Introduction to Machine Learning, Spring 2025
Homework 3

(Due April 6, 2025 at 11:59pm (CST))

March 18, 2025

1. Please write your solutions in English.
2. Submit your solutions to the course Gradescope.
3. If you want to submit a handwritten version, scan it clearly.
4. Late homeworks submitted within 3 days of the due date will be marked down 25% each day cumulatively Homeworks submitted more than 3 days after the due date will not be accepted unless there is a valid reason, such as a medical or family emergency.
5. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

1. [15 points] [Perceptron Learning Algorithm] Consider a binary classification problem. The input space is $\mathbb{R}^d$. The output space is $\{+1, -1\}$. For simplicity, we modified the input to be $\mathbf{x} = [x_0, x_1, \cdots, x_d]^\top$ with $x_0 = 1$. The output is predicted using the hypothesis:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x}), \tag{1}$$

where $\mathbf{w} = [w_0, w_1, \cdots, w_d]^\top$ and w_0 is the bias.

The *perceptron learning algorithm* determines $\mathbf{w}$ using a simple iterative method. Here is how it works. At iteration t, where $t = 0, 1, 2, \ldots$, there is a current value of the weight vector, call it $\mathbf{w}(t)$. The algorithm picks an example from $(\mathbf{x}_1, y_1) \cdots (\mathbf{x}_N, y_N)$ that is currently misclassified, call it $(\mathbf{x}(t), y(t))$, and uses it to update $\mathbf{w}(t)$. Since the example is misclassified, we have $y(t) \neq \text{sign}\left(\mathbf{w}^\top(t)\mathbf{x}(t)\right)$. The update rule is

$$\mathbf{w}(t+1) = \mathbf{w}(t) + y(t)\mathbf{x}(t). \tag{2}$$

(a) Show that $y(t)\mathbf{w}^\top(t)\mathbf{x}(t) < 0$. [Hint: $\mathbf{x}(t)$ is misclassified by $\mathbf{w}(t)$.] [5 points]

(b) Show that $y(t)\mathbf{w}^\top(t+1)\mathbf{x}(t) > y(t)\mathbf{w}^\top(t)\mathbf{x}(t)$. [5 points]

(c) As far as classifying $\mathbf{x}(t)$ is concerned, argue that the move from $\mathbf{w}(t)$ to $\mathbf{w}(t+1)$ is a move "in the right direction". [5 points]

Solution

2. [20 points] [Maximum Margin Classifier] Consider a data set of n d-dimensional sample points, $\{x_1, \ldots, x_n\}$. Each sample point, $x_i \in \mathbb{R}^d$, has a corresponding label, y_i, indicating to which class that point belongs. For now, we will assume that there are only two classes and that every point is either in the given class ($y_i = 1$) or not in the class ($y_i = -1$). Consider the linear decision boundary defined by the hyperplane

$$\mathcal{H} = \left\{ x \in \mathbb{R}^d : x \cdot w + \alpha = 0 \right\}.$$

The maximum margin classifier maximizes the distance from the linear decision boundary to the closest training point on either side of the boundary, while correctly classifying all training points. Suppose the points are linear seperable, and the margin is γ.

(a) The maximum margin classifier aims to maximize the distance from the training points to the decision boundary. Derive the distance from a point x_i to the hyperplane $\mathcal{H}$. [5 points]

(b) An in-class sample point is correctly classified if it is on the positive side of the decision boundary, and an out-of-class sample is correctly classified if it is on the negative side. Assuming all the points are correctly classified, write a set of n constraints to ensure that all n points are correctly classified. [5 points]

(c) Using the previous parts, write an optimization problem for the maximum margin classifier. For convinent, we should additionally add a constrain $\|w\| = 1$ [5 points]

(d) To simply the optimization problem, we can rewrite the optimization problem in part (c) by setting $w' = \dfrac{w}{\gamma}$ and $\alpha' = \dfrac{\alpha}{\gamma}$. Write the optimization problem for the simlified maximum margin classifier. [5 points]

Solution

3. [10 points] [Leave-one-out Cross-validation]

Select each training example in turn as the single example to be held-out, train the classifier on the basis of all the remaining training examples, test the resulting classifier on the held-out example, and count the errors.

