Introduction, Gradient Descent, and A1

CSE 447 / 517

January 6th, 2022 (Week 1)

Eisenstein (2019) 2, Appendix B

Logistics

- Submit the Academic Integrity Form on Canvas
- Submit the poll of virtual sections on Canvas by **Friday (1/7) 11:59 PM**
- Assignment 1 (A1) is due on **Wednesday, 1/12**
- Quiz 1 is due on **Monday, 1/10**
  - The quiz will be on **multinomial logistic regression**.
  - It is graded based on **completion** and contributes to your participation points.
  - We will go over the quiz in the section next week.

Agenda

- Binary Logistic Regression

- Gradient Descent

- A1 Overview / Q&A

Feature Vectors

- The features fully determine what a learned model “sees” about an example.
- We often stack the features into a feature vector:

$$\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^d$$

  which “embeds” the input x in d-dimensional space
- Example feature from lecture: word frequencies, idf… You can stack them to be a feature vector!

Logistic Regression

A logistic regression model usually has:

- A collection of feature functions, denoted $\phi_1, \ldots \phi_d$

  each mapping $\mathcal{V}^* \to \mathbb{R}$.

- A coefficient or "weight" for every feature, denoted $\theta_1, \ldots, \theta_d$

  each $\in \mathbb{R}$

From lecture slides 32

Binary Logistic Regression

The label set is $\mathcal{L} = \{+1, -1\}$.

the labels are arbitrary and can be changed as long as the classify() function is modified accordingly!

$$\text{score}_{\text{LR}}(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{j=1}^{d} \theta_j \phi_j(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

$$\text{classify}_{\text{LR}}(\boldsymbol{x}) = \text{sign}(\text{score}_{\text{LR}}(\boldsymbol{x}; \boldsymbol{\theta}))$$

Binary Logistic Regression

$$
\begin{aligned}
p_{LR}(Y = y \,|\, \mathbf{X} = \mathbf{x},\, \theta) &= \sigma(y \cdot \mathrm{score}_{LR}(\mathbf{x},\, \theta)) \\
&= \sigma\left(y \cdot \left(\theta^\top \phi(\mathbf{x})\right)\right) \\
&= \frac{1}{1 + e^{-(y \cdot (\theta^\top \phi(\mathbf{x})))}}
\end{aligned}
$$

from Lecture Slide 40

apply the definition of the score function

apply the definition of the standard logistic function

Symbol	Definition	Scalar / Vector
x	Input	Vector
y	Output	Scalar
$\boldsymbol{\theta}$	Parameters	Vector
$\boldsymbol{\phi}$**(x)**	Feature vector (Lecture Slide 31)	Vector

Gradient Descent

Goal: Given a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, find the weights θ* by maximum likelihood estimation.

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \prod_{i=1}^{n} p_{\mathrm{LR}}(Y = y_i \mid \boldsymbol{X} = \boldsymbol{x}_i; \boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^{n} \log p_{\mathrm{LR}}(Y = y_i \mid \boldsymbol{X} = \boldsymbol{x}_i; \boldsymbol{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^{n} \underbrace{-\log p_{\mathrm{LR}}(Y = y_i \mid \boldsymbol{X} = \boldsymbol{x}_i; \boldsymbol{\theta})}_{\text{sometimes called ``log loss'' or ``cross entropy''}}
\end{aligned}$$

Gradient Descent

Goal: Given a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, find the weights θ* by maximum likelihood estimation.

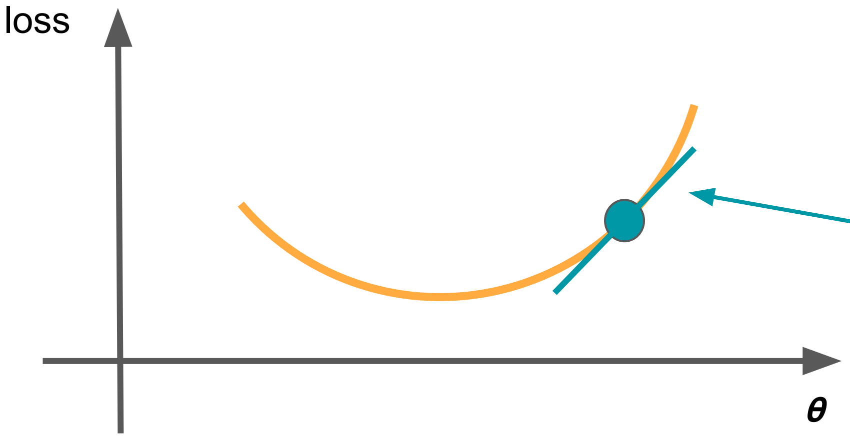
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \underbrace{\sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}_i)\right)\right)}_{\text{loss}(\boldsymbol{\theta})}$$

apply the definition of p_{LR} that we found in Section Slide 4

From Lecture Slide 48

Gradient Descent

Big idea: minimize the loss by “optimization along the (negative) gradient”.



“Gradient” aka the 1st derivative is the slope.

See Eisenstein pg. 37

Gradient Descent

Step 1: finding the gradient.

