CS540 Introduction to Artificial Intelligence

Lecture 2

Young Wu

Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

June 29, 2022

Zero-One Loss Function

Motivation

- An objective function is needed to select the "best" $\hat{f}$. An example is the zero-one loss.

$$\hat{f} = \operatorname*{argmin}_f \sum_{i=1}^{n} \mathbb{1}_{\{f(x_i) \neq y_i\}}$$

- argmin_f objective (f) outputs the function that minimizes the objective.
- The objective function is called the cost function (or the loss function), and the objective is to minimize the cost.

Squared Loss Function

Motivation

- Zero-one loss counts the number of mistakes made by the classifier. The best classifier is the one that makes the fewest mistakes.
- Another example is the squared distance between the predicted and the actual y value:

$$\hat{f} = \operatorname*{argmin}_{f} \frac{1}{2} \sum_{i=1}^{n} \left(f\left(x_i\right) - y_i\right)^2$$

Hypothesis Space

Motivation

- There are too many functions to choose from.
- There should be a smaller set of functions to choose $\hat{f}$ from.

$$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \left(f\left(x_i\right) - y_i\right)^2$$

- The set $\mathcal{H}$ is called the hypothesis space.

Linear Regression

Motivation

- For example, $\mathcal{H}$ can be the set of linear functions. Then the problem can be rewritten in terms of the weights.

$$\left(\hat{w}_1, \ldots, \hat{w}_m, \hat{b}\right) = \operatorname*{argmin}_{w_1, \ldots, w_m, b} \frac{1}{2} \sum_{i=1}^{n} \left(a_i - y_i\right)^2$$

where $a_i = w_1 x_{i1} + w_2 x_{i2} + \ldots + w_m x_{im} + b$

- The problem is called (least squares) linear regression.

Binary Classification

Motivation

- If the problem is binary classification, y is either 0 or 1, and linear regression is not a great choice.
- This is because if the prediction is either too large or too small, the prediction is correct, but the cost is large.

Activation Function

Motivation

- Suppose $\mathcal{H}$ is the set of functions that are compositions between another function g and linear functions.

$$\left(\hat{w}, \hat{b}\right) = \underset{w,b}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} (a_i - y_i)^2$$

$$\text{where } a_i = g\left(w^T x + b\right)$$

- g is called the activation function.

Linear Threshold Unit

Motivation

- One simple choice is to use the step function as the activation function:

$$g\left(\boxed{\cdot}\right) = \mathbb{1}_{\left\{\boxed{\cdot}\geqslant 0\right\}} = \begin{cases} 1 & \text{if } \boxed{\cdot} \geqslant 0 \\ 0 & \text{if } \boxed{\cdot} < 0 \end{cases}$$

- This activation function is called linear threshold unit (LTU).

Sigmoid Activation Function

Motivation

- When the activation function g is the sigmoid function, the problem is called logistic regression.

$$g\left(\boxed{\cdot}\right) = \frac{1}{1 + \exp\left(-\boxed{\cdot}\right)}$$

- This g is also called the logistic function.

Cross-Entropy Loss Function

Motivation

- The cost function used for logistic regression is usually the log cost function.

$$C(f) = -\sum_{i=1}^{n} \left(y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i)) \right)$$

- It is also called the cross-entropy loss function.

Logistic Regression Objective

Motivation

- The logistic regression problem can be summarized as the following.

$$\left(\hat{w}, \hat{b}\right) = \operatorname*{argmin}_{w,b} - \sum_{i=1}^{n} \left(y_i \log\left(a_i\right) + \left(1 - y_i\right) \log\left(1 - a_i\right)\right)$$

where $a_i = \dfrac{1}{1 + \exp\left(-z_i\right)}$ and $z_i = w^T x_i + b$

Logistic Regression

Description

- Initialize random weights.
- Evaluate the activation function.
- Compute the gradient of the cost function with respect to each weight and bias.
- Update the weights and biases using gradient descent.
- Repeat until convergent.

Gradient Descent Intuition

Definition

- If a small increase in w_1 causes the distances from the points to the regression line to decrease: increase w_1.
- If a small increase in w_1 causes the distances from the points to the regression line to increase: decrease w_1.
- The change in distance due to change in w_1 is the derivative.
- The change in distance due to change in $\begin{bmatrix} w \\ b \end{bmatrix}$ is the gradient.

