Logistic Regression Model

Fish

2019 年 6 月 18 日

Content

1 the exponential family

to work our way up to GLMs, we will begin by defining exponential family distributions. We say that a class of distributions is in the exponential family if it can be written in the form of

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta)) \tag{1}$$

here, η is called the natural parameter (also called the canonical parameter) of the distribution; T(y) is the sufficient statistic (for the distributions we consider, it will often be the case that T(y) = y); and $a(\eta)$ is the log partition function. The quantity $e^{-a(\eta)}$ essentially plays the role of a normalization constant, that makes sure the distribution $p(y;\eta)$ sums/integrates over y to 1.

A fixed choice of T, a and b defines a family (or set) of distributions that is parameterized by η; as we vary η, we then get different distributions within this family.

we now show that the Bernouli and the Gaussian distributions are examples of exponential family distributions.

1.1 Bernouli distribution

the Bernouli distribution with mean ϕ, written Bernouli(ϕ), specifies a distribution over $y \in \{0, 1\}$, so that $p(y = 1;\phi) = \phi$; $p(y = 0;\phi) = 1 - \phi$. As we varying ϕ, we obtain Bernouli distributions with different means. We now show that this class of Bernouli distributions, ones obtained by varying ϕ, is in the exponential family; i.e., that there is a choice of T, a and b so that Equation (1) becomes exactly the class of Bernouli distributions.

we write the Bernouli distributin as:

$$\begin{aligned} p(y;\phi) &= \phi^y(1-\phi)^{1-y} \\ &= \exp(y\log\phi + (1-y)\log(1-\phi)) \\ &= \exp((\log(\frac{\phi}{1-\phi}))y + \log(1-\phi)) \end{aligned} \tag{2}$$

thus, the natural parameter is given by $\eta = \log(\phi/(1-\phi))$. Interestingly, if we invert this definition for η by solving for ϕ in terms of η, we obtain $\phi = \frac{1}{(1+e^{-\eta})}$. This is the familiar sigmoid function! This will come up again when we derive logistic regression as a GLM. To complete the formulation of the Bernouli distribution as an exponential family distribution, we also have

$$T(y) = y \tag{3}$$

$$\begin{aligned} a(\eta) &= -log(1-\phi) \\ &= \log(1+e^{\eta}) \end{aligned} \tag{4}$$

$$b(y) = 1 \tag{5}$$

- focus on that in the process of derivation, thear is an Logistic equation

$$\phi = \frac{1}{1+e^{-\eta}} \tag{6}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{7}$$

- sigmoid or logistic function

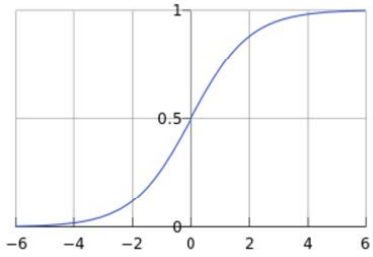

Fig 1: sigmoid or logistic function

- derivation of sigmoid function $f(x) = \frac{1}{1+e^{-x}}$

$$\begin{aligned} f'(x) &= (\frac{1}{1+e^{-x}})' \\ &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}} \cdot (1 - \frac{1}{1+e^{-x}}) \\ &= f(x) \cdot (1 - f(x)) \end{aligned} \tag{8}$$

1.2 Gaussian distribution

lets now move on to consider the Gaussian distribution. Recall that, when deriving linear regression, the value of σ^2 had no effect on our final choice of θ and $h_\theta(x)$. Thus, we can choose an arbitrary value for σ^2 without changing anything. To simlpify the derivation below, let set $\sigma^2 = 1$. We then have:

$$\begin{aligned} p(y;\mu) &= \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}(y-\mu)^2) \\ &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}y^2) \cdot \exp(\mu y - \frac{1}{2}\mu^2) \end{aligned} \tag{9}$$

$whether:$

$$\begin{aligned} \eta &= \mu \\ T(y) &= y \\ a(\eta) &= \mu^2/2 = \eta^2/2 \\ b(y) &= (1/\sqrt{2\pi})\exp(-y^2/2) \end{aligned}$$

2 gradient descent algorithm(BGD SGD MBGD)

- initialize θ (randomly initialize)

- iteration along the subtractive gradient direction, θ updated let $J(\theta)$ smaller

$$\theta = \theta - \alpha \cdot \frac{\partial J(\theta)}{\partial \theta} \qquad \alpha : \text{learning rate、step} \tag{10}$$

