Logistic Regression

2021年12月31日 星期五　　下午8:38

Binary linear classification

- classification: given a D-dimensional input $x \in \mathbb{R}^D$, predict a discrete-valued target
- binary: predict a binary target $t \in \{0, 1\}$ or sometimes $t \in \{-1, 1\}$ for computational convenience
- linear: model y is a linear function of x, followed by a threshold r:

$$z = \vec{w}^T \vec{x} + b$$

$$y = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$ ← r

Loss function

- defined to optimize the model

1st attempt: 0-1 loss

$$\mathcal{L}(y, t) = \begin{cases} 0 & \text{if } y = t \\ 1 & \text{if } y \neq t \end{cases}$$

problem: $\frac{\partial \mathcal{L}}{\partial z} = 0$ where $\mathcal{L}$ is defined, can't be used for optimization

2nd attempt: linear regression

$$\mathcal{L}(z, t) = \frac{1}{2}(z - t)^2$$

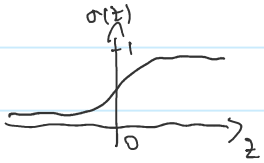
problem: even when classified correctly, loss can be very big

e.g., $t = 1$, $y = 101$, $\mathcal{L} = 5050$

3rd attempt: activation function

first, transform z to be within $[0, 1]$ using the sigmoid function

$$y = \sigma(z) = \frac{1}{1 + e^{-z}}$$



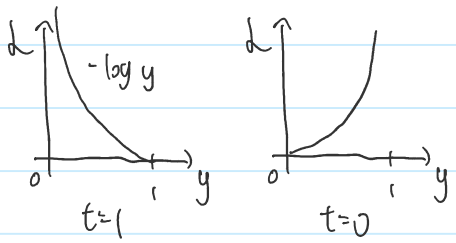
$$\mathcal{L} = \frac{1}{2}(y - t)^2$$

problem: since $(y - t) \in (0, 1)$, $\frac{\partial \mathcal{L}}{\partial y} \in (0, 1)$. When z is small/large, $\frac{\partial y}{\partial z} \approx 0$, hence $\frac{\partial \mathcal{L}}{\partial w_i} \approx 0$

3rd attempt plus: cross entropy loss

$$\mathcal{L} = \begin{cases} -\log y & \text{if } t = 1 \\ -\log(1 - y) & \text{if } t = 0 \end{cases}$$

$$= -t \log y - (1 - t) \log(1 - y)$$



Logistic regression

$$z = \vec{w}^T \vec{x}$$

$$y = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$L(y,t) = -t \log y - (1-t)\log(1-y)$$

$$J = \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, t^{(i)})$$

J: cost function; N: training examples

Gradient

$$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w_j}$$

we have:

$$\frac{\partial L}{\partial y} = \frac{\partial}{\partial y}[-t\log(y) - (1-t)\log(1-y)]$$

$$= -t \cdot \frac{1}{y} - (1-t)\cdot\frac{1}{1-y}\cdot(-1)$$

$$= -\frac{t}{y} + \frac{1-t}{1-y}$$

$$\frac{\partial y}{\partial z} = \frac{\partial}{\partial z}(1+e^{-z})^{-1}$$

$$= -1\cdot(1+e^{-z})^{-2}\cdot e^{-z}\cdot(-1)$$

since $y = (1+e^{-z})^{-1}$, we get $e^{-z} = \frac{1}{y} - 1$

the above equals to:

$$(\frac{1}{y})^{-2}\cdot(\frac{1}{y}-1)$$

$$= y^2(\frac{1}{y}-1)$$

$$= y - y^2$$

$$\frac{\partial z}{\partial w_j} = \frac{\partial}{\partial w_j}(w_1x_1 + w_2x_2 + \cdots + w_jx_j + \cdots + w_Nx_N)$$

$$= x_j$$

applying the chain rule, we get:

$$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w_j}$$

$$= (-\frac{t}{y} + \frac{1-t}{y})(y-y^2)x_j$$

$$= (y-t)x_j$$

hence:

$$\nabla_w L = (y-t)\vec{x}$$

Gradient descent

$$w_j \leftarrow w_j - \alpha\frac{\partial J}{\partial w_j}$$

$$= w_j - \frac{\alpha}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})x_j^{(i)}$$

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j}$$

$$= w_j - \frac{\alpha}{N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)}) x_j^{(i)}$$

Softmax Regression

2022年1月1日 星期六　上午12:32

classification with K classes

D input dimentions (features)

K output dimentions (classes)

K x D weights dimentions

K bias dimentions

$$K\left[\begin{matrix} & D \\ & W \end{matrix}\right] D\left[x\right] + K\left[\begin{matrix} 1 \\ b \end{matrix}\right] \to K\left[z\right]$$

$$\vec{z} = W\vec{x} + \vec{b}$$

Eliminating the bias with dummy feature $x_0 = 1$

$$\vec{z} = W\vec{x} \qquad K\left[\begin{matrix} 1 & D+1 \\ b & W \end{matrix}\right] \begin{matrix} 0 \\ \downarrow \end{matrix} \left[\begin{matrix} 1 \\ x \end{matrix}\right] \to K\left[z\right]$$

