Machine Learning. Homework 1.

Sofiia Petryshyn

APPS UCU, CS 2020

1 Weighted linear regression

$J(\theta) = (\frac{1}{2}) \sum_{i=1}^{n} \omega^i (\theta^T x^i - y^i)^2$

a)
Show that $J(\theta)$ can be written this way: $J(\theta) = (X\theta - \hat{y})^T W (X\theta - \hat{y})$.

Explanation of what W is.
X - input matrix, Y - output matrix. If we have a non-linear relationship between X and Y, we use weighted linear regression. This type of algorithm does not learn a fixed set of parameters, it computes θ for each x point. Bigger preference we give to the parameters θ lying closer to some x.
The matrix W is a non-negative diagonal matrix of weights. Each w^i associated with training point x^i. Typically, $w^i = exp(\frac{-(x^i - x)^2}{2\tau^2})$ if x_i is a vector, that $w^i = exp(\frac{-(x^i - x)^T \cdot (x^i - x)}{2\tau^2})$, where τ is bandwidth parameter. The parameter τ controls how quickly the weight of a training example falls off with distance of its x_i from the query point x.
w^i is bigger if the difference between x and x^i is smaller. Thus, the training-set-points that are closer to the x contribute more to the cost function $J(\theta)$ than the points that are further from x. Example of W matrix:

$$\begin{bmatrix} \mathrm{w}^i & 0 & 0 & 0 \\ 0 & \mathrm{w}^i & 0 & 0 \\ 0 & 0 & .. & 0 \\ 0 & 0 & 0 & \mathrm{w}^i \end{bmatrix}$$

Show that $J(\theta)$ can be written this way: $J(\theta) = (X\theta - \hat{y})^T W (X\theta - \hat{y})$.

$J(\theta) = (\frac{1}{2}) \sum_{i=1}^{n} \omega^i (\theta^T x^i - y^i)^2$

We can take out of the brackets diagonal matrix W,

$$J(\theta) = (\frac{1}{2}) \cdot W \sum_{i=1}^{n} (\theta^T x^i - y^i)^2$$

We can neglect the coefficient, rewrite the sum of squares in terms of matrices $((\theta X - \hat{y})^T \cdot (\theta X - \hat{y}))$ and w^i represent as a matrix:

$$J(\theta) = ((\theta X - \hat{y})^T \cdot W \cdot (\theta X - \hat{y}))$$

b)
If all the $w^i = 1$, than normal equation looks as follows: $X^T X \theta = X^T \hat{y}$.
Also in this case the value of θ gives us the highest precision:
$\theta = (X^T X)^{-1} X^T \hat{y}$.
Print the expression to find θ in weighted linear regression. Find the gradient $\bigtriangledown_\theta J(\theta)$ and, equating it to zero, derive the normal equation for finding θ. The expression will depend on X, W and $\hat{y}$.

As we know from the previous task,

$$J(\theta) = (\frac{1}{2}) \cdot ((\theta X - \hat{y})^T \cdot W \cdot (\theta X - \hat{y}))$$

$$J(\theta) = (\frac{1}{2}) \cdot ((\theta X - \hat{y})^T \cdot W \cdot (\theta X - \hat{y}))$$

$$J(\theta) = (\frac{1}{2}) \cdot (\theta^T X^T W X \theta - \theta^T X^T W \hat{y} - y^T W X \theta - y^T W y)$$

Now, to minimize the Jacobian function, we take a derivative, and in terms of the normal equation the derived equation going to be equalised to zero.

$$\bigtriangledown_\theta \mathrm{J}(\theta) = (\frac{1}{2}) \cdot \bigtriangledown_\theta tr(\theta^T X^T W X \theta - \theta^T X^T W \hat{y} - y^T W X \theta - y^T W y) =$$

$$= (\frac{1}{2}) \cdot (\bigtriangledown_\theta tr(\theta^T X^T W X \theta) - 2 \cdot \bigtriangledown_\theta tr(\theta^T X^T W y) - \bigtriangledown_\theta tr \hat{y}^T W y) =$$

$$= (\frac{1}{2}) \cdot 2(X^T W X \theta) + 2(X^T W y) =$$

$$= (X^T W X \theta) + (X^T W y) =$$

$$= (X^T W) \cdot (X \theta - y)$$

Normal equation:

$(X^T W) \cdot (X \theta - y) = 0$

$X^T W X \theta = X^T W y) = 0$

$\hat{\theta} = (X^T W X)^{-1} \cdot (X^T W y)$

2 Poisson regression and family of exponential models

Predict the number of calls to support your site on a given day. An entire number of people apply to the support service per day, which is usually no more than 7-10, so you decided to use the Poisson distribution to model such appeals.

a)

The Poisson probability distribution is as follows:

$$P(y;\lambda) = \frac{e^{-\lambda}\lambda^{y}}{y!}$$

Show that the Poisson distribution belongs to the exponential family and indicate to what $h(x), \eta(\lambda), T(x), B(\lambda)$

Exponential family is the class of probability models, where pmf or density can be written:

$$p(x|\lambda) = h(x) \cdot exp(\eta(\lambda)T(x) - B(\lambda))$$

Poisson distribution we can rewrite:

$$p(x|\lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!} =$$

$$= \frac{1}{x!}exp(x(log(\lambda) - \lambda)) =$$

$$= h(x) \cdot exp(\eta(\lambda)T(x) - B(\lambda))$$

, where $h(x) = \frac{1}{x!}$; $\eta(\lambda) = log(\lambda)$; $T(x) = x; B(\lambda) = \lambda$.

b)

What is canonical response function for this distribution.

