Lecture 11

Thursday, October 5, 2023 10:02 AM

Linear regression

$$\hat{y} = \vec{\omega}^T \vec{x}$$

$$\underbrace{= \omega_0 + \omega_1 x_1 + \cdots + \omega_d x_d}_{\text{hyperplane}}$$

$$\underbrace{\vec{\omega} = \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_d \end{pmatrix} \quad \vec{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}}_{\text{augmented space}}$$

$$\mathcal{L} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

SSE
sum of squared errors

$P(y=1 \mid \vec{x})$ vs $P(y=0 \mid \vec{x}) = 1 - P(y=1 \mid \vec{x})$

binary logistic regression

y is categorical | classification

$y_i \in \{0, 1\}$ binary class

$$P(y=1 \mid \vec{x}) \equiv \theta(\underline{\vec{\omega}^T \vec{x}}) \quad \in \mathbb{R}^{d+1}$$

(augmented space)

$\theta \equiv$ Sigmoid

$$\theta(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z} \longleftarrow \quad 1 - \theta(z) = \theta(-z)$$

$$\boxed{\theta(\vec{\omega}^T \vec{x}) = \frac{e^{\vec{\omega}^T \vec{x}}}{1+e^{\vec{\omega}^T \vec{x}}}}$$

$$P(y=0 \mid \vec{x}) = 1 - \theta(\vec{\omega}^T \vec{x}) = \theta(-\vec{\omega}^T \vec{x})$$

$$\theta(z) = \frac{e^z}{1+e^z} \quad , \text{ then } \underbrace{\theta'(z)}_{\text{derivative}} = \frac{\partial \theta(z)}{\partial z} = \theta(z) \cdot (1-\theta(z))$$

$$= \theta(z) \cdot \theta(-z)$$

$$P(y \mid \vec{x}) = \begin{cases} \theta(z) & \text{if } y=1 \\ 1-\theta(z) & \text{if } y=0 \end{cases} \qquad z = \vec{\omega}^T \vec{x}$$

$$D = \{ \vec{x}_i , y_i \}_{i=1}^{n} \qquad \text{dataset}$$

$$\boxed{P(y \mid \vec{x}) = \theta(z)^y (1-\theta(z))^{1-y}} \qquad \text{model}$$

alternative expression

n equations

$$\boxed{P(y_i \mid \vec{x}_i) = \theta(\vec{\omega}^T x_i)^{y_i} (1-\theta(\vec{\omega}^T x_i))^{1-y_i}} \quad \forall i = 1 \dots n$$

$\vec{\omega} \in \mathbb{R}^{d+1}$ is the unknown

Q: how to find $\vec{\omega}$? what's the objective?

Objective : Maximize the likelihood $\equiv$ Maximum likelihood Estimation (MLE)

$$\underbrace{L(\vec{\omega})}_{\text{likelihood}} = P(D \mid \vec{\omega})$$

$$\max_{\vec{\omega}} \quad P(D \mid \vec{\omega})$$

The $\vec{\omega}$ that gives the highest prob to the observed sample is the optimal one!

$$\max_{\vec{\omega}} L(\vec{\omega}) \equiv \underbrace{P(D \mid \vec{\omega})}_{\text{joint prob over the } n \text{ points}}$$

$$= P\left(\{\vec{x}_i, y_i\}_{i=1}^{n} \mid \vec{\omega} \right)$$

assume all points are independent

$$= \prod_{i=1}^{n} P(y_i \mid \vec{x}_i, \vec{\omega})$$

product of all prob (of each point)

$$L(\vec{\omega}) = \prod_{i=1}^{n} \sigma(\vec{\omega}^T \vec{x}_i)^{y_i} \left(1 - \sigma(\vec{\omega}^T \vec{x}_i)\right)^{1-y_i}$$

$$\frac{\partial L(\vec{\omega})}{\partial \vec{\omega}} \equiv \nabla_{\vec{\omega}}$$

$$\max \log L(\vec{\omega}) = \sum^{n} \log \left(\sigma(z_i)^{y_i} \, \sigma(-z_i)^{1-y_i} \right), \; z_i = \omega^T x_i$$

$$\max_{\vec{\omega}} \log L(\vec{\omega}) = \sum_{i=1}^{n} \log \left(\underline{\theta(z_i)^{y_i} \; \theta(-z_i)^{1-y_i}} \right) ; \quad z_i = w^T x_i$$

$$= \sum_{i=1}^{n} \log \left(\theta(z_i)^{y_i}\right) + \log\left(\theta(-z_i)^{\underline{1-y_i}}\right)$$

$$\boxed{\max_{\vec{\omega}} \log L(\vec{\omega}) = \sum_{i=1}^{n} y_i \log \theta(z_i) + (1-y_i) \log\left(1-\theta(z_i)\right)}$$

