}{1+\exp\left[\beta_0 + \sum_j \beta_j \boldsymbol{x}_j\right]} \tag{1}$$

$$P(Y=1 \mid \boldsymbol{x}) = \frac{\exp\left[\beta_0 + \sum_j \beta_j \boldsymbol{x}_j\right]}{1+\exp\left[\beta_0 + \sum_j \beta_j \boldsymbol{x}_j\right]} \tag{2}$$

- Discriminative prediction: $P(y \mid \boldsymbol{x})$
- Classification uses: sentiment analysis, spam detection
- What we didn't talk about is how to learn β from data

Logistic Regression: Objective Function

One idea: find the parameter that maximize the likelihood of observing the training data.

Logistic Regression: Objective Function

One idea: find the parameter that maximize the likelihood of observing the training data.
Maximize likelihood

$$\text{Obj} \equiv \log P(Y \mid X, \beta) = \sum_i \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, \beta))$$
$$= \sum_i y^{(i)} \left(\beta_0 + \sum_j \beta_j \boldsymbol{x}_j^{(i)}\right) - \log\left[1 + \exp\left(\beta_0 + \sum_j \beta_j \boldsymbol{x}_j^{(i)}\right)\right]$$

Logistic Regression: Objective Function

Minimize negative log likelihood (loss)

$$\begin{aligned}\mathscr{L} \equiv -\log P(Y \mid X, \beta) &= -\sum_i \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, \beta)) \\ &= \sum_i -y^{(i)} \left(\beta_0 + \sum_j \beta_i \boldsymbol{x}_j^{(i)}\right) + \log\left[1 + \exp\left(\beta_0 + \sum_j \beta_i \boldsymbol{x}_j^{(i)}\right)\right]\end{aligned}$$

Logistic Regression: Objective Function

Minimize negative log likelihood (loss)

$$
\begin{aligned}
\mathscr{L} \equiv -\log P(Y \mid X, \beta) &= -\sum_i \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, \beta)) \\
&= \sum_i -y^{(i)} \left(\beta_0 + \sum_j \beta_i \boldsymbol{x}_j^{(i)}\right) + \log\left[1 + \exp\left(\beta_0 + \sum_j \beta_i \boldsymbol{x}_j^{(i)}\right)\right]
\end{aligned}
$$

Training data $\{(\boldsymbol{x}, y)\}$ are fixed. Objective function is a function of β ... what values of β give a good value?

Logistic Regression: Objective Function

Minimize negative log likelihood (loss)

$$\mathscr{L} \equiv -\log P(Y \mid X, \beta) = -\sum_i \log P(y^{(i)} \mid \boldsymbol{x}^{(i)}, \beta))$$

$$= \sum_i -y^{(i)} \left(\beta_0 + \sum_j \beta_i \boldsymbol{x}_j^{(i)} \right) + \log \left[1 + \exp \left(\beta_0 + \sum_j \beta_i \boldsymbol{x}_j^{(i)} \right) \right]$$

Training data $\{(\boldsymbol{x}, y)\}$ are fixed. Objective function is a function of β ... what values of β give a good value?

$$\beta^* = \arg\min_{\beta} \mathscr{L}(\beta)$$

Convexity

$\mathscr{L}(\beta)$ is convex for logistic regression.

Proof.

- Logistic loss $-yv + \log(1 + \exp(v))$ is convex.
- Composition with linear function maintains convexity.
- Sum of convex functions is convex.

