Fikri Farhan Witjaksono

Machine Learning 2020

1. Consider a dataset of N items, in which the i^{th} item has two elements: one real-valued input x_i and the respective output y_i i.e., the set $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\}$

We use the following normal (Gaussian) model to fit the data, which has an unknown parameter w (the variance is known in advance and is set to 1):

$$y_i \sim \mathcal{N}(\log(wx_i), 1)$$

(a) Describe a maximum likelihood approach to infer w and write down the log-likelihood objective for this problem.

Answer:

a) y_i is distributed as $\mathcal{N}(\mu, \sigma^2)$, where $\mu = \log(wx_i)$ and $\sigma^2 = 1$

$$P(y_i; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - \log(wx_i))^2}{2}\right)$$

The full data likelihood is written by (assuming i.i.d):

$$L = \prod_{i=1}^{N} p(y_i; \mu, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - \log(wx_i))^2}{2}\right)$$

$$= (2\pi)^{-N/2} \prod_{i=1}^{N} \exp\left(-\frac{(y_i - \log(wx_i))^2}{2}\right)$$

Then, log-likelihood is:

$$LL = \log\left[(2\pi)^{-N/2} \prod_{i=1}^{N} \exp\left(-\frac{(y_i - \log(wx_i))^2}{2}\right)\right]$$

$$= -\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{N}(y_i - \log(wx_i))^2)$$

To infer the unknown parameter w, we maximize the log-likelihood, which means we maximize the probability/likelihood that the data is generated by the distribution.

(b) Complete the right side of the following equation for the maximum log-likelihood solution. Write down your calculation.

$$\sum_{i=1}^{N} y_i = ?$$

Answer:

To maximize the likelihood, we maximize log-likelihood as it is computationally ~~so~~ easier to work with:

For this, we take the derivative of LL w.r.t. w - it to zero:

$$\frac{\partial LL}{\partial w} = 0 \implies \frac{\partial \sum_{i=1}^{N}(y_i - \log(wx_i))^2}{\partial w} = 0$$

$$\Longrightarrow \sum_{i=1}^{N} \frac{x_i}{w x_i} \left(y_i - \log(w x_i)\right) = 0$$

$$\Longrightarrow \frac{1}{w} \sum_{i=1}^{N} \left(y_i - \log(w x_i)\right) = 0$$

$$\Longrightarrow \sum_{i=1}^{N} y_i = \sum_{i=1}^{N} \log(w x_i) = \sum_{i=1}^{N} \log(x_i) + N \log w$$

c) We add $-\alpha \|w\|_2^2$ to the full log-likelihood objective, where the hyperparameter α is fixed in advance ($\alpha \geq 0$). Complete the equation for the maximum log-likelihood solution of this new objective.

Answer:

The scalar w we have: $\|w\|_2^2 = w^2$

the new log-likelihood is:

$$LL^{new} = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \log(w x_i)\right)^2 - \alpha w^2$$

Thus the derivative is:

$$\frac{\partial LL^{new}}{\partial w} = -\frac{1}{2} \frac{\partial}{\partial w} \left(\sum_{i=1}^{N} \left(y_i - \log(w x_i)\right)^2 \right) - \frac{\partial}{w}(w^2) \cdot \alpha$$

$$= -\frac{1}{2} \sum_{i=1}^{N} \left(-\frac{2}{w}\right)\left(y_i - \log(w x_i)\right) - 2w\alpha$$

$$= \frac{1}{w} \sum_{i=1}^{N} \left(y_i - \log(w x_i)\right) - 2w\alpha$$

$$\frac{\partial LL^{new}}{\partial w} = 0 \Longrightarrow + \frac{1}{w} \sum_{i=1}^{N} \left(y_i - \log(w x_i)\right) - 2w\alpha = 0$$

$$\Longrightarrow 2\alpha w^2 = \sum_{i=1}^{N} \left(y_i - \log(w x_i)\right)$$

$$\Longrightarrow \sum_{i=1}^{N} y_i = \sum_{i=1}^{N} \log(w x_i) + 2\alpha w^2$$

d) What does happen if $\alpha \to \infty$ in the new objective of part (c)?

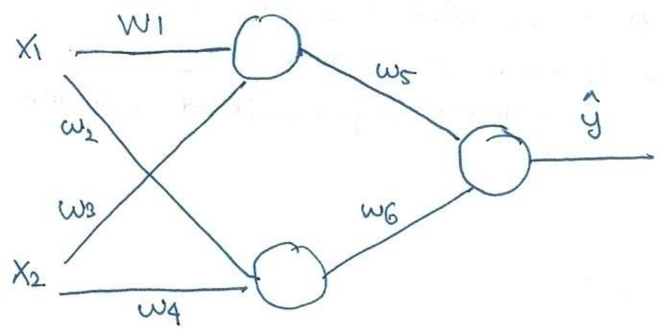
Answer:

If $\alpha \to \infty \Longrightarrow -\alpha \|w\|_2^2 \longrightarrow -\infty$

Since we want to maximize the likelihood, then $\|w\|_2^2$ should become small, i.e. $w \longrightarrow 0$ as $\alpha \longrightarrow \infty$ in order to maximize the likelihood.

