linear-classification

Shumin An

May 2019

1 Introduction

Actually, we can infer some machine learning algorithms from Linear Regression. The reasons are as follows:

1. Linear
   (a) Feature Linear (polynominal regression)
   (b) Global Linear (linear classification)
   (c) Parameter Linear (neural network)
2. Global (decision tree)
3. Non-processed data (PCA)

First, we focus on **1.a Feature Linear**. I call it because in linear regression, the power of feature$(x - i)$ equals to 1. If we break that rule, we will have a polynominal expression. For example:

$$w_1 x^2 + w_2 x^3 + ... + w_n x^{n+1}$$

Second, about **1.b Global Linear**. The output of linear regression is a linear sum without any further processing. So if we add a non-linear function(activation function), then it comes to the Linear classification.

Third, for **Parameter Linear**. This one is a little bit hard to explain. What I thought is the final w of a given regression is constant, but for neural network, if we change a initial value, then the result may also changed.

For **2.Global**. We didn't separate the space. Like in two-dimentional space, there's only one expression. Not like for x-axis in [0,1] is a expression, and for [1,2] is another expression. So decision tree actually divides the space into different parts and do further computation.

The last one is **3.Non-processed data**. In Linear Regression, seems like we do not process the data and use it directly. So what about feature selection for dimension reduction. Here comes to the PCA and other processing techniques.

2 Perception

The activation function is:

$$sign(a) = \begin{cases} 1, a \geq 0 \\ -1, a<0 \end{cases}$$

And we could define our loss function:

$$L(w) = \sum_{x_i \in \mathcal{D}_{wrong}} -y_i w^T x_i$$

So the partial derivative is:

$$\frac{\partial}{\partial w} L(w) = \sum_{x_i \in \mathcal{D}_{wrong}} -y_i x_i$$

Every time we update w, we use:

$$w^{t+1} \leftarrow w^t + \lambda y_i x_i$$

3 LDA

In LDA, the idea is we choose a direction first, and project the sample data along that direction(e.g $w^T x$). After projection, our data should meet the conditions as follows:

1. shorten inner-class distance
2. lengthen inter-class distance

To deal with the No.1, the first thing comes to our mind is to minimize the variance of each class. So assume:

$$z = w^T \cdot x (= |w| \cdot |x| \cos\theta)$$

and N1, N2 represents the number of sample data each class. According the our goal, we have:

$$
\begin{aligned}
C_1 : Var_z[C_1] &= \frac{1}{N_1}\sum_{i=1}^{N_1}(z_i - \overline{z_{c1}})(z_i - \overline{z_{c1}})^T \\
&= \frac{1}{N_1}\sum_{i=1}^{N_1}(w^T x_i - \frac{1}{N_1}\sum_{j=1}^{N_1} w^T x_j)(w^T x_i - \frac{1}{N_1}\sum_{j=1}^{N_1} w^T x_j)^T \\
&= w^T \frac{1}{N_1}\sum_{i=1}^{N_1}(x_i - \overline{x_{c1}})(x_i - \overline{x_{c1}})^T w \\
&= w^T S_1 w \qquad (1) \\
C_2 : Var_z[C_2] &= \frac{1}{N_2}\sum_{i=1}^{N_2}(z_i - \overline{z_{c2}})(z_i - \overline{z_{c2}})^T \\
&= w^T S_2 w \qquad (2)
\end{aligned}
$$

The inner-class distance is:

$$Var_z[C_1] + Var_z[C_2] = w^T(S_1 + S_2)w$$

For No.2, we could use the mean of each class:

$$
\begin{aligned}
(\overline{z_{c1}} - \overline{z_{c2}})^2 &= (\frac{1}{N_1}\sum_{i=1}^{N_1} w^T x_i - \frac{1}{N_2}\sum_{i=1}^{N_2} w^T x_i)^2 \\
&= (w^T(\overline{x_{c1}} - \overline{x_{c2}}))^2 \\
&= w^T(\overline{x_{c1}} - \overline{x_{c2}})(\overline{x_{c1}} - \overline{x_{c2}})^T w \qquad (3)
\end{aligned}
$$

Considering two conditions above, we could get our object function:

$$
\begin{aligned}
\hat{w} = \underset{w}{argmax}\, J(w) &= \underset{w}{argmax}\, \frac{(\overline{z_{c1}} - \overline{z_{c2}})^2}{Var_z[C_1] + Var_z[C_2]} \\
&= \underset{w}{argmax}\, \frac{w^T(\overline{x_{c1}} - \overline{x_{c2}})(\overline{x_{c1}} - \overline{x_{c2}})^T w}{w^T(S_1 + S_2)w} \\
&= \underset{w}{argmax}\, \frac{w^T S_b w}{w^T S_w w} \qquad (4)
\end{aligned}
$$

Actually, we only expect the direction but not value of w, so we have:

$$
\begin{aligned}
&\frac{\partial}{\partial w} J(w) = 2S_b w(w^T S_w w)^{-1} - 2w^T S_b w(w^T S_w w)^{-2} S_w w = 0 \\
&\Longrightarrow S_b w(w^T S_w w) = (w^T S_b w) S_w w \\
&\Longrightarrow w \propto S_w^{-1} S_b w = S_w^{-1}(\overline{x_{c1}} - \overline{x_{c2}})(\overline{x_{c1}} - \overline{x_{c2}})^T w \propto S_w^{-1}(\overline{x_{c1}} - \overline{x_{c2}}) \qquad (5)
\end{aligned}
$$

$S_w^{-1}(\overline{x_{c1}} - \overline{x_{c2}})$ is the direction we are looking for.

