Introduction to Machine Learning (CSCI-UA.473): Homework 2

Instructor: Sumit Chopra

October 14, 2021

Theory

Question T1: Model Selection (5 points)

Consider that we are learning a logistic regression M^1 and a support vector machine M^2, and we have partitioned the data into three subsets: D_{train} (training set), D_{val} (validation set), and D_{test} test set. The two models are iteratively optimized on D_{train} over T steps, and now we have T logistic regression parameter configurations (i.e., weights and biases) $M^1_1, M^1_2, \ldots, M^1_T$ and T support vector configurations $M^2_1, M^2_2, \ldots, M^2_T$ all with different parameters. We now evaluate the expected cost for all the $2T$ models on training set, validation set, and test set. Thus we have $6T$ quantities $\mathcal{L}^i_{\text{train},t}, \mathcal{L}^i_{val,t}$, and $\mathcal{L}^i_{\text{test},t}$ where $i \in \{1, 2\}$ and $t \in \{1, 2, \ldots, T\}$

1. Which i and t should we pick as the best model and why? (2.5 points)

   Answer: To access fitness of the models, we can compare $\mathcal{L}^1_{\text{train},t}$ with $\mathcal{L}^2_{\text{train},t}$ and see how the logistic regression model M^1 and the SVM model M^2 converges. At each step, the model with lower in-sample error better fits the data set.

   However, since the Goal of Machine Learning is to lower the generalization error as much as possible given D, we should pick model $M^i_{t^*}$ with the lowest $\mathcal{L}^i_{val,t^*}$ on the validation set. This is because the validation set is the best way to evaluate the generalization error without polluating the model with the test set.

2. How should we report the generalization error of the model? (2.5 points)

   Answer: Generalization error for each model M^i_t can be reported as $\mathcal{L}^i_{\text{test},t}$. The test set provides the best evaluation of out-of-sample error because since it contains unseen data points randomly sampled from the same population. To report the overall generalization error, we can take the average of the test set error for each model.

Question T2: Gradient of Multi-Class Logistic Regression (10 points)

The loss function on a single sample (x, y) for a logistic regression model with parameters w for the multi-class classification problem can be written as

$$\mathcal{L}_w(x, y) = -\sum_{j=1}^{K} y_j \cdot \log p_j,$$

where K is the number of classes, y_j is the ground truth label corresponding to the j-th class for the current sample, and p_j is defined as:

$$\begin{aligned} p_j &= \sigma\left(w^T \cdot x\right)_j \\ &= \frac{e^{w_j^T \cdot x}}{\sum_{j=1}^{K} e^{w_j^T \cdot x}} \end{aligned}$$

The function $\sigma()$ is also called the Softmax and the loss function $\mathcal{L}_w$ is called the cross-entropy loss: by far the most popular loss function used to solve multiclass classification tasks.

Compute the gradient of the above loss function with respect to the parameter vector w. Show all the steps of the derivation.

Answer: First compute the gradient of the Softmax function p_j with respect to the parameter w_i and w_j where $i \neq j$:

$$\begin{aligned} \frac{\partial p_j}{\partial w_j} &= \frac{\partial}{\partial w_j} \frac{e^{w_j^T \cdot x}}{\sum_{k=1}^{K} e^{w_k^T \cdot x}} \\ &= \frac{\frac{\partial e^{w_j^T \cdot x}}{\partial w_j} \cdot \sum_{k=1}^{K} e^{w_k^T \cdot x} - e^{w_j^T \cdot x} \cdot \sum_{k=1}^{K} \frac{\partial e^{w_k^T \cdot x}}{\partial w_j}}{[\sum_{k=1}^{K} e^{w_k^T \cdot x}]^2} \\ &= x \frac{e^{w_j^T \cdot x}}{\sum_{k=1}^{K} e^{w_k^T \cdot x}} - \frac{e^{w_j^T \cdot x} \cdot [\sum_{k \neq j}^{K} 0 + x e^{w_j^T \cdot x}]}{[\sum_{k=1}^{K} e^{w_k^T \cdot x}]^2} \\ &= x \frac{e^{w_j^T \cdot x}}{\sum_{k=1}^{K} e^{w_k^T \cdot x}} - x \frac{e^{w_j^T \cdot x} \cdot e^{w_j^T \cdot x}}{[\sum_{k=1}^{K} e^{w_k^T \cdot x}]^2} \\ &= x p_j - x p_j^2 \end{aligned}$$

$$
\begin{aligned}
\frac{\partial p_j}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{e^{w_j^T \cdot x}}{\sum_{j=1}^K e^{w_j^T \cdot x}} \\
&= e^{w_j^T \cdot x} \frac{\partial \left(\sum_{j=1}^K e^{w_j^T \cdot x} \right)^{-1}}{\partial w_i} \\
&= -e^{w_j^T \cdot x} \left(\sum_{j=1}^K e^{w_j^T \cdot x} \right)^{-2} \frac{\partial e^{w_i^T \cdot x}}{\partial w_i} \\
&= -x \cdot e^{w_j^T \cdot x} \cdot e^{w_i^T \cdot x} \left(\sum_{j=1}^K e^{w_j^T \cdot x} \right)^{-2} \\
&= -x p_j p_i
\end{aligned}
$$