Log-Likelihood (LL)

Convert into minimization objective (ie. turn into a loss / error)

NLL $\equiv$ binary $\underline{\text{cross entropy}}$

negative log Likelihood

$$\min_{\vec{\omega}} BCE(\vec{\omega}) \equiv \min_{\vec{\omega}} NLL(\vec{\omega}) \equiv -\log L(\vec{\omega})$$

$$\max_{\vec{\omega}} \log L(\vec{\omega}) = \max_{\vec{\omega}} \sum_{i=1}^{n} \underbrace{y_i \underline{\log \theta(z_i)}}_{} + (1-y_i) \log(1-\theta(z_i))$$

$$\nabla_{\vec{\omega}} = \frac{\partial \log L(\vec{\omega})}{\partial \vec{\omega}} =$$

$$\underbrace{y_i \log\left(\text{sigmoid}(z_i)\right)}_{}$$

$$\nabla_{\vec{\omega}} = \sum_{i=1}^{n} \left(y_i - \theta(\vec{\omega}^T \vec{x}_i)\right) \vec{x}_i$$

$$\frac{\partial}{\partial \vec{\omega}} \quad y_i \log\left(\text{sigmoid}\left(\boxed{\vec{\omega}^T x_i}\right)\right)$$

dot

$$\nabla_{\vec\omega} = \sum_{i=1} \big(\underbrace{y_i}_{\text{true class}} - \underbrace{\theta(\omega x_i)}_{\text{predicted prob of } y_i=1}\big) x_i$$

true class

predicted prob of $y_i = 1$

$$\frac{\partial}{\partial \vec\omega} \qquad \underbrace{\qquad}_{\text{dot}}$$

$$\underbrace{\frac{1}{\cancel{\theta(\vec\omega^T x_i)}}}_{\text{derivative of log}} \cdot \underbrace{\cancel{\theta(\vec\omega^T x_i)} \cdot (1-\theta(\omega^T x_i))}_{\text{derivative of sigmoid}} \; \vec{x_i}$$

derivative of log

derivative of sigmoid

$$\frac{\partial \log L(\vec\omega)}{\partial \vec\omega} = 0$$

$$\sum \big(y_i - \theta(\vec\omega^T x_i)\big) \cdot x_i = 0$$

$$\boxed{\sum \theta(\vec\omega^T x_i) x_i = \sum y_i x_i}$$

No closed form solution!

$$\boxed{\nabla_{\vec\omega} = \sum_{i=1}^{n} \underbrace{\big(y_i - \theta(\vec\omega^T x_i)\big)}_{\text{Scalar}} \cdot \underbrace{\vec{x_i}}_{\text{vector}}}$$

Scalar

vector

$$\nabla_{\vec\omega} \in \mathbb{R}^{d+1}$$

$$\vec\omega^{(t)} = \vec\omega^{(t-1)} + \eta \cdot \nabla_{\vec\omega}$$

η: step size (eta)

batch gradient ascent!

Stochastic Gradient Ascent

based on a single point

Practical

based on a single point

$$\boxed{\nabla_{\vec{\omega}}(x_i) = \left(y_i - \theta(\vec{\omega}^T x_i)\right)\vec{x}_i}$$

Practical

pick a mini-batch B of points

$$\vec{\omega}^{(t)} = \vec{\omega}^{(t-1)} + \eta \cdot \nabla_{\vec{\omega}}(x_i)$$

$\vec{\omega}$: training

testing / inference ?

given an unknown $\vec{x}_{n+1}$

$\underbrace{\text{unknown}}_{y_{n+1}\,?}$

$$\boxed{f(\vec{x}_{n+1}) = \begin{cases} \hat{y} = 1 & \text{if } \theta(\vec{\omega}^T x_{n+1}) \geq 0.5 \\ \hat{y} = 0 & \text{otherwise} \end{cases}}$$