Thus, the regularizer shifts w towards zero!

2) Given a set of N triplets $\{(x_{i1}, x_{i2}, y_i)\}$, $1 \leq i \leq N$, the goal is to design a model to predict y_i based on the input attributes x_{i1} and x_{i2}. For this, we use the following neural network.



In this model $w_1, \ldots, w_6$ are the free parameters that should be learned. The (nonlinear) activation is defined as (q is a hyperparameter which is fixed in advance):

$$f(z) = \begin{cases} (z+|z|)^q & \text{if } z \geq 0 \\ (z-|z|)^q & \text{otherwise} \end{cases}$$

The error of the network is measured by

$$\mathcal{E} = \frac{1}{N} \sum_{i=1}^{N} (y-\hat{y})^2$$

(a) Write down the gradients of the error $\mathcal{E}$ wrt all the parameters. Show an outline of your derivation (you do not need to compute the exact derivatives, but sufficiently describe the outline).

Answer:

We first introduce some notations:

$\hat{y}$ = output of neural network

z_0 = input to the activation of the output node.

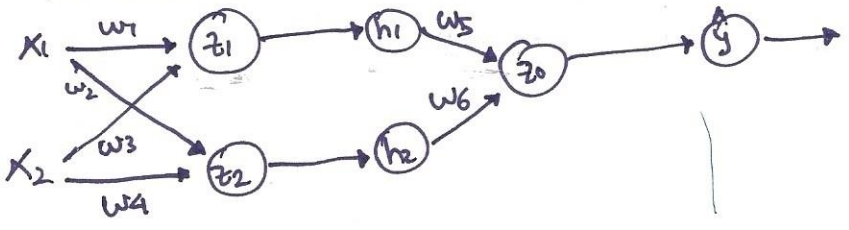
h_1 = the output of the first hidden node

h_2 = the output of the second hidden node.

z_1 = input to the activation of first hidden node

z_2 = input to the activation of second hidden node

The network then can be seen as:



$$\frac{\partial \mathcal{E}}{\partial w_5} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_5} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial w_5}$$

$$\frac{\partial \mathcal{E}}{\partial w_6} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_6} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial w_6}$$

$$\frac{\partial \mathcal{E}}{\partial w_1} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial w_1} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \mathcal{E}}{\partial w_2} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial w_2} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_2} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_2}$$

$$\frac{\partial \mathcal{E}}{\partial w_3} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_3} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial w_3} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_3} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_3}$$

$$\frac{\partial \mathcal{E}}{\partial w_4} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_4} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial w_4} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_4} = \frac{\partial \mathcal{E}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_0} \cdot \frac{\partial z_0}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_4}$$

b) Describe a gradient descent algorithm to estimate the parameters.

Answer:

1. Initialize the parameters and set a learning rate r.
2. Repeat:

$$w_j^{(t+1)} = w_j^{(t)} - r \frac{\partial \mathcal{E}^{(t)}}{\partial w_j^{(t)}}, \quad j = 1, \dots 6$$

until convergence

↳ the error does not change or the parameters do not change anymore.

$\Longrightarrow$ As a result, it will yield a local optima.

c) For $q=1$, can you derive an equivalent but simpler neural network (i.e. a network without a hidden layer)? Prove your answer.

Answer:

The activation function can be written as:

$$f(z) = (2z)^q$$

If $q=1 \Longrightarrow f(z) = 2z \Longrightarrow$ It is just a simple linear transformations.

Thus:

$$\left.\begin{aligned} \hat{y} &= f(z_0) = 2z_0 = 2(w_5 h_1 + w_6 h_2) \\ h_1 &= 2z_1 = 2(w_1 x_1 + w_3 x_2) \\ h_2 &= 2z_2 = 2(w_2 x_1 + w_4 x_2) \end{aligned}\right\}$$

$\Rightarrow \hat{y} = 2\left(w_5\left(2\left(w_1x_1 + w_3x_2\right)\right) + w_6\left(2\left(w_2x_1 + w_4x_2\right)\right)\right)$

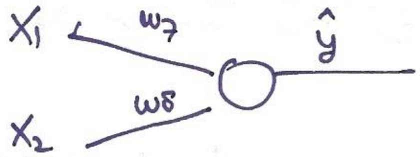
$= 4w_1w_5x_1 + 4w_3w_5x_2 + 4w_2w_6x_1 + 4w_4w_6x_2$

$= (4w_1w_5 + 4w_2w_6)x_1 + (4w_3w_5 + 4w_4w_6)x_2$

$\Rightarrow \hat{y} = w_7x_1 + w_8x_2$

where $\begin{cases} w_7 = 4w_1w_5 + 4w_2w_6 \\ w_8 = 4w_3w_5 + 4w_4w_6 \end{cases}$

thus, we can replace the network with a simpler network that just computes the weighted sum of the input attributes:



and we <u>do not</u> need any activation function $f(z)$

(d) For $g=1$, is the model equivalent to a linear regression model? Explain your answer.

