Homework 2

Miaopeng Yu my2533

October 1, 2021

1 Question T1

1.1 Which i and t should we pick as the best model and why?

We should pick the model M_j^i with the smallest expected cost on the validation set $\mathcal{L}_{val,t}^i$.

Since we train the model on the training set, the model has already recorded the feature of training set, so it is more likely to have a lower cost. But we cannot guarantee that if this model can generalize to the unseen data.

As for the test sets, we cannot use the test sets until we have already selected the model, or it will pollute the data. So we cannot use it to select model.

1.2 How should we report the generalization error of the model?

We can use the expected cost of the test set $\mathcal{L}_{test,t}^i$ as the generalization error of the model.

Because the test set is chosen randomly from the data, and it is also unseen to the model before we calculate the cost of it. So it can better generalize the whole data. Which can be used to estimate the generalization error of the model.

2 Question T2

$w = [w_1, w_2, \cdots, w_K]$, where w_j is a K dimensional column vector.

w_{ij} refers to the value located at the i-th row and j-th column.

$$p_j = \sigma(w^T \cdot x)_j = \frac{e^{w_j^T \cdot x}}{\sum_{k=1}^K e^{w_k^T \cdot x}}$$

$$\frac{\partial p_j}{\partial w_j} = \frac{w_j \cdot e^{w_j^T \cdot x} \left(\sum_{k=1}^{K} e^{w_k^T \cdot x}\right) - w_j \cdot e^{w_j^T \cdot x} \cdot \left(e^{w_j^T \cdot x}\right)}{\left(\sum_{k=1}^{K} e^{w_k^T \cdot x}\right)^2}$$

$$= w_j \cdot \frac{e^{w_j^T \cdot x} \left(\sum_{k \neq j}^{K} e^{w_i^T \cdot x}\right)}{\left(\sum_{k=1}^{K} e^{w_i^T \cdot x}\right)^2}$$

$$\therefore \frac{\partial p_j}{w_{ij}} = w_{ij} \cdot \frac{e^{w_j^T \cdot x} \left(\sum_{k \neq j}^{K} e^{w_i^T \cdot x}\right)}{\left(\sum_{k=1}^{K} e^{w_i^T \cdot x}\right)^2}$$

$$\therefore \frac{\partial \mathcal{L}_w(x,y)}{\partial w_{ij}} = \frac{\partial \mathcal{L}_w(x,y)}{\partial p_j} \cdot \frac{\partial p_j}{\partial w_{ij}} = -\frac{y_j \cdot w_{ij}}{p_j} \cdot \frac{e^{w_j^T \cdot x} \left(\sum_{k \neq j}^{K} e^{w_i^T \cdot x}\right)}{\left(\sum_{k=1}^{K} e^{w_i^T \cdot x}\right)^2}$$

3 Question T3

Suppose $x_1, x_2, \cdots, x_n$ are independent, and $x_i \sim N(\mu, \sigma^2)$.

3.1 Write down the expression of the log-likelihood $\mathcal{L}_{\mu,\sigma}(\mathcal{D})$ of the data set $\mathcal{D}$ as a function of μ and σ.

$P(x_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$

$\therefore P(x_1, x_2, \cdots, x_n) = \Pi_{i=1}^{n} P(x_i) = \dfrac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}}$

$\therefore \mathcal{L}_{\mu,\sigma}(\mathcal{D}) = \log(P(x_1, x_2, \cdots, x_n)) = -\frac{n}{2}\log(2\pi\sigma^2) - \dfrac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}$

3.2 Compute the partial derivative of $\mathcal{L}_{\mu,\sigma}(\mathcal{D})$ with respect to μ, equate to zero and solve for μ.

$\frac{\partial \mathcal{L}_{\mu,\sigma}(\mathcal{D})}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) = 0$

$\therefore \mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}$

3.3 Compute the partial derivative of $\mathcal{L}_{\mu,\sigma}(\mathcal{D})$ with respect to σ, equate to zero and solve for σ.

$\frac{\partial \mathcal{L}_{\mu,\sigma}(\mathcal{D})}{\partial \sigma} = -\dfrac{n}{\sigma} + \dfrac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^3} = 0$

$\because \sigma > 0$

$\therefore \sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

4 Question T4

We can use a gradient-based optimization algorithm called Proximal Gradient Method by using subgradient.[1]

A vector g is a subgradient of f at x iff $(g, 1)$ defines non-vertical supporting hyperplane to epi(f) at $(x, f(x))$, i.e.

$$\begin{pmatrix} g \\ -1 \end{pmatrix}^T \left(\begin{pmatrix} y \\ t \end{pmatrix} - \begin{pmatrix} x \\ f(x) \end{pmatrix} \right) \leq 0, \forall (y,t) \in \text{epi}(f)$$

And the subdifferential $\partial f(x)$ of f at x is the set of all subgradients.

And we can simply replace the update step of gradient descent method $\theta_{n+1} = \theta_n - \text{step\_size} * \nabla f$ with $\theta_{n+1} = \theta_n - \text{step\_size} * g$

For any x that satisfies $1 - y \cdot f_\theta(x) > 0$, the gradient descent method can work well, and $\partial \mathcal{L}_{Hinge}(x, y, \theta) = \{\nabla \mathcal{L}_{Hinge}(x, y, \theta)\}$. So $g = \nabla \mathcal{L}_{Hinge}(x, y, \theta)$.

For x that satisfies $1 - y \cdot f_\theta(x) < 0$, the gradient descent method can work well too, and $\partial \mathcal{L}_{Hinge}(x, y, \theta) = \{0\}$. So $g = 0$.

For x that satisfies $1 - y \cdot f_\theta(x) = 0$, then $\mathcal{L}_{Hinge}(x, y, \theta)$ is not differentiable at this point. But since $0 \in \partial \mathcal{L}_{Hinge}(x, y, \theta)$, and a function reaches its optimal point when $0 \in \partial \mathcal{L}_{Hinge}(x, y, \theta)$, so we can simply use 0 to update the parameter. So $g = 0$.

By setting the subgradient of this function $\mathcal{L}_{Hinge}(x, y, \theta)$, we can again use the gradient-based optimization algorithm.

References

[1] The two materials I refer to are the lecture notes of Method of Optimization, and the lecture notes are attached in the same file.