Tsinghua-Berkeley Shenzhen Institute
Learning From Data
Fall 2022

Written Assignment 1

Issued: Saturday 24th September, 2022 **Due:** Saturday 8th October, 2022

COMMENTS

- Mention *collaborators* in your assignments. See the policies for details.
- Provide sufficient arguments in your proof.

POLICIES

- **Acknowledgments:** We expect you to make an honest effort to solve the problems individually. As we sometimes reuse problem set questions from previous years, covered by papers and web pages, we expect the students **NOT** to copy, refer to, or look at the solutions in preparing their answers (relating to an unauthorized material is considered a violation of the honor principle). Similarly, we expect to not to google directly for answers (though you are free to google for knowledge about the topic). If you do happen to use other material, it must be acknowledged here, with a citation on the submitted solution.
- **Required homework submission format:** You can submit homework either as one single PDF document or as handwritten papers. Written homework needs to be provided during the class in the due date, and PDF document needs to be submitted through Tsinghua's Web Learning (`http://learn.tsinghua.edu.cn/`) before the end of due date.
- **Collaborators:** In a separate section (before your answers), list the names of all people you collaborated with and for which question(s). If you did the HW entirely on your own, **PLEASE STATE THIS**. Each student must understand, write, and hand in answers of their own.

1.1. (Sigmoid Function) Show that the sigmoid function

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

satisfies the following properties.

(a) (0.5 points) $\sigma(-x) = 1 - \sigma(x)$.

(b) (0.5 points) $\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$.

1.2. (Ridge Regression) Ridge regression was developed as a possible solution to the imprecision of least square estimators when linear regression models have some multicollinear (highly correlated) independent variables.

We can formulate the ridge regression loss function as the following

$$J(\boldsymbol{\theta}) \overset{\text{def}}{=} ||\boldsymbol{y} - X\boldsymbol{\theta}||^2 + \lambda||\boldsymbol{\theta}||^2,$$

where X is the design matrix, $\boldsymbol{y}$ is the corresponding label vector, and $\boldsymbol{\theta}$ is the weight vector. For an appropriate λ,

(a) (1 point) calculate $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$,

(b) (1 point) give the gradient descend iteration equation with learning rate α,

(c) (1 point) derive the optimal parameter $\boldsymbol{\theta}^*$ for the normal equation method.

1.3. (MAP) Suppose we have m samples $x_1, x_2, ..., x_m$ independently drawn from a normal distribution with known variance σ^2 and unknown mean θ, i.e.

$$P(x_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_i-\theta)^2}{2\sigma^2}).$$

Maximum likelihood estimation (MLE) assumes that the optimal parameter θ is the one that generates the observed data with the highest probability, i.e. $\theta_{MLE} \overset{\text{def}}{=} \arg\max_\theta P(x_1, x_2, ..., x_m|\theta)$. However, what if we know some additional prior information about the distribution of θ? e.g. Let θ be a random variable following a Gaussian distribution, i.e. $\theta \sim \mathcal{N}(\nu, \mu^2)$. We can calculate the posterior distribution of θ using Bayes' theorem and derive the MAP estimator θ_{MAP}, i.e.

$$\theta_{MAP} \overset{\text{def}}{=} argmax_\theta P(\theta|x_1, \ldots, x_m) = \frac{P(x_1, \ldots, x_m|\theta)P(\theta)}{P(x_1, \ldots, x_m)}.$$

(a) (1 point) Find the MLE estimator for θ;

(b) (1 point) Find the MAP estimator for θ;

(c) (1 point) Compare the estimators of MLE and MAP when n is very large.

1.4. (Softmax Regression)(3 points) In multivariate classification problems, we use softmax function to derive the likelihood of each possible label y and predict the most probable one for data $\boldsymbol{x} \in \mathbb{R}^n$. To train parameter matrix $\boldsymbol{\Theta} \in \mathbb{R}^{n\times k}$ from the given samples $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right), i = 1, \ldots, m$, we need to calculate the derivative of the softmax model's log-likelihood function

$$\ell(\boldsymbol{\Theta}) \overset{\text{def}}{=} \sum_{i=1}^{m} \log p(y^{(i)}|\boldsymbol{x}^{(i)}; \boldsymbol{\Theta}) = \sum_{i=1}^{m}\sum_{l=1}^{k} \mathbf{1}\left\{y^{(i)} = l\right\} \log \frac{e^{\boldsymbol{\theta}_l^{\mathrm{T}}\boldsymbol{x}^{(i)}}}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^{\mathrm{T}}\boldsymbol{x}^{(i)}}}.$$

Calculate $\nabla_{\boldsymbol{\theta}_1} \ell(\boldsymbol{\Theta})$.

1.5. (Implicit Bias of Gradient Descent)(Bonus 3 points) Consider the problem of solving an under-determined system of linear equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $m < n$. Of course, the solution is not unique. Nevertheless, let us solve it by minimizing the least square error

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \overset{\text{def}}{=} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2,$$

say using the simplest gradient descent algorithm:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k).$$

If we initialize x_0 as the origin $\mathbf{0}$, then when the above gradient descent algorithm converges,

(a) (1.5 points) calculate $\boldsymbol{x}_\infty$,

(b) (1.5 points) prove that $\boldsymbol{x}_\infty = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{y}$. Hint: Use the SVD of $\boldsymbol{A}$.

This is a phenomenon widely exploited in the practice of learning deep neural networks. Although due to over-parameterization, parameters that minimize the cost function might not be unique, the choice of optimization algorithms with proper initialization (here gradient descent starting from the origin) introduces implicit bias for the optimization path and converges to a desirable solution.