<u>Answer:</u>

Yes; as we saw in (c), for $g=1$ we have $\hat{y} = w_7x_1 + w_8x_2$

i.e.: $\hat{y} = w^Tx$, where $w^T = [w_7, w_8]$

This is just the formulation of a basic linear regression model.

3.

(a) What would be the training error of the optimal linear SVM? Explain your answer.

<u>Answer:</u>

(a) the training error would be related to misclassification error, i.e., number of <u>training</u> items that are wrongly classified.

In this dataset, for optimal solution, no item would be misclassified, thus training error is zero.

b) Due to computational bottlenecks, we pick a subset of the items which would yield exactly the same solution as the SVM on the original data. Which items would you choose (mark them)? Explain your answer.

Answer:

the support vectors corresponding to γ's:

In optimum: $t_{new} = \text{sign}\left(\sum_{n=1}^{N} \alpha_n t_n x_n^T x_{new} + b\right)$, $b = t_n - \sum_{m=1}^{N} \alpha_m t_m x_m^T x_m$

Thus, zero α_n's will not have an impact on t_{new}, b, and they can be discarded. Therefore, we can train the model using only the support vectors.

c) Assume the data dimensionality is d and the number of training data point is N. Then, what is the computational complexity for predicting the class label of a new test data point?

Answer:

$O(d)$ is the computational complexity.

Computing w^*, b in general takes $\underbrace{O(d*N)}_{w^*}$, $\underbrace{O(N*d)}_{b}$, but they can be computed independent of test (prediction).

In other words, they can be obtained immediately after training and do NOT depend on particular test data point $\Rightarrow$ we do not take them into account for test.

Note that if number of support vectors $\ll N$

$\Longrightarrow$ Complexity of w^*, b would be $O(d)$ and $O(d)$ (instead of $O(N*d)$)

$\Rightarrow$ For test: we need to compute $\underbrace{w^* x_{new}}_{O(d)}$, and $\text{sign}(w^* x_{new} + b)$

$\Rightarrow$ test complexity $= O(d)$

Clustering / Unsupervised learning

4. a) Consider the k-means cost function for clustering N d-dimensional data points into k clusters. Compute the optimal parameters when $k=1$.

Answer:

K-means cost function is written by:

$$R(\mu, Z; X) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \| x_n - \mu_k \|_2^2 \, , \; z_{nk} \in \{0,1\} \, , \; \sum_k z_{nk} = 1$$

If $k=1 \implies z_{nk}$ is always 1

$$R(\mu, Z; X) = \sum_{n=1}^{N} z_{nk} \| x_n - \mu_k \|_2^2$$

$$= \sum_{n=1}^{N} z_{nk} (x_n - \mu_k)^T (x_n - \mu_k)$$

there is only one mean: $\implies$

$$R(\mu, Z; X) = \sum_{n=1}^{N} (x_n - \mu)^T (x_n - \mu)$$

$$\frac{\partial R}{\partial \mu} = \frac{\partial \sum_{n=1}^{N} (x_n - \mu)^T (x_n - \mu)}{\partial \mu} = 2 \sum_{n=1}^{N} x_n - \mu$$

$$\frac{\partial R}{\partial \mu} = 0 \implies \sum_{n=1}^{N} (x_n - \mu) = 0 \implies \mu = \frac{1}{N} \sum_{n=1}^{N} x_n$$

b) Derive the optimal parameters of the model when $k=N$.
Now, we use Gaussian Mixture Model (GMM) to cluster the data, wherein the covariance matrices are fixed and given in advance. In addition, we aim to obtain the correct number of clusters via Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC)

Answer:

We know $R(\mu, Z; X) \geq 0$

then, if we show that for a solution $R(\mu, Z; X) = 0$, then this solution will be the optimal solution, because it yields the minimum possible cost.

If $\mu_k = x_k \quad \forall 1 \leq k \leq N$

$$\Longrightarrow R(\mu, Z; X) = \sum_{i=1}^{N} z_{nk} \| x_n - x_n \|_2^2 = 0$$

thus, the solution in which each mean is exactly one of the data points yields $R=0$ which is an optimal solution.

c) Assume we know that all the estimated clusters should have the same size. Describe how you would apply AIC and BIC to compute the correct number of clusters. Discard the steps for calculating the log-likelihoods.

Answers:

for $k^{min} \leq k \leq k^{max}$

choose k^*_{AIC} and k^*_{BIC} for which AIC and BIC numbers are minimal, i.e.:

$$k^*_{AIC} = \arg\min_k AIC_k$$

$$k^*_{BIC} = \arg\min_k BIC_k$$

AIC_k is obtained by: $\underbrace{-ll}_{\text{negative log-likelihood}} + \underbrace{c(U,Z)}_{\text{complexity}}$

BIC_k is obtained by: $-ll + \frac{1}{2} c(U,Z) \ln N$

$$c(U,Z) = \underbrace{k * d}_{\text{related to means}} + \underbrace{1}_{\text{related to } \pi_k\text{'s}} \to 0 \; (N/k)$$

If we know the size of one clusters (π_1) / the size of other cluster is the same.

* all clusters have the same size