Machine Learning, 2024 Fall
Homework 1

Notice

Due 23:59 (CST), Oct 29, 2024

Plagiarizer will get 0 points.

LaTeXis highly recommended. Otherwise you should write as legibly as possible.

A Linear Regression with Multiple Variable

In class, we primarily focused on cases where our target variable y is a scalar value, briefly mentioning scenarios involving multi-dimensional predictions. However, in the real world, we are often more interested in high-dimensional cases. We follow the notation in slides:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_m^T \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y^m \end{bmatrix}$$

Thus for each training example, $y^{(i)}$ is vector-valued, with p entries. We wish to use a linear model to predict the outputs, as in least squares, by specifying the parameter matrix Θ in

$$y = \Theta^T \mathbf{x}$$

where $\Theta \in \mathbb{R}^{n \times p}$.

(1) Given that the average total error in the one-dimensional case is expressed as:

$$J(\Theta) = \frac{1}{2} \sum_{i=1}^{m} \left(\Theta^T x^{(i)} - y^{(i)} \right)^2$$

Please provide the error in multiple-dimensional without using any matrix-vector notation.

Ans:

$$J(\Theta) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{p} \left(\left(\Theta^T x^{(i)} \right)_j - y_j^{(i)} \right)^2$$

(2) Given your answer above, please provide the objective function can be expressed as:

$$J(\Theta) = \frac{1}{2} \operatorname{tr} \left((X\Theta - Y)^T (X\Theta - Y) \right)$$

where the trace of a square matrix A, denoted tr(A), is the sum of the elements on its main diagonal.
Ans:

$$\begin{aligned}
J(\Theta) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p}\left(\left(\Theta^T x^{(i)}\right)_j - y_j^{(i)}\right)^2 \\
&= \frac{1}{2}\sum_i\sum_j (X\Theta - Y)_{ij}^2 \\
&= \frac{1}{2}\sum_i (X\Theta - Y)^T(X\Theta - Y)\Big)_{ii} \\
&= \frac{1}{2}\operatorname{tr}\left((X\Theta - Y)^T(X\Theta - Y)\right)
\end{aligned}$$

(3) Find the closed form solution for Θ which minimizes $J(\Theta)$. You MAY find following equation useful:

$$\frac{\partial}{\partial \mathbf{X}}\operatorname{Tr}(\mathbf{XA}) = \mathbf{A}^T$$

$$\frac{\partial}{\partial \mathbf{X}}\operatorname{Tr}\left(\mathbf{X}^T\mathbf{BX}\right) = \mathbf{BX} + \mathbf{B}^T\mathbf{X}$$

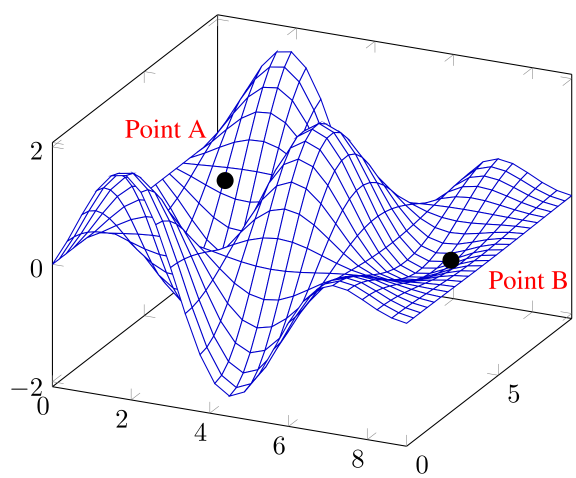
Ans:

$$\begin{aligned}
\nabla_\Theta J(\Theta) &= \nabla_\Theta\left[\frac{1}{2}\operatorname{tr}\left((X\Theta - Y)^T(X\Theta - Y)\right)\right] \\
&= \nabla_\Theta\left[\frac{1}{2}\operatorname{tr}\left(\Theta^T X^T X\Theta - \Theta^T X^T Y - Y^T X\Theta - Y^T T\right)\right] \\
&= \frac{1}{2}\nabla_\Theta\left[\operatorname{tr}\left(\Theta^T X^T X\Theta\right) - \operatorname{tr}\left(\Theta^T X^T Y\right) - \operatorname{tr}\left(Y^T X\Theta\right) + \operatorname{tr}\left(Y^T Y\right)\right] \\
&= \frac{1}{2}\nabla_\Theta\left[\operatorname{tr}\left(\Theta^T X^T X\Theta\right) - 2\operatorname{tr}\left(Y^T X\Theta\right) + \operatorname{tr}\left(Y^T Y\right)\right] \\
&= \frac{1}{2}\left[X^T X\Theta + X^T X\Theta - 2X^T Y\right] \\
&= X^T X\Theta - X^T Y
\end{aligned}$$