Gradient

Definition

- The gradient is the vector of derivatives.
- The gradient of $f(x_i) = w^T x_i + b = w_1 x_{i1} + w_2 x_{i2} + \ldots + w_m x_{im} + b$ is:

$$\nabla_w f = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \ldots \\ \frac{\partial f}{\partial w_m} \end{bmatrix} = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \ldots \\ x_{im} \end{bmatrix} = x_i$$

$$\nabla_b f = 1$$

Chain Rule

Definition

- The gradient of $f\left(x_i\right) = g\left(w^T x_i + b\right) = g\left(w_1 x_{i1} + w_2 x_{i2} + \ldots + w_m x_{im} + b\right)$ can be found using the chain rule.

$$\nabla_w f = g'\left(w^T x_i + b\right) x_i$$

$$\nabla_b f = g'\left(w^T x_i + b\right)$$

- In particular, for the logistic function g :

$$g\left(\boxdot\right) = \frac{1}{1 + \exp\left(-\boxdot\right)}$$

$$g'\left(\boxdot\right) = g\left(\boxdot\right)\left(1 - g\left(\boxdot\right)\right)$$

Gradient Descent Step

Definition

- For logistic regression, use chain rule twice.

$$w = w - \alpha \sum_{i=1}^{n} (a_i - y_i)\, x_i$$

$$b = b - \alpha \sum_{i=1}^{n} (a_i - y_i)$$

$$a_i = g\left(w^T x_i + b\right), g\left(\boxed{\cdot}\right) = \frac{1}{1 + \exp\left(-\boxed{\cdot}\right)}$$

- α is the learning rate. It is the step size for each step of gradient descent.

Perceptron Algorithm

Definition

- Update weights using the following rule.

$$w = w - \alpha \left(a_i - y_i \right) x_i$$

$$b = b - \alpha \left(a_i - y_i \right)$$

$$a_i = \mathbb{1}_{\{w^T x_i + b \geqslant 0\}}$$

Logistic Regression, Part 1

Algorithm

- Inputs: instances: $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$
- Outputs: weights and biases: $w_1, w_2, ..., w_m$ and b
- Initialize the weights.

$$w_1, ..., w_m, b \sim \text{ Unif } [-1, 1]$$

- Evaluate the activation function.

$$a_i = g\left(w^T x_i + b\right), g\left(\boxed{\cdot}\right) = \frac{1}{1 + \exp\left(-\boxed{\cdot}\right)}$$

Logistic Regression, Part 2

Algorithm

- Update the weights and bias using gradient descent.

$$w = w - \alpha \sum_{i=1}^{n} (a_i - y_i)\, x_i$$

$$b = b - \alpha \sum_{i=1}^{n} (a_i - y_i)$$

- Repeat the process until convergent.

$$|C - C^{\text{prev}}| < \varepsilon$$

Stopping Rule and Local Minimum

Discussion

- Start with multiple random weights.
- Use smaller or decreasing learning rates. One popular choice is $\frac{\alpha}{\sqrt{t}}$, where t is the iteration count.
- Use the solution with the lowest C.

Regression vs Classification

Discussion

- Logistic regression is usually used to solve classification problems (y is discrete or categorical), not regression problems (y is continuous).
- This course (and machine learning in general) will focus on solving classification problems.

Other Non-linear Activation Function

Discussion

- Activation function: $g\left(\boxed{\cdot}\right) = \tanh\left(\boxed{\cdot}\right) = \dfrac{e^{\boxed{\cdot}} - e^{-\boxed{\cdot}}}{e^{\boxed{\cdot}} + e^{-\boxed{\cdot}}}$
- Activation function: $g\left(\boxed{\cdot}\right) = \arctan\left(\boxed{\cdot}\right)$
- Activation function (rectified linear unit): $g\left(\boxed{\cdot}\right) = \boxed{\cdot}\mathbb{1}_{\left\{\boxed{\cdot} \geqslant 0\right\}}$
- All these functions lead to objective functions that are convex and differentiable (almost everywhere). Gradient descent can be used.

Convexity

Discussion

- If a function is convex, gradient descent with any initialization will converge to the global minimum (given sufficiently small learning rate).
- If a function is not convex, gradient descent with different initializations may converge to different local minima.
- A twice differentiable function is convex if and only its second derivative is non-negative.
- In the multivariate case, it means the Hessian matrix is positive semidefinite.

Positive Semidefinite

Discussion

- Hessian matrix is the matrix of second derivatives:

$$H : H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

- A matrix H is positive semidefinite if $x^T Hx \geqslant 0 \forall\ x \in \mathbb{R}^n$.
- A symmetric matrix is positive semidefinite if and only if all of its eigenvalues are non-negative.