- gradient descent algorithm

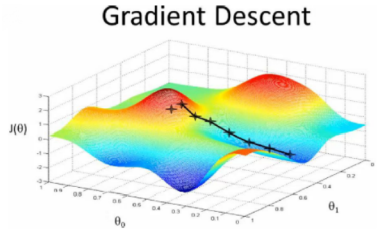

Fig 2: gradient descent algorithm

- gradient direction

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j}\frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2 \cdot \frac{1}{2}(h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j}(h_\theta(x) - y) \\
&= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j}(\sum_{i=0}^{m} \theta_i x_i - y) \\
&= (h_\theta(x) - y)x_j
\end{aligned} \tag{11}$$

- BGD、SGD、MBGD

  (1) bach gradient descent, teh original form, mean that update gradient using all sample every iteration.

  (a) partial derivate target function

$$\frac{\nabla J(\theta_0, \theta_1)}{\nabla \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)x_j^i \tag{12}$$

among $i = 1, 2, ..., m$ show sample number, $j = 0, 1$ show feature number, here, we use partial part $x_0^i = 1$

(b) update parameter one times iteration:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1} m(h_\theta(x^i) - y^i) x_j^i \tag{13}$$

note that here is a sum function, indicate that deal with all sample, and compare following article

(c) Pseudo code form as following:

repeat{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1} m(h_\theta(x^i) - y^i) x_j^i$$

(for $j = 0, 1$)

}

(d) advantages:

i. calculate all sample every derivation, meanwhile operate matrix and implement parallel

ii. confirm the direction due to all dataset could represent sample population, and turn to direction fo extreme value more correctly. when target function is convex function, BGD must be able to get global optimization.

(e) disadvantages:

i. when sample number m is large, it needs calculate all sample every derivation step, and the train progress will be slow.

(f) convergence curve, that derivation number is less for it.

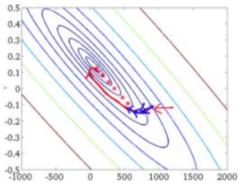

Fig 3: Bach Gradient Descent

(2) stochastic gradient descent, update parameter using only a sample every iteration. let train speed be accelerated.

(a) target function for one sample

$$J^i(\theta_0, \theta_1) = \frac{1}{2}(h_\theta(x^i) - y^i)^2 \tag{14}$$

(b) derivation for target function

$$\frac{\nabla J^i(\theta_0, \theta_1)}{\theta_j} = (h_\theta(x^i) - y^i)x_j^i \tag{15}$$

(c) parameter updated

$$\theta_j := \theta_j - \alpha(h_\theta(x^i) - y^i)x_j^i \tag{16}$$

(d) Pseudo code form as following:

Loop{

for i = 1 to m, {

$$\theta_j := \theta_j - \alpha(h_\theta(x^i) - y^i)x_j^i$$

(for $j = 0, 1$)

}

}

(e) advantages

i. because that the loss function is not belong to the all train dataset, and in every iteration, optimize the loss function of anyone train data randomly, it will accellerate the update speed more quickly.

(f) disadvantages

i. accuracy will be decline. although target function is convex function, SGD couldn't still converge linearly.

ii. maybe converge to the local optimization, due to signal sample couldn't represent the trend of all sample.

iii. be not easy to realize.

(g) explain why SGD is faster than BGD for the convergence speed.

answer: here assump that it is 30W sample, one the hand, for BGD, here need calculate 30W sample every iteration and update parameter one times. So it maybe to be iterated for many times like 10 times. on the other hand, for SGD, it only needs a sample every iteration, thus when update the parameter using 30W sample, during the period, it could converge to a suitable minimize value by SGD. what's more, at convergence, BGD have calculated for $10 \times 30W$ times, and SGD have calculated for $1 \times 30W$ times.

(h) convergence curve, the iteration times is pretty high, the sovelution space seem like be blind. as following:

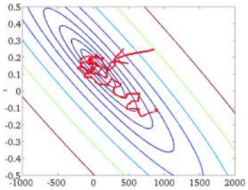

Fig 4: Stochastic Gradient Descent

(3) mini batch gradient descent, batch_size sample to update parameter every iteration. here, we assump that batch_size = 10, sample number m = 1000.

(a) Pseudo code form as following:

repeat{

for i = 1,11,21,31,...,991 {

$$\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_\theta(x^k) - y^k) x_j^k$$

(for $j = 0, 1$)

}

}

(b) advantages:

i. by matrix operation, optimize nenural network parameter isn't slower much than on one batch signal data every times.

ii. it could decline iteration times enormously that converge when using one batch. meanwhile, it could make the result gotten convergence to close the impact of gradient descent.(for example, above 30W, set batch_size = 100, need iterate for 3000 times, more less than 30W times of SGD)

iii. be able to realize parallel.