Setting this expression to zero we obtain

$$\Theta = \left(X^T X\right)^{-1} X^T Y$$

B Gradient Descent



(a) What is a convex problem? Why does gradient descent work for solving convex problems?

(b) Why descent direction is the direction that gives the largest decrease in function value only holds for small step size.

(c) If we visualize the optimization plane, gradient descent is much like a "downhill process," starting from a point with a high loss (the initial point), selecting an appropriate direction (the negative gradient direction), and taking a small step forward. Please reasonably analyze, based on the above figure, whether initial points A and B can both be optimized to the same final point.

Refer to Lecture5

C Maximum Likelihood estimator

We begin by considering a single binary random variable and following a Bernoulli distribution.

$$\operatorname{Bern}(x \mid \mu)=\mu^{x}(1-\mu)^{1-x}$$

(1) Please provide the mean and variance of this distribution and the maximum likelihood estimator.

It is easily verified that it has mean and variance given by

$$\begin{aligned}\mathbb{E}[x]&=\mu\\ \operatorname{var}[x]&=\mu(1-\mu).\end{aligned}$$

Now suppose we have a data set $\mathcal{D}=\{x_1,\ldots,x_N\}$ of observed values of x. We can construct the likelihood function, which is a function of μ, on the assumption that the observations are drawn independently from $p(x \mid \mu)$, so that

$$p(\mathcal{D} \mid \mu)=\prod_{n=1}^{N} p\left(x_n \mid \mu\right)=\prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n}$$

In a frequentist setting, we can estimate a value for μ by maximizing the likelihood function, or equivalently by maximizing the logarithm of the likelihood. In the case of the Bernoulli distribution, the log likelihood function is given by

$$\ln p(\mathcal{D} \mid \mu)=\sum_{n=1}^{N} \ln p\left(x_n \mid \mu\right)=\sum_{n=1}^{N}\left\{x_n \ln \mu+\left(1-x_n\right) \ln (1-\mu)\right\}.$$

At this point, it is worth noting that the log likelihood function depends on the N observations x_n only through their sum $\sum_n x_n$. This sum provides an example of a sufficient statistic for the data under this distribution, and we shall study the important role of sufficient statistics in some detail. If we set the derivative of $\ln p(\mathcal{D} \mid \mu)$ with respect to μ equal to zero, we obtain the maximum likelihood estimator

$$\mu_{\mathrm{ML}}=\frac{1}{N} \sum_{n=1}^{N} x_n$$

(2) Note that the log-likelihood function of bernoulli distribution is also known as cross-entropy cost function. In practical applications, we often first use softmax to predict valid probabilities, which are then input into the cross-entropy function to calculate the loss. Show that the softmax function is equivalent to the sigmoid function in the 2-class case with 0-1 labels.

$$\begin{aligned}\frac{e^{\mathbf{w}_1^T \mathbf{x}}}{e^{\mathbf{w}_1^T \mathbf{x}}+e^{\mathbf{w}_2^T \mathbf{x}}}&=\frac{1}{1+e^{\mathbf{w}_1^T \mathbf{x}-\mathbf{w}_2^T \mathbf{x}}}\\ &=\frac{1}{1+e^{-(\mathbf{w}_2-\mathbf{w}_1)^T \mathbf{x}}}\\ &=\sigma(\hat{\mathbf{w}}^T \mathbf{x})\end{aligned}$$

where $\hat{\mathbf{w}}=\mathbf{w}_2-\mathbf{w}_1$

D Your First learning algorithm-PLA

We introduced a simple learning algorithm called PLA. The weight update rule has the nice interpretation that it moves in the direction of classifying x(t) correctly. The update rule is

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

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

(b) Show that $y(t)\mathbf{w}^{\mathrm{T}}(t+1)\mathbf{x}(t) > y(t)\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t)$.

