Machine Learning: Yoshinari Fujinuma

University of Colorado Boulder

LECTURE 9

Slides adapted from Chenhao Tan, Jordan Boyd-Graber, Chris Ketelsen

Learning objectives

- How to learn weights β for logistic regression

Outline

Objective function

Gradient Descent

Regularization

Outline

Objective function

Gradient Descent

Regularization

Reminder: Logistic Regression

Logistic (sigmoid) function σ is defined as

$$\sigma = \frac{1}{1 + \exp -\beta^T x} \tag{1}$$

$$P(Y = 1 \mid \boldsymbol{x}) = \sigma \tag{2}$$

$$P(Y = 0 \mid \boldsymbol{x}) = 1 - \sigma \tag{3}$$

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

Logistic Regression: Objective Function

Find the parameter that maximize the likelihood of observing the training data with N examples i.e., $(\boldsymbol{x}, y) = \{(\boldsymbol{x}_0, y_0), (\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$.

Logistic Regression: Objective Function

Find the parameter that maximize the likelihood of observing the training data with N examples i.e., $(\boldsymbol{x}, y) = \{(\boldsymbol{x}_0, y_0), (\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$.

$$
\begin{aligned}
\text{Likelihood} &= P(Y \mid X, \beta) \\
&= \prod_i P(y_i \mid \boldsymbol{x}_i, \beta) \\
&= \prod_i \begin{cases} \sigma_i & \text{if } y_i = 1 \\ 1 - \sigma_i & \text{if } y_i = 0 \end{cases}
\end{aligned}
$$

What is the problem of this likelihood function?

- Especially, considering $0 < \sigma < 1$?

Logistic Regression: Objective Function

Idea: Use the log-likelihood

Logistic Regression: Objective Function

Idea: Use the log-likelihood

$$
\begin{aligned}
\text{Log-Likelihood (LL)} &= \log P(Y \mid X, \beta) \\
&= \sum_i \log P(y_i \mid \boldsymbol{x}_i, \beta) \\
&= \sum_i \begin{cases} \log \sigma_i & \text{if } y_i = 1 \\ \log(1 - \sigma_i) & \text{if } y_i = 0 \end{cases}
\end{aligned}
$$

Logistic Regression: Objective Function

Maximize the log-likelihood or Minimize negative log likelihood (NLL) $\mathscr{L}$ is

$$\mathscr{L}(\beta) = -\sum_i \log P(y_i \mid \boldsymbol{x}_i, \beta)$$

Logistic Regression: Objective Function

Maximize the log-likelihood or Minimize negative log likelihood (NLL) $\mathscr{L}$ is

$$\mathscr{L}(\beta) = -\sum_i \log P(y_i \mid \boldsymbol{x}_i, \beta)$$

So back to the main question today.
What values should β be?

Logistic Regression: Objective Function

Maximize the log-likelihood or Minimize negative log likelihood (NLL) $\mathscr{L}$ is

$$\mathscr{L}(\beta) = -\sum_i \log P(y_i \mid \boldsymbol{x}_i, \beta)$$

So back to the main question today.
What values should β be?

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

Outline

Objective function

Gradient Descent

Regularization

Simple Example

Given a paramter w and the loss function L we want to optimize

$$L(w) = w^2$$

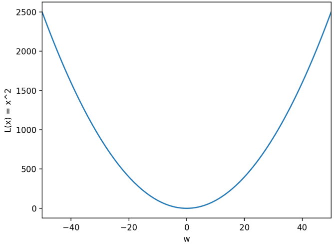
Assume $w = 20$, which way should I move to miminize the loss?

Simple Example

Given a paramter w and the loss function L we want to optimize

$$L(w) = w^2$$

Assume $w = 20$, which way should I move to miminize the loss?
Answer: Downhill, but the derivative $\frac{\partial L}{\partial w} = 2w$ tells you uphill. So let's take the negative i.e., $-2w$.

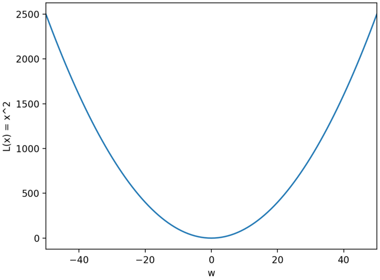


Simple Example

We now know which way is downhill, but how far we want to go?

Simple Example

We now know which way is downhill, but how far we want to go?
Another hyperparameter: a small step size called **learning rate** η to go.