(c) disadvantages:

i. the unsuitable selection maybe bring some problem.

(d) the influence due to the selection fo batch_size

i in reasonable range, the benefit increasing batch_size:

1) the memory utilization rate increased, the parallel affect of big matrix multiply increased

2) the iteration frequency declined when run a epoch (all dataset), for equal number of data, the calculate speed increased.

3) in certain range, the more big batch_size in general,

the decline direction is preciser, the wave of train is more small.

ii the badness of increasing batch_size blindly

1) although the memory utilization rate increased, the capacity of memory couldn't support
2) although the iteration frequency declined when run a epoch (all dataset), if want to get the equal precision, the cost time also increased, and the updated speed become slowly.
3) when batch_size increase in a certain extent, the ensure decline direction don't vary.

(4) the picture show the convergence processing of three kind of gradient descent algorithm.

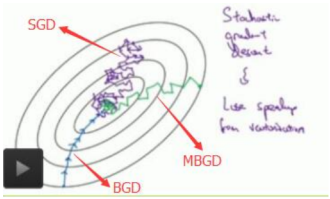

Fig 5: three kinds of Gradient Descent Algorithm

3 logistic regression

1. Logistic/sigmoid function

$$\begin{aligned} h_\theta(x) &= g(\theta^T x) \\ &= \frac{1}{1+e^{-\theta^T x}} \end{aligned} \tag{17}$$

$$\begin{aligned} g'(x) &= (\frac{1}{1+e^{-x}})' \\ &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}} \cdot (1 - \frac{1}{1+e-x}) \\ &= g(x) \cdot (1 - g(x)) \end{aligned} \tag{18}$$

(19)

2. parameter estimate of Logistic regerssion

   (a) assump that :

$$P(y=1|x;\theta) = h_\theta(x)$$

$$p(y=0|x;\theta) = 1 - h_\theta(x)$$

   (b) so likelihood term :

$$p(y|x;\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y}$$

   (c) the likelihood function:

$$\begin{aligned} L(\theta) &= p(\vec{y}|X;\theta) \\ &= \prod_{i=1}^{m} p(y^i|x^i;\theta) \\ &= \prod_{i=1}^{m} (h_\theta(x^i))^{y^i} (1-h_\theta(x^i))^{1-y^i} \end{aligned} \tag{20}$$

(d) log likelihood function

$$\begin{aligned} \ell &= \log L(\theta) \\ &= \sum_{i=1}^{m} y^i \log h(x^i) + (1-y^i) \log\left(1 - h(x^i)\right) \end{aligned} \tag{21}$$

$$\begin{aligned} \frac{\partial}{\partial \theta_j}\ell(\theta) &= (y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)})\frac{\partial}{\partial \theta_j} g(\theta^T x) \\ &= (\frac{y}{g(\theta^T x)} - \frac{1-y}{1-g(\theta^T x)}) g(\theta^T x)(1-g(\theta^T x))\frac{\partial}{\partial \theta_j}\theta^T x \\ &= (y(1-g(\theta^T x)) - (1-y)g(\theta^T x))x_j \\ &= (y - h_\theta(x))x_j \end{aligned} \tag{22}$$

3. iteration of parameter

(a) study rule of logistic regerssion parameter

$$\theta_j := \theta_j + \alpha(y^i - h_\theta(x^i))x_j^i \tag{23}$$

(b) retult compared with the linear regression they have same format.

i. repeat{

$$\theta_j := \theta_j + \alpha\frac{1}{m}\sum_{i=1} m(y^i - h_\theta(x^i))x_j^i$$

(for $j = 0, 1$)

}

ii. Loop{

for i = 1 to m, {

$$\theta_j := \theta_j + \alpha(y^i - h_\theta(x^i))x_j^i$$

(for $j = 0, 1$)

}

}

4 log linear model

1. the odds fo event, indicate that the ratio between odds happen and not happen probability.