$\square$

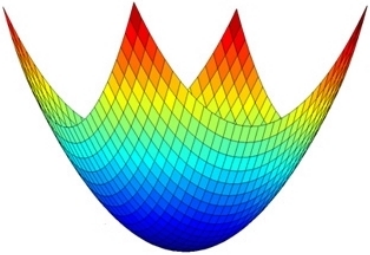
Outline

Objective function

Gradient Descent

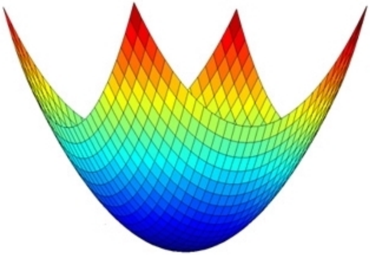
Empirical Risk Minimization

Convexity



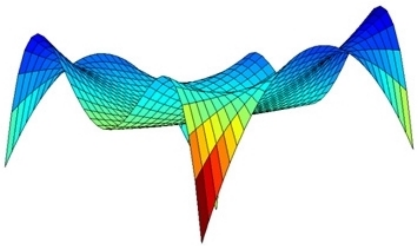
- Convex function
- Doesn't matter where you start, if you go down along the gradient

Convexity



- Convex function
- Doesn't matter where you start, if you go down along the gradient
- Gradient!

Convexity

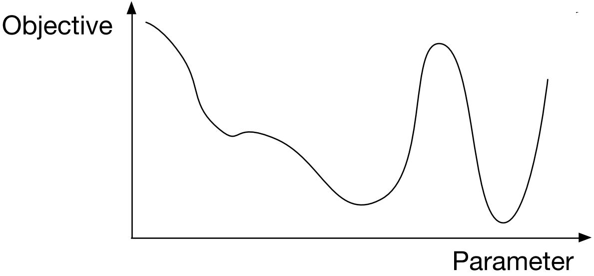


- It would have been much harder if this is not convex.

Gradient Descent (non-convex)

Goal

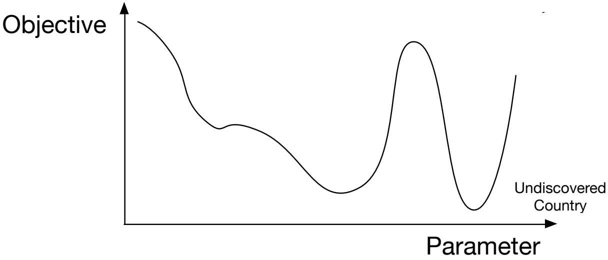
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

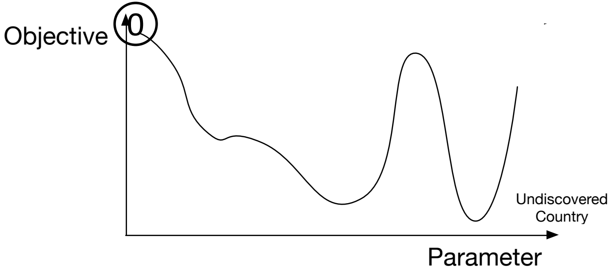
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

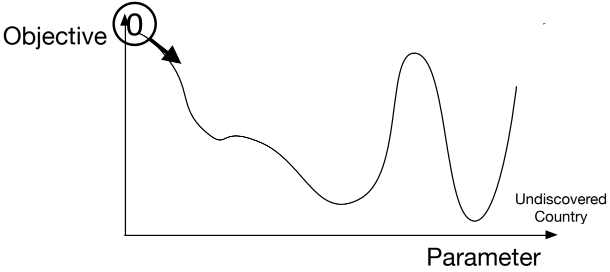
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

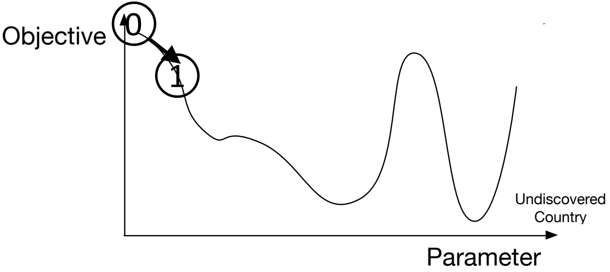
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

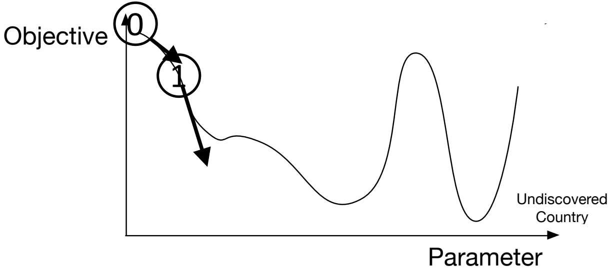
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