Start from the loss function:

$$\text{loss} = \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \cdot \theta^\top \phi(\mathbf{x_i}))\right)$$

Differentiate with respect to the parameters:

$$\frac{\partial \text{loss}}{\partial \theta} = \sum_{i=1}^{n} \frac{\exp\left(-y_i \cdot \theta^\top \phi(\mathbf{x_i})\right)}{1+\exp(-y_i \cdot \theta^\top \phi(\mathbf{x_i}))} \cdot -y_i \cdot \phi(x_i)$$

Gradient Descent

Step 1: finding the gradient.

Simplify the gradient:

$$\begin{aligned}\frac{\partial \text{loss}}{\partial \theta} &= \sum_{i=1}^{n} \left(1 - \sigma\left(y_i \cdot \theta^\top \phi(\mathbf{x_i})\right)\right) \cdot -y_i \cdot \phi(\mathbf{x_i}) \\ &= \sum_{i=1}^{n} (1 - \sigma(y_i \cdot \text{score}_{\text{LR}}(\mathbf{x};\ \theta))) \cdot -y_i \cdot \phi(\mathbf{x_i}) \\ &= \sum_{i=1}^{n} (1 - \mathrm{p}_{\text{LR}}(Y = y_i \,|\, \mathbf{X} = \mathbf{x_i}, \theta)) \cdot -y_i \cdot \phi(\mathbf{x_i})\end{aligned}$$

Gradient Descent

Step 2: take a step.

Update the parameters:

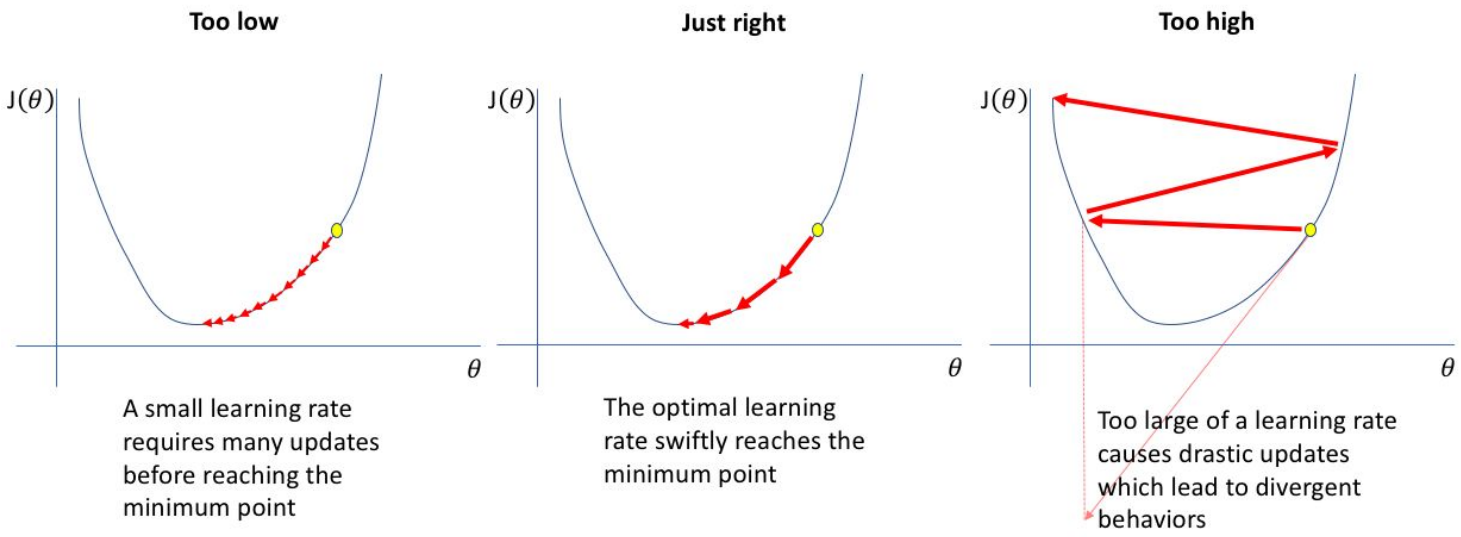
$$\theta \leftarrow \theta - \alpha \frac{\partial \text{loss}}{\partial \theta}$$

where α is the learning rate.

Step 3: repeat Step 1-2 until converge (i.e. loss basically stops decreasing).

Gradient Descent

Things to consider: how to choose learning rate? Another hyperparameter!



From https://www.jeremyjordan.me/nn-learning-rate/

Stochastic Gradient Descent

Input: initial value $\boldsymbol{\theta}$, number of epochs T, learning rate α

For $t \in \{1, \ldots, T\}$:

- Choose a random permutation π of $\{1, \ldots, N\}$.
- For $i \in \{1, \ldots, N\}$:

$$\boldsymbol{\theta} \leftarrow \mathbf{w} - \alpha \cdot \nabla_{\boldsymbol{\theta}} g_{\pi(i)}$$

Output: $\boldsymbol{\theta}$

From Lecture Slide 53

A1 - Overview

Preparing the data:

- Randomly selected 400 samples, set them aside as test set (you never touch this until evaluation)
- Tokenization (split texts into “tokens”)

Build a classifier:

- Sentiment lexicon-based classifier
- Logistic regression classifier
    - You pick the text features
    - You have to implement gradient descent

Evaluate your model

- Use the test set to compute accuracy and F1 score

Test the significance (extra credit)

- See Eisenstein (2019) Section 4.4.3 (p.g. 84-87)