2. log odds logit function

$$P(y=1|x;\theta) = h_\theta(x) \tag{24}$$

$$P(y=0|x;\theta) = 1 - h_\theta(x) \tag{25}$$

$$\log it(p) = \log \frac{p}{1-p} = \log \frac{h_\theta(x)}{1-h_\theta(x)} = \log \left(\frac{\frac{1}{1+e^{-\theta^T x}}}{\frac{e^{-\theta^T x}}{1+e-\theta^T x}} \right) = \theta^T x \tag{26}$$

3. loss function of logistic regression $y_i \in \{0,1\}$

$$y_i \in \{0,1\} \tag{27}$$

$$\hat{y_i} = \begin{cases} p_i & y_i = 1 \\ 1-p_i & y_i = 0 \end{cases} \tag{28}$$

$$L(\theta) = \prod_{i=1}^{m} p_i^{y_i}(1-p_i)^{1-y_i} \tag{29}$$

$$\Rightarrow \ell(\theta) = \sum_{i=1}^{m} \ln \left[p_i^{y_i}(1-p_i)^{1-y_i} \right] \tag{30}$$

$$\overset{p_i=\frac{1}{1+e^{-f_i}}}{\longrightarrow} \ell\theta = \sum_{i=1}^{m} \ln \left[(\frac{1}{1+e^{-f_i}})^{y_i} (\frac{1}{1+e^{f_i}})^{1-y_i} \right] \tag{31}$$

$$\begin{aligned} \therefore loss(y_i, \hat{y_i}) &= -l(\theta) \\ &= \sum_{i=1}^{m} \left[y_i \ln(1+e^{-f_i}) + (1-y_i)\ln(1+e^{f_i}) \right] \end{aligned} \tag{32}$$

4. loss function of logistic regression $y_i \in \{-1, 1\}$

$$y_i \in \{-1, 1\} \tag{33}$$

$$\hat{y_i} = \begin{cases} p_i & y_i = 1 \\ 1 - p_i & y_i = -1 \end{cases} \tag{34}$$

$$L(\theta) = \prod_{i=1}^{m} p_i^{(y_i+1)/2}(1-p_i)^{-(y_i-1)/2} \tag{35}$$

$$\Rightarrow \ell(\theta) = \sum_{i=1}^{m} \ln\left[p_i^{(y_i+1)/2}(1-p_i)^{-(y_i-1)/2}\right] \tag{36}$$

$$\xrightarrow{p_i=\frac{1}{1+e^{-f_i}}} \ell\theta = \sum_{i=1}^{m} \ln\left[(\frac{1}{1+e^{-f_i}})^{(y_i+1)/2}(\frac{1}{1+e^{f_i}})^{-(y_i-1)/2}\right] \tag{37}$$

$$\begin{aligned} \therefore loss(y_i, \hat{y_i}) &= -l(\theta) \\ &= \sum_{i=1}^{m}\left[\frac{1}{2}(y_i+1)\ln(1+e^{-f_i}) - \frac{1}{2}(y_i-1)\ln(1+e^{f_i})\right] \\ &= \begin{cases} \sum_{i=1}^{m}\left[\ln(1+e^{-f_i})\right] & y_i = 1 \\ \sum_{i=1}^{m}\left[\ln(1+e^{f_i})\right] & y_i = -1 \end{cases} \end{aligned} \tag{38}$$

$$\Rightarrow loss(y_i, \hat{y_i}) = \sum_{i=1}^{m}\left[\ln(1+e^{-y_i \cdot f_i})\right] \tag{39}$$

5 softmax regression

1. K classfication, the parameter of kth class is $\vec{\theta_k}$, form two dimension matrix $\theta_{k\times n}$

2. probability:

$$p(c = k|x;\theta) = \frac{\exp(\theta_k^T x)}{\sum_{l=1}^{K}\exp(\theta_l^T x)}, \quad k = 1, 2, ..., K \tag{40}$$

3. likelihood function:

$$L(\theta) = \prod_{i=1}^{m} \prod_{k=1}^{K} p(c = k|x^i; \theta)^{y_k^i} = \prod_{i=1}^{m} \prod_{k=1}^{K} \frac{\exp(\theta_k^T x^i)^{y_k^i}}{\sum_{l=1}^{K} \exp(\theta_l^T x^i)} \tag{41}$$

4. log likelihood:

$$J_m(\theta) = \ln L(\theta) = \sum_{i=1}^{m} \sum_{k=1}^{K} \left(y_k^i \cdot \theta_k^T x^i - \ln \sum_{l=1}^{K} \exp(\theta_l^T x^i) \right) \tag{42}$$

5. stochastic gradient:

$$J(\theta) = \sum_{k=1}^{K} y_k \cdot \left(\theta_k^T x - \ln \sum_{l=1}^{K} \exp(\theta_l^T x) \right) \tag{43}$$

$$\frac{\partial J(\theta)}{\partial \theta_k} = (y_k - p(y_k|x; \theta)) \cdot x \tag{44}$$