Multiclass logistic Regression

$\{\vec{x}_i, y_i\}_{i=1}^{n}$

$y_i \in \underbrace{\{c_1, c_2, \ldots, c_k\}}_{k \text{ - symbolic classes}}$

given $y_i, \vec{x}_i$

predict k probabilities

$$P(y_i = c_1 \mid \vec{x}_i) = \pi_1(\vec{x}_i)$$

$$P(y_i = c_2 \mid \vec{x}_i) = \pi_2(\vec{x}_i)$$

$$P(y_i = c_1 \mid x_i) = \pi_1(x_i)$$

$$P(y_i = c_2 \mid \vec{x}_i) = \pi_2(\vec{x}_i)$$

$$\vdots \qquad\qquad \vdots$$

$$P(y_i = c_k \mid \vec{x}_i) = \pi_k(\vec{x}_i)$$

$$\boxed{\sum_{j=1}^{k} \pi_j(\vec{x}_i) = 1}$$

we need k different weight vectors

$$\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_k \in \mathbb{R}^{d+1}$$

$$\pi_1(\vec{x}_i) = f(\vec{w}_1^T x_i)$$

$$\pi_2(\vec{x}_i) = f(\vec{w}_2^T \vec{x}_i)$$

$$\vdots$$

$$\pi_k(\vec{x}_i) = f(\vec{w}_k^T x_i)$$

~~$f =$ sigmoid~~

f has to be some function that sum to 1 with all values in $[0, 1]$

$f \equiv$ Softmax function

$$f(\vec{w}_j^T x_i) = \frac{e^{\vec{w}_j^T \vec{x}_i}}{\underbrace{\sum_{a=1}^{k} e^{\vec{w}_a^T x_i}}_{\text{over all classes}}}$$

j-th class

$$= \text{softmax}(\vec{w}_j^T \vec{x}_i)$$

$$= \text{softmax}(\vec{w}_j \cdot \vec{x}_i)$$

$$= \text{softmax}\left(\vec{w}_j^T x_i \,\middle|\, \underline{\vec{w}_1^T x_i, \vec{w}_2^T x_i, \ldots, \vec{w}_k^T x_i}\right)$$

Sigmoid

$$\theta(\vec{w}^T \vec{x}_i) = \frac{e^{\vec{w}^T \vec{x}_i}}{1+e^{\vec{w}^T \vec{x}_i}} = \frac{e^{\vec{w}^T x_i}}{e^0 + e^{\vec{w}^T x_i}}$$

Unknown parameters

$\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_k$ are all unknown

$$W = \begin{bmatrix} | & | & & | \\ \vec{w}_1 & \vec{w}_2 & \cdots & \vec{w}_k \\ | & | & & | \end{bmatrix} \quad (d+1) \times K$$

weight matrix

$$\text{MLE}: \max_W \log L(W) = \sum_{i=1}^{n} \log \underline{P(y_i \mid W)}$$

Under independence assumption

$$P(y_i | W) = \begin{cases} \pi_1(\vec{w}_1^T \vec{x}_i) & \text{if } y_i = C_1 \\ \pi_2(\vec{w}_2^T \vec{x}_i) & \text{if } y_i = C_2 \\ \vdots & \\ \pi_k(\vec{w}_k^T x_i) & \text{if } y_i = C_k \end{cases}$$

e.g.

$K = 4$

	C_1	C_2	C_3	C_4
$\vec{e}_1 =$	1	0	0	0
$\vec{e}_2 =$	0	1	0	0
$\vec{e}_3 =$	0	0	1	0

One_hot encoding of each class

$\vec{e}_2 = [\,0 \;\; 1 \;\; 0 \;\; 0\,]$

$\vec{e}_3 = [\,0 \;\; 0 \;\; 1 \;\; 0\,]$

$\vec{e}_4 = [\,0 \;\; 0 \;\; 0 \;\; 1\,]$

One_hot encoding of each class

$$(\vec{x}_i, y_i) \longrightarrow (\vec{x}_i, \vec{y}_i)$$

$$c_1 \longrightarrow \vec{e}_1$$

$$c_2 \qquad\qquad \vec{e}_2$$

$$\vec{y}_i \in \{e_1, e_2, \dots, e_k\}$$

encode the k classe into one_hot vectors for the true class.

x_1	x_2	y
1	2	$\boxed{c_3}$

$\Rightarrow$ $\vec{x}_i = (1,2)$ $\quad \vec{y}_i = \vec{e}_3 = (0010)^T$

$\underline{\vec{x}_i \in \mathbb{R}^{d+1}}$ after augment $\qquad \vec{y}_i \in \underline{\mathbb{R}^k}$ One_hot

$$P(\vec{y}_i \mid x_i) = \Big[\, \pi_1(\vec{w}_1^T x_i)^{y_{i1}=0} \cdot \pi_2(\vec{w}_2^T \vec{x}_i)^{y_{i2}=0} \cdot \pi_3(\vec{w}_3^T x_i)^{y_{i3}=1} \cdots \pi_k(\vec{w}_k^T \vec{x}_i)^{y_{ik}=0} \,\Big]$$

e.g. $\vec{y}_i = \vec{e}_3 = (0010) = (y_{i1}, y_{i2}, y_{i3}, y_{i4})$ with $y_{i1}=0,\ y_{i2}=0,\ y_{i3}=1,\ y_{i4}=0$