In canonical form the density has the form:

$$p(x|\lambda) = h(x) \cdot exp(\eta T(x) - A(\eta))$$

, here $\eta = \eta(\lambda)$ - the Natural Parameter;
$A(\eta)$ instead of $B(\lambda)$ - normalization constant,
$log(A(\eta)) = \sum_{x \subset X} h(x)exp(\eta T(x))$.

For Poisson distribution:

$$p(x|\lambda) = h(x) \cdot exp(\eta T(x) - A(\eta))$$

, where $\eta = log(\lambda)$; $A(\eta) = B(\lambda) = \lambda = e^{\eta}$

c)
For the training sample $(x^i, y^i); i = 1, ...m$ the logarithmic function of plausibility (log-likelihood) will be:

$$L(\theta) = log(P(y^i|x^i;\theta))$$

Find the derivative $\frac{\partial}{\partial\theta}L(\theta)$ and formulate the rule of weight renewal by the method of stochastic gradient lifting, if y has a Poisson distribution and a canonical response function.

We are going to create a matrix Z equal to the training sample $(x^i, y^i); i = 1, ...m$. This way a defined value of $z^i = (x^i, y^i); i = 1, ...m$. The likelihood function is:

$$L(\lambda|z_1, ...z_m) = \prod_{i=1}^{m} exp(-\lambda)\frac{1}{z_i!}\lambda^{z_i}$$

If we apply logarithm to the above function, we are getting rid of the multiplication, so we are getting the sum:

$$L(\lambda|z_1, ...z_m) = \sum_{i=1}^{m}(-\lambda - \ln(x_i!) + x_i \ln(\lambda)) =$$

$$= -m\lambda - \sum_{i=1}^{m} \ln(z_i!) + \ln(\lambda)\sum_{i=1}^{m}(z_i)$$

Find the derivative $\frac{\partial}{\partial\lambda}L(\lambda)$:

$$\frac{\partial}{\partial\lambda}L(\lambda) = \frac{\partial}{\partial\lambda}(-m\lambda - \sum_{i=1}^{m} \ln(z_i!) + \ln(\lambda)\sum_{i=1}^{m}(z_i)) =$$

$$= -m + \frac{1}{\lambda}\sum_{i=1}^{m}(z_i))$$

Formulate the rule of weight renewal:

$$\theta_i = \theta_i + \alpha(\frac{\partial}{\partial\lambda}L(\lambda)) =$$

$$= \theta_i + \alpha * (-m + \frac{1}{\lambda}\sum_{i=1}^{m}(z_i)))$$

, where α is a small number of deviation of the model, while we are in search of the best fit.

3 Weighted logistic regression

Some models of teaching with a teacher, such as neural networks, for example, have the function of a hypothesis of very high variability. Intuitively, this means that they have a very "flexible" boundary of classes (decision boundary). However, their predictions are often difficult to explain. In some problems (for example, in automated diagnostics in medicine), explaining the solutions of machine learning models is as important as their accuracy. Without this, doctors do not trust the conclusions of the car.
However, we can modify logistic regression so that it is able to make qualitative predictions even in conditions of a very nonlinear class boundary (decision boundary). Predicting logistic regression is fairly easy to explain. Your task is to modify the logistic regression so that it predicts as accurately as possible the nearest points to the query point (query point), similar to the weighted linear regression.
To calculate the weights, we will use the same formula that was used for the weighted linear regression, written in vector form:

$$\omega^i = exp(-\frac{(x^i-x)^T(x^i-x)}{2\tau^2})$$

a)
Derive the formula of the cost function for weighted logistic regression.
As logistic regression uses sigmoid function to classify the output of the model. So to minimize the loss/cost function we use cross-entropy loss, or log loss. This function measures the performance of a classification model where the output is probability values in range 0 and 1.
Here is a definition of the hypothesis function, which is sigmoid function of the logistic model.

$$h_\theta(x) = g(\theta^T X) = \frac{1}{1+e^{-(\theta^T X)}} * w^i$$

Here I would also show the derivative of the sigmoid, as we will need it further.

$$\frac{\partial\sigma(x)}{\partial(x)} = w^i * (\frac{0*(1+e^{-x})-(e^{-x}*(-1))}{(1+e^{-x})^2}) =$$

$$= w^i * (\frac{e^{-x}}{(1+e^{-x})^2}) = w^i * (\frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2}) =$$

$$= w^i * (\frac{1}{(1+e^{-x})} * (1 - \frac{1}{(1+e^{-x})})) = w^i * \sigma(x) * (1-\sigma(x))$$