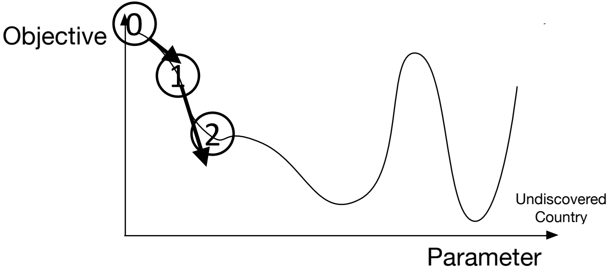
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

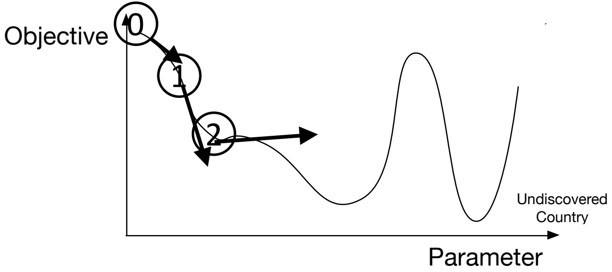
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

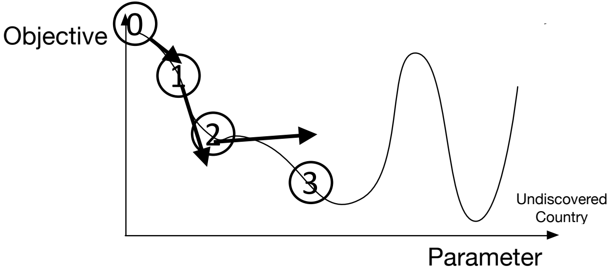
Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

Optimize loss function with respect to variables β



Gradient Descent (non-convex)

Goal

Optimize loss function with respect to variables β

$$\beta_j^{l+1} = \beta_j^l - \eta \frac{\partial \mathscr{L}}{\partial \beta_j}$$

Gradient Descent (non-convex)

Goal

Optimize loss function with respect to variables β

$$\beta_j^{l+1} = \beta_j^l - \eta \frac{\partial \mathscr{L}}{\partial \beta_j}$$

Luckily, (vanilla) logistic regression is convex

Gradient for Logistic Regression

To ease notation, let's define

$$\pi_i = \frac{\exp \beta^T x_i}{1 + \exp \beta^T x_i} \tag{3}$$

Our objective function is

$$\mathscr{L} = -\sum_i \log p(y_i \,|\, x_i) = \sum_i \mathscr{L}_i = \sum_i \begin{cases} -\log \pi_i & \text{if } y_i = 1 \\ -\log(1 - \pi_i) & \text{if } y_i = 0 \end{cases} \tag{4}$$

Taking the Derivative

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \frac{\partial \mathscr{L}_i(\vec{\beta})}{\partial \beta_j} = \sum_i \begin{cases} -\frac{1}{\pi_i}\frac{\partial \pi_i}{\partial \beta_j} & \text{if } y_i = 1 \\ -\frac{1}{1-\pi_i}\left(-\frac{\partial \pi_i}{\partial \beta_j}\right) & \text{if } y_i = 0 \end{cases} \tag{5}$$

If we plug in the derivative,

$$\frac{\partial \pi_i}{\partial \beta_j} = \pi_i(1-\pi_i)x_j, \tag{6}$$

we can merge these two cases

$$\frac{\partial \mathscr{L}_i}{\partial \beta_j} = -(y_i - \pi_i)x_j. \tag{7}$$

Gradient for Logistic Regression

Gradient

$$\nabla_\beta \mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \ldots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n} \right] \tag{8}$$

Update

$$\Delta \beta \equiv \eta \nabla_\beta \mathscr{L}(\vec{\beta}) \tag{9}$$

$$\beta_i' \leftarrow \beta_i - \eta \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{10}$$