Let the superscript $-i$ denote the parameters we would obtain by finding the SVM classifier f without the ith training example. Define the *leave-one-out CV error* as

$$\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(y_i, f(\mathbf{x}_i; \mathbf{w}^{-i}, b^{-i})), \tag{3}$$

where $\mathcal{L}$ is the zero-one loss. Prove that

$$\text{leave-one-out CV error} \leq \frac{\text{number of support vectors}}{\text{n}} \tag{4}$$

Solution

4. [10 points] [Probability and Estimation]

The Poisson distribution is a useful discrete distribution which can be used to model the number of occurrences of something per unit time. For example, in networking, the number of packets to arrive in a given time window is often assumed to follow a Poisson distribution. $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}, n > 1$ are i.i.d. samples from exponential distribution with parameter $\lambda > 0$, i.e., $X \sim \text{Expo}(\lambda)$. Recall the PDF of exponential distribution is

$$p(x \mid \lambda) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

(a) To derive the posterior distribution of λ, we assume its prior distribution follows gamma distribution with parameters $\alpha, \beta > 0$, i.e., $\lambda \sim \text{Gamma}(\alpha, \beta)$ (since the range of gamma distribution is also $(0, +\infty)$, thus it's a plausible assumption). The PDF of λ is given by

$$p(\lambda \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\lambda\beta}$$
$$\text{where} \qquad \Gamma(\alpha) = \int_0^{+\infty} t^{\alpha-1} e^{-t} dt, \ \alpha > 0.$$

Show that the posterior distribution $p(\lambda \mid \mathcal{D})$ is also a gamma distribution and identify its parameters. Hints: Feel free to drop constants. [5 points]

(b) Derive the maximum a posterior (MAP) estimation for λ under $\text{Gamma}(\alpha, \beta)$ prior. [5 points]

Solution

5. [10 points] [Linear Classification] Consider the "Multi-class Logistic Regression" algorithm. Given training set $\mathcal{D} = \{(x^i, y^i) \mid i = 1, \dots, n\}$ where $x^i \in \mathbb{R}^{p+1}$ is the feature vector and $y^i \in \mathbb{R}^k$ is a one-hot binary vector indicating k classes. We want to find the parameter $\hat{\beta} = [\hat{\beta}_1, \dots, \hat{\beta}_k] \in \mathbb{R}^{(p+1)\times k}$ that maximize the likelihood for the training set. Introducing the softmax function, we assume our model has the form

$$p(y_c^i = 1 \mid x^i; \beta) = \frac{\exp(\beta_c^\top x^i)}{\sum_{c'} \exp(\beta_{c'}^\top x^i)}$$

where y_c^i is the c-th element of y^i.

(a) Complete the derivation of the conditional log likelihood for our model, which is

$$\ell(\beta) = \ln \prod_{i=1}^{n} p(y_t^i \mid x^i; \beta) = \sum_{i=1}^{n} \sum_{c=1}^{k} \left[y_c^i(\beta_c^\top x^i) - y_c^i \ln \left(\sum_{c'} \exp(\beta_{c'}^\top x^i) \right) \right].$$

For simplicity, we abbreviate $p(y_t^i = 1 \mid x^i; \beta)$ as $p(y_t^i \mid x^i; \beta)$, where t is the true class for x^i. [5 points]

(b) Derive the gradient of $\ell(\beta)$ w.r.t. β_1, i.e.,

$$\nabla_{\beta_1} \ell(\beta) = \nabla_{\beta_1} \sum_{i=1}^{n} \sum_{c=1}^{k} \left[y_c^i(\beta_c^\top x^i) - y_c^i \ln \left(\sum_{c'} \exp(\beta_{c'}^\top x^i) \right) \right]$$

Remark: Log likelihood is always concave; thus, we can optimize our model using gradient ascent. (The gradient of $\ell(\beta)$ w.r.t. $\beta_2, \dots, \beta_k$ is similar, you don't need to write them) [5 points]

Solution