The gradient of the loss function with respect to w is given by:

$$
\begin{aligned}
\boldsymbol{\nabla}_w \mathcal{L}_w(x, y) &= -\frac{\partial}{\partial w_j} \sum_{i=1}^K [y_i \cdot \log p_i] \\
&= -\sum_{i=1, i \neq j}^K y_i \frac{\partial \log p_i}{\partial w_j} - y_j \frac{\partial \log p_j}{\partial w_j} \\
&= -\sum_{i=1, i \neq j}^K y_i p_i^{-1} \frac{\partial p_i}{\partial w_j} - y_j p_j^{-1} \frac{\partial p_j}{\partial w_j} \\
&= \sum_{i=1, i \neq j}^K y_i p_i^{-1} x p_j p_i - \left[y_j p_j^{-1} (x p_j - x p_j^2) \right] \\
&= \sum_{i=1, i \neq j}^K x y_i p_j + x y_j p_j - x y_j \\
&= x p_j \sum_{i=1}^K y_i - x y_j \\
&= \boxed{x p_j - x y_j}
\end{aligned}
$$

Question T3: Maximum Likelihood Estimate of a Gaussian Model (10 Points)

Assume you are given a dataset $\mathcal{D}$ of n real numbers $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$, where $x_i \in \text{Re}, \forall i$. Derive the maximum likelihood estimate of the mean μ and variance σ, of the 1-dimensional Gaussian distribution. Note that μ and σ are the learnable parameters.

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. (2 points)

   Answer:

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

$$\begin{aligned}
\mathcal{L}_{\mu,\sigma}(\mathcal{D}) &= \log\left(\prod_{i=1}^{n} P(x_i)\right) \\
&= \log\left(\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) \\
&= \log\left(\frac{1}{\sigma^n (2\pi)^{n/2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}\right) \\
&= \boxed{-\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}
\end{aligned}$$

2. Compute the partial derivative of $\mathcal{L}_{\mu,\sigma}(\mathcal{D})$ with respect to μ, equate to zero and solve for μ. (4 points)

   Answer:

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

Let $\frac{\partial \mathcal{L}_{\mu,\sigma}(\mathcal{D})}{\partial \mu} = 0$ and solve for μ:

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) = 0$$
$$\frac{1}{\sigma^2}\sum_{i=1}^{n} x_i - \frac{n}{\sigma^2}\mu = 0$$
$$\mu = \boxed{\frac{1}{n}\sum_{i=1}^{n} x_i}$$

3. Compute the partial derivative of $\mathcal{L}_{\mu,\sigma}(\mathcal{D})$ with respect to σ, equate to zero and solve for σ. (4 points)

Answer:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mu,\sigma}(\mathcal{D})}{\partial \sigma} &= \frac{\partial}{\partial \sigma}\left(-\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right) \\ &= -\frac{n}{2}\frac{\partial}{\partial \sigma}\log\left(2\pi\sigma^2\right) - \frac{\partial}{\partial \sigma}\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2 \\ &= -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i - \mu)^2\end{aligned}$$

Let $\frac{\partial \mathcal{L}_{\mu,\sigma}(\mathcal{D})}{\partial \sigma} = 0$ and solve for σ:

$$-\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i - \mu)^2 = 0$$
$$\frac{-n\sigma^2 + \sum_{i=1}^{n}(x_i - \mu)^2}{\sigma^3} = 0$$
$$-n\sigma^2 + \sum_{i=1}^{n}(x_i - \mu)^2 = 0$$
$$n\sigma^2 = \sum_{i=1}^{n}(x_i - \mu)^2$$
$$\sigma = \boxed{\sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}}}$$

Question T4: Hinge loss gradients (5 points)

Unlike the Cross-Entropy loss, the Hinge loss (defined below), is not differentiable everywhere with respect to the parameters θ :

$$\mathcal{L}_{\text{Hinge}}\left(x, y, \theta\right) = \max\left[0, 1 - y \cdot f_\theta(x)\right],$$

for some parametric function f_θ. Does it mean that we cannot use a gradient-based optimization algorithm for finding a solution that minimizes the hinge loss? If not, what can we do about it?

Answer: $\mathcal{L}_{\text{Hinge}}\left(x, y, \theta\right)$ is differentiable everywhere with respect to θ except for $x = x^*$ such that $1 - y \cdot f_\theta(x^*) = 0$. Computing the partial derivatives and subdifferentials gives:

$$\boldsymbol{\nabla}_\theta \mathcal{L}_{\text{Hinge}}\left(x, y, \theta\right) = \begin{cases} 0 & \text{if } y \cdot f_\theta(x) > 1 \\ [0, -y \cdot \boldsymbol{\nabla}_\theta f_\theta(x)] & \text{if } y \cdot f_\theta(x) = 1 \\ -y \cdot \boldsymbol{\nabla}_\theta f_\theta(x) & \text{if } y \cdot f_\theta(x) < 1 \end{cases}$$

Provided that the Hinge loss function is intended to penalize incorrectly classified data points where y and $f_\theta(x)$ take different signs, we can consider $(x, y) \in \mathcal{D}$ where $1 - y \cdot f_\theta(x^*) = 0$ a correct classification. To ensure convergence, $\boldsymbol{\nabla}_\theta \mathcal{L}_{\text{Hinge}}\left(x, y, \theta\right)$ is zero for all correctly classified points. Therefore the gradient of the Hinge loss function is given by:

$$\boldsymbol{\nabla}_\theta \mathcal{L}_{\text{Hinge}}\left(x, y, \theta\right) = \begin{cases} 0 & \text{if } y \cdot f_\theta(x) \geq 1 \\ -y \cdot \boldsymbol{\nabla}_\theta f_\theta(x) & \text{if } y \cdot f_\theta(x) < 1 \end{cases}$$

Practicum

See the accompanying Python notebook.

Question P1: Metrics for a binary classifier (20 points)

Question P2: Support Vector Machines (50 points)