To minimize the cost function we are taking a log of it. As the cost function must classify the data, it will exist in two cases prediction is 0 or 1.

$$cost(h_\theta(x), y)) = \begin{cases} \text{- log(h}_\theta(x)), & \text{if } y = 1 \\ \text{- log(1 - h}_\theta(x)), & \text{if } y = 0 \end{cases}$$

Combining both the equation:

$$cost(h_\theta(x), y)) = -ylog(h_\theta(x)) - (1-y)log(1-h_\theta(x))$$

To normalize the function and represent it for all the values of the vectors, we will rewrite it as follows:

$$J(h_\theta(x), y)) = -\frac{1}{m}\sum_{i=1}^{m} \cdot cost(h_\theta(x), y)) =$$

$$= -\frac{1}{m}\sum_{i=1}^{m} \cdot (ylog(h_\theta(x)) + (1-y)log(1-h_\theta(x))) =$$

$$= -\frac{1}{m}\sum_{i=1}^{m} \cdot (y \cdot log(\frac{w^i}{1+e^{-(\theta^T X)}}) + (1-y)log(1 - \frac{w^i}{1+e^{-(\theta^T X)}}))$$

So, we have got the cost function.

b)
Derive the rule of updating the scales θ for weighted logistic regression by the gradient descent method.
Gradient descent method tells us:

$$\theta_j = \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

We have to take the derivative of $J(\theta)$.

$$\frac{\partial J(\theta)}{\partial(\theta_j)} = -\frac{1}{m} \cdot (\sum_{i=1}^{m}[y_i * \frac{w^i}{h_\theta(x)} * \sigma(x) * (1-\sigma(x))\frac{\partial\theta^T x}{\partial\theta_j}] + \sum_{i=1}^{m}[(1-y_i) * \frac{w^i}{1-h_\theta(x)} * (-\sigma(x) * (1-\sigma(x)))\frac{\partial\theta^T x}{\partial\theta_j}])$$

$$\frac{\partial J(\theta)}{\partial(\theta_j)} = -\frac{1}{m} \cdot (\sum_{i=1}^{m}[y_i * \frac{w^i}{h_\theta(x)} * h_\theta(x^i) * (1-h_\theta(x^i)) * x_j^i] + \sum_{i=1}^{m}[(1-y_i) * \frac{w^i}{1-h_\theta(x)} * (-h_\theta(x^i) * (1-h_\theta(x^i))) * x_j^i])$$

$$\frac{\partial J(\theta)}{\partial(\theta_j)} = -\frac{1}{m} \cdot (\sum_{i=1}^{m}[y^i * w^i - h_\theta(x^i) * w^i] * x_j^i) = -\frac{1}{m} \cdot (\sum_{i=1}^{m}[w^i * (y^i - h_\theta(x^i))] * x_j^i)$$

Convert the formula into matrix form:

$$\frac{\partial J(\theta)}{\partial(\theta_j)} = \frac{1}{m} \cdot X^T \cdot W \cdot [h_\theta(x) - y]$$

c)

Derive the solution of weighted logistic regression by the method of normal equations.

In this section we have to equalise the derivative of cost function to zero.

$$\frac{\partial J(\theta)}{\partial(\theta_j)} = \frac{1}{m} \cdot X^T \cdot W \cdot [h_\theta(x) - y] = 0$$

As it is non-linear regression, so that we have a deal with logistic regression, we have a convex function and we cannot find the optimum in closed form solution.

From the equation above we would have $\frac{1}{m} = 0$, what is not true;

$X^T = \hat{0}$, $W = \hat{0}$, that are also not trues, $[h_\theta(x) - y]$, which would mean that the model is perfect, what we call overfitting, and what is impossible.

4 Scaling of features in linear regression

a)

You have constructed a linear regression to predict fuel consumption per 100 km. way. You used the weight of the passenger car, the engine capacity and the speed in kilometers per hour as features. After the introduction of the model into production, it turned out that the sign of speed comes not in kilometers per hour, but in miles per hour. Your model is already implemented on the chip, it is very expensive to change it or the format of the input data. However, changing the prediction before giving it to the user is easy and cheap. How can the prediction be modified so that it is correct?

As we have different units than those that we want, we have to normalize these features. In this case we have miles/hour, but we need km/hour, we know that it is necessary to make the units of input the same as output. We know that km/hour is just miles/hour plus a constant. The same property would work when we divide the values by the standart deviation. Now the feature miles/hour has the standart deviation equal to 1.

The standart deviation is the square root of varience, the variance is the expected value of the square of the differences between the actual values and the mean ($Var = \frac{\sum_{i=1}^{n}(x_i - x)^2}{n-1}$) of the distribution.

So, as we have $\sigma = 1$ the Euclidean distances would not be large, we would not have a problem of bigger numbers in units 'mile/hour' and smaller numbers of unit 'km/hour'. In this case everything is standardised.