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

(a)

If $x(t)$ is misclassified by $w(t)$, then $w^T(t)x(t)$ has different signs of $y(t)$, thus $y(t)w^T(t)x(t) < 0$.

(b)

$$\begin{aligned} y(t)w^T(t+1)x(t) &= y(t)(w(t)+y(t)x(t))^T x(t) \\ &= y(t)\left(w^T(t)+y(t)x^T(t)\right)x(t) \\ &= y(t)w^T(t)x(t) + y(t)y(t)x^T(t)x(t) \\ &> y(t)w^T(t)x(t) \end{aligned}$$

(c)

From previous problem, we see that $y(t)w^T(t)x(t)$ is increasing with each update. If $y(t)$ is positive, but $w^T(t)x(t)$ is negative, we move $w^T(t)x(t)$ toward positive by increasing it. If however $y(t)$ is negative, but $w^T(t)x(t)$ is positive, $y(t)w^T(t)x(t)$ increases means $w^T(t)x(t)$ is decreasing, i.e. moving toward negative region.

So the move from $w(t)$ to $w(t+1)$ is a move "in the right direction" as far as classifying $x(t)$ is concerned.

E Logistic Regression

Answer the following multiple-choice questions, keeping in mind that there may be more than one correct answer.

(1) Is logistic regression a supervised machine learning algorithm?

- TRUE
- FALSE

(2) Is Logistic regression mainly used for regression?

- TRUE
- FALSE

(3) Is it possible to apply a logistic regression algorithm on a 3-class classification problem?

- TRUE
- FALSE

(4) Which of the following methods do we use to best fit the data in logistic regression?

- Least Square Error
- Jaccard distance
- Maximum Likelihood
- all of them

F Logistic Regression

We are given a sample in which each point has only one feature. Consider a binary classification problem in which sample values $x \in \mathbb{R}$ are drawn randomly from two different class distributions. The first class, with label $y = 0$, has its mean to the left of the mean of the second class, with label $y = 1$. We will use a modified version of logistic regression to classify these data points. We model the posterior probability at a test point $z \in \mathbb{R}$ as

$$P(y = 1|z) = s(z - \alpha),$$

where $\alpha \in \mathbb{R}$ is the sole parameter that we are trying to learn and $s(\gamma) = \frac{1}{1+e^{-\gamma}}$ is the logistic function. The decision boundary is $z = \alpha$ (because $s(z) = \frac{1}{2}$ there).

We will learn the parameter α by performing gradient descent on the logistic loss function. That is, for a data point x with label $y \in 0, 1$, we find the α that minimizes

$$J(\alpha) = -y \ln s(x - a) - (1 - y) \ln(1 - s(x - \alpha)).$$

(a) Derive the stochastic gradient descent update for J with step size $\epsilon > 0$, given a sample value x and a label y. Hint: feel free to use s as an abbreviation for $s(x - \alpha)$.

(b) Is $J(\alpha)$ convex over $\alpha \in \mathbb{R}$? Justify your answer.

(c) Now we consider multiple sample points. Since the feature dimension $d = 1$, we are given an $n \times 1$ design matrix X and a vector $y \in \mathbb{R}^n$ of labels. Consider batch gradient descent on the cost function $\sum_{i=1}^{n} J(\alpha; X_i, y_i)$. There are circumstances in which this cost function does not have a minimum over $\alpha \in \mathbb{R}$ at all. What is an example of such a circumstance?

(a) By the chain rule,

$$\frac{d}{d\alpha} s(x - \alpha) = -s(1 - s).$$

Hence,

$$\begin{aligned} J'(\alpha) &= \frac{ys(1-s)}{s} - \frac{(1-y)s(1-s)}{1-s} \\ &= y(1-s) - (1-y)s \\ &= y - s \end{aligned}$$

So the stochastic gradient descent update rule is

$$\alpha^{(t+1)} \leftarrow \alpha^{(t)} + \epsilon(s(x - \alpha) - y).$$

(b) Consider the second order derivative of J

$$J''(\alpha) = \frac{d}{d\alpha}(y - s) = s(1 - s).$$

As the logistic function is always in the range $(0, 1)$, $s(1 - s)$ is always positive, so $J(\alpha)$ is convex.

(c) If all the sample points are of only one class, then $J(\alpha)$ is either monotonically increasing or monotonically decreasing over $\alpha \in \mathbb{R}$.