$k=4$

$$\boxed{\max_{W} \log L(W) = \sum_{i=1}^{n} y_{i1} \log \pi_1(w_1^T x_i) + y_{i2} \log \pi_2(w_2^T x_i) \cdots}$$

$$\max_{W} \log L(W) = \sum_{i=1} y_{i1} \log \pi_1(w_1^T x_i) + y_{i2} \log \pi_2(w_2^T x_i) \cdots + y_{ik} \log \pi_k(w_2^T x_i)$$

Cross-entropy = NLL

CE

Solve for $W = \begin{bmatrix} \vec{w}_1 & \vec{w}_2 & \cdots & \vec{w}_k \end{bmatrix}$ k different weight vectors

$$\frac{\partial \log L(W)}{\partial \vec{w}_1}, \quad \frac{\partial \log L(W)}{\partial \vec{w}_2}, \quad \cdots, \quad \frac{\partial \log L(W)}{\partial \vec{w}_k}$$

$$\nabla_{w_1} \qquad \nabla_{w_2} \qquad \nabla_{w_k}$$

for the a-th class

$$\nabla_{w_a} = \sum_{i=1}^{n} \left(\underbrace{y_{ia}}_{\text{true prob}} - \underbrace{\pi_a(\vec{w}_a^T \vec{x}_i)}_{\text{pred prob}} \right) \cdot \vec{x}_i$$

Stochastic version

$$\nabla_{w_a}(\vec{x}_i) = \left(y_{ia} - \pi_a(w_a^T x_i) \right) x_i$$

Gradient Ascent

$$w_a^{(0)} = \text{random } (d+1) \text{ dim vector} \quad \forall\, a = 1 \ldots k$$

Uniform $[0,1)$ or random normal $(\,, \vec{0}, \sigma = 0.1)$

Uniform $[0, 1]$ or random normal $(\, , \bar{0}, \sigma = 0.1)$

for $i = 1 \cdots n$ in random order ← stochastic

epoch

for $j = 1 \cdots k$ ← all classes

$\pi_j(\vec{\omega}^T x_i)$ ← softmax values

$$\nabla_{\omega_j}(x_i) = \left(y_{ij} - \pi_j(\vec{\omega}_j^T \vec{x}_i)\right) \cdot x_i$$

$$\boxed{\omega_j^{(t+1)} = \omega_j + \eta \cdot \nabla_{\omega_j}(\vec{x}_i)}$$

Check for Convergence

$$\delta = \sum_{j=1}^{k} \left\| \vec{\omega}_j^{t+1} - \vec{\omega}_j^{t} \right\|$$

Stop if $\delta < \varepsilon$

← threshold, e.g. $\varepsilon = 0.0001$

Logistic model

↳ Computes log-odd ratios

e.g. binary case

$$\text{log odds ratio} \equiv \log\left(\frac{P(y_i = 1 \mid x_i)}{P(y_i = 0 \mid x_i)}\right) = \vec{\omega}^T \vec{x}_i$$

or

$(\vec{\omega} - \vec{0})^T x_i$

think of this as weight vector

$$\underbrace{\qquad\qquad}_{\text{Odds ratio}}$$

think of this as weight vector for class $y=0$

also true for k-class situation (softmax)

fix k-th class

$$\text{log odds ratio}_1 \equiv \log\left(\frac{P(y_i = c_1 \mid x_i)}{P(y_i = c_k \mid x_i)}\right) = \cancel{\vec{w}_1^T \vec{x}_i}$$

(lor)

mistake

Should be $(\vec{w}_1 - \vec{w}_k)^T \vec{x}_i$

$$\text{lor}_2 = \log\left(\frac{P(y_i = c_2 \mid x_i)}{P(y_i = c_k \mid x_i)}\right) = \cancel{\vec{w}_2^T \vec{x}_i}$$

mistake

should be $(\vec{w}_2 - w_k)^T x_i$

i.e. log odds ratio is related to the difference of the weight vectors