Solution to Exercise for Chapter 5 —— Machine Learning Basics

Xiao Zhang & Fnu Suya

1. To better understand the machine learning concepts in Chapter 5, let's consider another simple machine learning algorithm (beyond linear regression): **logistic regression**, which is widely used for classification. In particular, we focus on binary classification problem in which the output $y \in \{0, 1\}$. As introduced in Section 5.7.1, the relationship between the input feature vector $\boldsymbol{x} \in \mathbb{R}^p$ and the output class y is modeled as

$$\mathbb{P}(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^\top \boldsymbol{x}), \tag{1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^p$ is the model parameter we are going to estimate, and $\sigma(z) = \frac{1}{1+\exp(-z)}$ is the logistic sigmoid function.

   (a) Given a set of m training examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^m$, construct the maximum likelihood estimator $\boldsymbol{\theta}_{\text{ML}}$ for logistic regression. (you may want to review Section 5.5.1)

   Solution: According to the model specified by logistic regression, for each example $\{\boldsymbol{x}, y\}$, the conditional probability can be written in the following compact form

$$\mathbb{P}(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \big[\sigma(\boldsymbol{\theta}^\top \boldsymbol{x})\big]^y \cdot \big[1 - \sigma(\boldsymbol{\theta}^\top \boldsymbol{x})\big]^{(1-y)}.$$

   Suppose that the training examples $\{\boldsymbol{x}_i, y_i\}$ are independent to each other, then the (conditional) likelihood function is

$$L(\boldsymbol{\theta}) = \prod_{i=1}^m \mathbb{P}(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^m \Big\{ \big[\sigma(\boldsymbol{\theta}^\top \boldsymbol{x})\big]^y \cdot \big[1 - \sigma(\boldsymbol{\theta}^\top \boldsymbol{x})\big]^{(1-y)} \Big\}.$$

   To avoid underflow, the maximum likelihood estimator $\boldsymbol{\theta}_{\text{ML}}$ is usually computed via minimizing the negative log likelihood

$$\boldsymbol{\theta}_{\text{ML}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^m \Big\{ -y_i \log\big(\sigma(\boldsymbol{\theta}^\top \boldsymbol{x})\big) - (1 - y_i) \log\big(1 - \sigma(\boldsymbol{\theta}^\top \boldsymbol{x})\big) \Big\}.$$

   (b) Another estimator for logistic regression can be obtained by minimizing the mean square error (as in the case of linear regression), namely

$$\boldsymbol{\theta}_{\text{MSE}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{m} \sum_{i=1}^m \big(\sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i) - y_i\big)^2.$$

But in practice, no one use $\boldsymbol{\theta}_{\text{MSE}}$ for logistic regression. Can you explain the reasons why $\boldsymbol{\theta}_{\text{ML}}$ is preferred over $\boldsymbol{\theta}_{\text{MSE}}$?

Solution: There are two main reasons: (1) Maximum likelihood estimator is the best estimator asymptotically, which has nice properties such as consistency, asymptotic normality and etc. (2) In terms of optimization, the objective function (negative log loss) for MLE is convex, while the objective (square loss) for MSE is non-convex. In general, minimizing a convex objective function is much easier than solving a non-convex optimization problem.

(c) Different from linear regression, maximum likelihood estimator for logistic regression doesn't have explicit analytical form. Thus, people turn to iterative algorithm such as gradient descent to approximate $\boldsymbol{\theta}_{\text{ML}}$. Consider the negative log likelihood as the cost function, write down the gradient descent algorithm. (you may want to review Section 4.5)

Solution: Let $J(\boldsymbol{\theta})$ be the negative log likelihood computed in (a). The gradient with respect to $\boldsymbol{\theta}$ can be computed as

$$\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \left\{ -y_i \cdot \frac{\sigma'(\boldsymbol{\theta}^\top \boldsymbol{x}_i)}{\sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i)} \boldsymbol{x}_i - (1-y_i) \cdot \frac{-\sigma'(\boldsymbol{\theta}^\top \boldsymbol{x}_i)}{1-\sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i)} \boldsymbol{x}_i \right\} \\
&= \sum_{i=1}^{m} \left\{ -y_i \big(1-\sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i)\big) \boldsymbol{x}_i + (1-y_i)\sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i) \boldsymbol{x}_i \right\} \\
&= \sum_{i=1}^{m} \left\{ \big[\sigma(\boldsymbol{\theta}^\top \boldsymbol{x}_i) - y_i\big] \boldsymbol{x}_i \right\},
\end{aligned}$$

where the second equation holds because the derivative of sigmoid function $\sigma'(z) = \sigma(z) \cdot \big(1-\sigma(z)\big)$. Based on the computed gradient, we can lay out the gradient descent algorithm, as demonstrated in Algorithm 1.

Algorithm 1 Gradient Descent for Logistic Regression

Input: Cost function $J(\boldsymbol{\theta})$; step size ϵ; tolerance parameter $\delta > 0$

while $\|\nabla J(\boldsymbol{\theta})\|_2 > \delta$ **do**

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \nabla J(\boldsymbol{\theta})$

end while

Output: $\boldsymbol{\theta}$

(d) In practice, people add an additional ℓ_2 regularizer $\lambda\|\boldsymbol{\theta}\|_2^2 = \lambda\sum_{i=1}^{p}\theta_i^2$ to the cost function to achieve better performance, where $\lambda > 0$ is the regularization parameter. Can you briefly explain why this regularization term can help reduce overfitting? (you may refer to Section 5.2.2)

Solution: Generally speaking, ℓ_2 regularization term controls the model capacity by restricting the parameter space of $\boldsymbol{\theta}$ in terms of ℓ_2 norm. According to Section 5.2.2, a model with appropriate capacity can reduce overfitting and generalize better. (see the link l2 regularization and overfitting for more details).

(e) Besides ℓ_2 regularizer, another widely-used regularization technique for logistic regression is ℓ_1 regularization, where $\|\boldsymbol{\theta}\|_2^2$ is replaced by $\|\boldsymbol{\theta}\|_1 = \sum_{i=1}^{p}|\theta_i|$ in the objective function. Explain the difference between ℓ_1 and ℓ_2 regularization.

Solution: As for the property of optimal solution, ℓ_1 regularization can induce sparsity, which can be further used for feature selection; while ℓ_2 regularization cannot. However, optimizing an objective with ℓ_2 regularization is much easier than ℓ_1 regularized problem, because ℓ_1 norm is not differentialble everywhere. Thus, we need to use subgradient as a substitute when solving a ℓ_1 regularized objective. (see Section 3.4 of The Element of Statistical Learning for more details regarding ridge regression and Lasso)

(f) The machine learning course taught by Andrew Ng on Coursera is provided with programming exercises, which are available here: `https://github.com/rieder91/MachineLearning`. Exercise 2 would be a good practice for (regularized) logistic regression. The solution written in python can be found in this url: `https://github.com/nex3z/machine-learning-exercise`.