4 Logistic Regression

From Bayesian Method, we have:

$$p(C_1|x) = \frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)}$$

let $a = \ln \frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)}$ so:

$$p(C_1|x) = \frac{1}{1 + \exp(-a)}$$

Here's the sigmoid function. But why we need that? I think its just because we want a mapping which gives

$$w^T x -> [0, 1]$$

So for a observation, the probability of classification y is:

$$p(y|x) = p_1^y p_0^{1-y}$$

Then for N times observation:

$$\hat{w} = \underset{w}{argmax}\, J(w) = \underset{w}{argmax} \sum_{i=1}^{N} (y_i \log p_1 + (1 - y_i) \log p_0)$$

$$p_1' = (\frac{1}{1 + \exp(-a)})' = p_1(1 - p_1)$$

$$J'(w) = \sum_{i=1}^{N} y_i(1 - p_1)x_i - p_1 x_i + y_i p_1 x_i = \sum_{i=1}^{N} (y_i - p_1)x_i$$

5 GDA

So far, this is the most difficult part I think. The idea is modeling based on joint probability distribution, and using MAP to solve the parameters. Here comes to our assumption:

1. $y \sim Bernoulli(\phi)$
2. $x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$
3. $x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma)$

Then the MAP is:

$$\underset{\phi,\mu_0,\mu_1,\Sigma}{argmax} \log p(X|Y)p(Y) = \underset{\phi,\mu_0,\mu_1,\Sigma}{argmax} \sum_{i=1}^{N} (\log p(x_i|y_i) + \log p(y_i))$$

$$= \underset{\phi,\mu_0,\mu_1,\Sigma}{argmax} \sum_{i=1}^{N} ((1 - y_i) \log \mathcal{N}(\mu_0, \Sigma) + y_i \log \mathcal{N}(\mu_1, \Sigma) + y_i \log \phi + (1 - y_i) \log(1 - \phi)) \tag{6}$$

First we focus on ϕ, the partial derivative is:

$$\sum_{i=1}^{N} \frac{y_i}{\phi} + \frac{y_i - 1}{1 - \phi} = 0$$

$$\Longrightarrow \phi = \frac{\sum\limits_{i=1}^{N} y_i}{N} = \frac{N_1}{N} \tag{7}$$

Next for μ:

$$\begin{aligned} \hat{\mu_1} &= \underset{\mu_1}{argmax} \sum_{i=1}^{N} y_i \log \mathcal{N}(\mu_1, \Sigma) \\ &= \underset{\mu_1}{argmin} \sum_{i=1}^{N} y_i (x_i - \mu_1)^T \Sigma^{-1} (x_i - \mu_1) \end{aligned} \tag{8}$$

Since:

$$\sum_{i=1}^{N} y_i (x_i - \mu_1)^T \Sigma^{-1} (x_i - \mu_1) = \sum_{i=1}^{N} y_i x_i^T \Sigma^{-1} x_i - 2 y_i \mu_1^T \Sigma^{-1} x_i + y_i \mu_1^T \Sigma^{-1} \mu_1$$

We multiplied Σ on the left side:

$$\sum_{i=1}^{N} -2 y_i \Sigma^{-1} x_i + 2 y_i \Sigma^{-1} \mu_1 = 0$$

$$\Longrightarrow \mu_1 = \frac{\sum\limits_{i=1}^{N} y_i x_i}{\sum\limits_{i=1}^{N} y_i} = \frac{\sum\limits_{i=1}^{N} y_i x_i}{N_1} \tag{9}$$

Same for μ_0:

$$\mu_0 = \frac{\sum\limits_{i=1}^{N} (1 - y_i) x_i}{N_0}$$

The biggest trouble is solving Σ, we have:

$$\begin{aligned} \sum_{i=1}^{N} \log \mathcal{N}(\mu, \Sigma) &= \sum_{i=1}^{N} \log\left(\frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}}\right) + \left(-\frac{1}{2} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu)\right) \\ &= Const - \frac{1}{2} N \log |\Sigma| - \frac{1}{2} Trace((x_i - \mu)^T \Sigma^{-1} (x_i - \mu)) \\ &= Const - \frac{1}{2} N \log |\Sigma| - \frac{1}{2} Trace((x_i - \mu)(x_i - \mu)^T \Sigma^{-1}) \\ &= Const - \frac{1}{2} N \log |\Sigma| - \frac{1}{2} N Trace(S \Sigma^{-1}) \end{aligned} \tag{10}$$

Since:

$$\frac{\partial}{\partial A}(|A|) = |A|A^{-1} \tag{11}$$

$$\frac{\partial}{\partial A}Trace(AB) = B^T \tag{12}$$

So we have:

$$\begin{aligned}
&[\sum_{i=1}^{N}((1-y_i)\log\mathcal{N}(\mu_0,\Sigma) + y_i\log\mathcal{N}(\mu_1,\Sigma)]' \\
&= Const - \frac{1}{2}N\log|\Sigma| - \frac{1}{2}N_1 Trace(S_1\Sigma^{-1}) - \frac{1}{2}N_2 Trace(S_2\Sigma^{-1})
\end{aligned} \tag{13}$$

And:

$$\begin{aligned}
N\Sigma^{-1} - N_1 S_1^T\Sigma^{-2} - N_2 S_2^T\Sigma^{-2} &= 0 \\
\Longrightarrow \Sigma &= \frac{N_1 S_1 + N_2 S_2}{N}
\end{aligned} \tag{14}$$