Gradient for Logistic Regression

Gradient

$$\nabla_\beta \mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \dots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n}\right] \tag{8}$$

Update

$$\Delta\beta \equiv \eta \nabla_\beta \mathscr{L}(\vec{\beta}) \tag{9}$$

$$\beta_i' \leftarrow \beta_i - \eta \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{10}$$

η: step size, must be greater than zero

Gradient for Logistic Regression

Gradient

$$\nabla_\beta \mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \dots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n}\right] \tag{8}$$

Update

$$\Delta\beta \equiv \eta \nabla_\beta \mathscr{L}(\vec{\beta}) \tag{9}$$

$$\beta_i' \leftarrow \beta_i - \eta \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{10}$$

Overfitting

- It is not ideal to maximize the likelihood of training data

Overfitting

- It is not ideal to maximize the likelihood of training data
  - When to stop?
  - Simple models (avoid β to get too big)

Overfitting

- It is not ideal to maximize the likelihood of training data
  - When to stop?
  - Simple models (avoid β to get too big)
    Regularization

Outline

Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg\min_{\beta} -\ln\left[p(y^{(j)} \mid x^{(j)}, \beta)\right] \tag{11}$$

Regularized

$$\beta^* = \arg\min_{\beta} -\ln\left[p(y^{(j)} \mid x^{(j)}, \beta)\right] + \frac{1}{2}\lambda \sum_i \beta_i^2 \tag{12}$$

Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg\min_{\beta} - \ln \left[p(y^{(j)} \,|\, x^{(j)}, \beta) \right] \tag{11}$$

Regularized

$$\beta^* = \arg\min_{\beta} - \ln \left[p(y^{(j)} \,|\, x^{(j)}, \beta) \right] + \frac{1}{2}\lambda \sum_i \beta_i^2 \tag{12}$$

λ is the “regularization” parameter (a hyperparameter) that trades off between likelihood and having small parameters

Alternative view of regularization

Can also get to regularization by putting prior beliefs on parameters

$$p(\beta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \beta)p(\beta)$$

Then MAP estimate for β is $\hat{\beta}$ which maximizes posterior

Alternative view of regularization

Can also get to regularization by putting prior beliefs on parameters

$$p(\beta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \beta)p(\beta)$$

Then MAP estimate for β is $\hat{\beta}$ which maximizes posterior

Ridge: Assume Gaussian prior $p(\beta_j) = \mathcal{N}(\beta_j \mid 0, \tau^2)$, we will obtain the same regularized objective function
You can learn more about this view in “Bayesian statistics”

Risk minimization

$$\min_{\beta} \sum_{i} \ell(y^{(i)}, h_{\beta}(\boldsymbol{x}^{(i)})) + \lambda R(\beta)$$

Risk minimization

$$\min_{\beta} \sum_i \ell(y^{(i)}, h_\beta(\boldsymbol{x}^{(i)})) + \lambda R(\beta)$$

Loss functions (ℓ)

Describe how well the model fits the training data

- $-y\hat{y} + log(1 + exp(\hat{y}))$

Regularization (R)

Control the complexity of the model

- $||\beta||^2 = \sum_j \beta_j^2$

Risk minimization

$$\min_{\beta} \sum_{i} \ell(y^{(i)}, h_{\beta}(\boldsymbol{x}^{(i)})) + \lambda R(\beta)$$

Loss functions (ℓ)

Describe how well the model fits the training data

- $-y\hat{y} + log(1 + exp(\hat{y}))$
- $(y - \hat{y})^2$
- $\max\{0, 1 - y\hat{y}\}$

Regularization (R)

Control the complexity of the model

- $||\beta||^2 = \sum_j \beta_j^2$
- $||\beta||_p = \left(\sum_j |\beta_j|^p\right)^{\frac{1}{p}}$
  - ℓ_1-regularization: $\sum_j |\beta_j|$

Summary

- Follow the gradient to fit the logistic regression model
- Most machine learning methods fall into the framework of (loss + regularization)