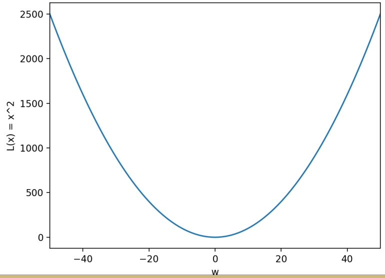


Simple Example

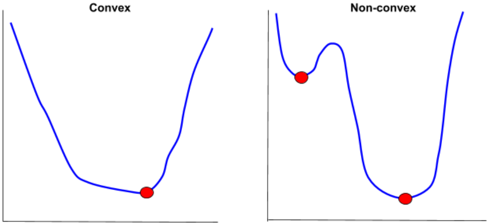
- $-\frac{\partial L}{\partial w} = -2w$ tells you the direction
- learning rate η tells you how far you want to go to minimize the loss

We update the parameter w to the updated parameter w' by

$$w' \leftarrow w - \eta \frac{\partial L}{\partial w}$$



Convexity



- NLL is convex
- Doesn't matter where you start, if you go down along the gradient

Image from `https://automaticaddison.com/how-to-choose-an-optimal-learning-rate-for-gradient-descent/`

Gradient for Logistic Regression

Again, σ is defined as

$$\sigma_i = \frac{1}{1 + \exp -\beta^T x_i} \tag{4}$$

Our objective function is

$$\mathscr{L}(\beta) = -\sum_i \log p(y_i \,|\, x_i) = \sum_i \begin{cases} -\log \sigma_i & \text{if } y_i = 1 \\ -\log(1 - \sigma_i) & \text{if } y_i = 0 \end{cases} \tag{5}$$

Taking the Derivative

Apply chain rule:

$$\frac{\partial \mathscr{L}}{\partial \beta_j} = \sum_i \begin{cases} -\frac{1}{\sigma_i} \frac{\partial \sigma_i}{\partial \beta_j} & \text{if } y_i = 1 \\ -\frac{1}{1-\sigma_i} \left(-\frac{\partial \sigma_i}{\partial \beta_j} \right) & \text{if } y_i = 0 \end{cases} \tag{6}$$

The derivative of logistic/sigmoid function σ with respect to β_j is,

$$\frac{\partial \sigma_i}{\partial \beta_j} = \sigma_i(1 - \sigma_i)x_{ij}, \tag{7}$$

we can merge two cases of $y_i = 0$ or $y_i = 1$ as

$$\frac{\partial \mathscr{L}_i}{\partial \beta_j} = -(y_i - \sigma_i)x_{ij}. \tag{8}$$

Gradient for Logistic Regression

Gradient

$$\nabla_\beta \mathscr{L}(\vec{\beta}) = \left[\frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_0}, \ldots, \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_n}\right] \tag{9}$$

Update

$$\Delta\beta = \eta \nabla_\beta \mathscr{L}(\vec{\beta}) \tag{10}$$

$$\beta_i' \leftarrow \beta_i - \eta \frac{\partial \mathscr{L}(\vec{\beta})}{\partial \beta_i} \tag{11}$$

Overfitting

- Maximize the likelihood of training data can have the risk of overfitting

Overfitting

- Maximize the likelihood of training data can have the risk of overfitting
  - When to stop?
  - Simple models (avoid β to get too big)

Overfitting

- Maximize the likelihood of training data can have the risk of overfitting
  - When to stop?
  - Simple models (avoid β to get too big)
    Regularization

Outline

Objective function

Gradient Descent

Regularization

Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg\min_{\beta} - \sum_i \log \left[p(y_i \,|\, \boldsymbol{x}_i, \beta) \right] \tag{12}$$

Regularized

$$\beta^* = \arg\min_{\beta} - \sum_i \log \left[p(y_i \,|\, \boldsymbol{x}_i, \beta) \right] + \frac{1}{2} \lambda \sum_j \beta_j^2 \tag{13}$$

Regularized Conditional Log Likelihood

Unregularized

$$\beta^* = \arg\min_{\beta} - \sum_i \log \left[p(y_i \,|\, \boldsymbol{x}_i, \beta) \right] \tag{12}$$

Regularized

$$\beta^* = \arg\min_{\beta} - \sum_i \log \left[p(y_i \,|\, \boldsymbol{x}_i, \beta) \right] + \frac{1}{2}\lambda \sum_j \beta_j^2 \tag{13}$$

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

Overview

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

Overview

$$\min_{\beta} \sum_i \ell(y_i, h_\beta(\boldsymbol{x}_i)) + \lambda R(\beta)$$

Loss functions (ℓ)

Describe how well the model fits the training data

- $y \log \hat{y} + (1 - y) \log(1 - \hat{y})$
- $(y - \hat{y})^2$

Regularization (R)

Control the complexity of the model

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