Turn linear prediction to probabilities with Softmax activation

$y_k = \text{Softmax}(z_1, \cdots, z_K)_k = \frac{e^{z_k}}{\sum_{k'} e^{z_{k'}}}$

property: $\sum_k y_k = 1$

Softmax - cross - entropy loss

$L(y, t) = -\vec{t}^T \log \vec{y}$

note: this is actually calculating the cross entropy of the true class, the losses of the other classes are zeroed out as $\vec{t}$ is one-hot

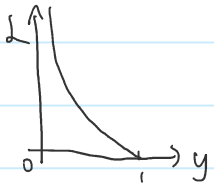
$$(-1) \cdot \underbrace{[00\cdots 1 \cdots 0]}_{t} \cdot \log\left(K\left[y\right]\right) = 1[L]$$

Softmax Regression

$\vec{z} = W\vec{x}$

$\vec{y} = \text{Softmax}(\vec{z})$

$L = -\vec{t}^T (\log \vec{y})$



Gradient

$$\frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial y_k} \cdot \frac{\partial y_k}{\partial z_k} \cdot \frac{\partial z_k}{\partial W_{ij}}$$

we have:

$\frac{\partial L}{\partial} \cdot \frac{\partial}{\partial}$ [illegible]

we have:

$$\frac{\partial L}{\partial y_k} = \frac{\partial}{\partial y_k}[-(t_0 \log y_0 + t_1 \log y_1 + \cdots + t_k \log y_k + \cdots + t_K \log y_K)]$$

$$= \frac{\partial}{\partial y_k}(-t_k \log y_k)$$

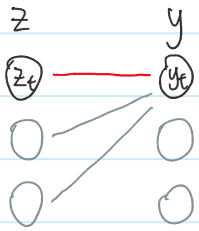
$$= -\frac{t_k}{y_k}$$

$$= \begin{cases} -\frac{1}{y_t} & \text{when } k \text{ is the true class } t \ (t_k = 1) \\ 0 & \text{when } k \text{ is a false class } (t_k = 0) \end{cases}$$

since $\frac{\partial L}{\partial y_k} = 0$ when k is a false class, we are only concerned about y_k when k is the true class t

for $\frac{\partial y_t}{\partial z_k}$:

case 1 k is the true class for z_k



$$\frac{\partial y_t}{\partial z_k} = \frac{\partial y_t}{\partial z_t}$$

$$= \frac{\partial}{\partial z_t}\left(\frac{e^{z_t}}{\sum_{k'} e^{z_{k'}}}\right)$$

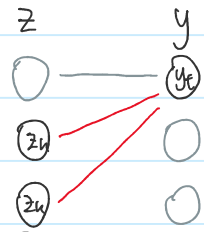
$$= \frac{e^{z_t} \cdot \sum_{k'} e^{z_{k'}} - e^{z_t} \cdot e^{z_t}}{(\sum_{k'} e^{z_{k'}})^2}$$

$$= \frac{e^{z_t}}{\sum_{k'} e^{z_{k'}}}\left(\frac{\sum_{k'} e^{z_{k'}}}{\sum_{k'} e^{z_{k'}}} - \frac{e^{z_t}}{\sum_{k'} e^{z_{k'}}}\right)$$

$$= y_t(t_t - y_t)$$

$$= y_t(t_k - y_k) \quad \text{when } k \text{ is the true class}$$

case 2 k is a false class for z_k



$$\frac{\partial y_t}{\partial z_k} = \frac{\partial}{\partial z_k}\left(\frac{e^{z_t}}{\sum_{k'} e^{z_{k'}}}\right)$$

$$= \frac{0 \cdot \sum_{k'} e^{z_{k'}} - e^{z_t} \cdot e^{z_k}}{(\sum_{k'} e^{z_{k'}})^2}$$

$$= -\frac{e^{z_t} \cdot e^{z_k}}{(\sum_{k'} e^{z_{k'}})^2}$$

$$= -\frac{e^{z_t}}{\sum_{k'} e^{z_{k'}}} \cdot \frac{e^{z_k}}{\sum_{k'} e^{z_{k'}}}$$

$$= -y_t \cdot y_k$$

$$= y_t(t_k - y_k) \quad \text{when } k \text{ is a false class}$$

Combining the two cases, we get:

$$\frac{\partial y_t}{\partial} = \quad (t \quad \quad)$$

Combining the two cases, we get:

$$\frac{\partial y_t}{\partial z_k} = y_t(t_k - y_k)$$

finally:

$$\frac{\partial z_k}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}}(w_{k,0}x_0 + w_{k,1}x_1 + \cdots + w_{k,j}x_j + \cdots + w_{k,D}x_D)$$

$$= x_j$$

applying the chain rule, we get:

$$\frac{\partial L}{\partial w_{kj}} = \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_{kj}}$$

$$= -\frac{1}{y_t} \cdot y_t(t_k - y_k) \cdot x_j$$

$$= (t_k - y_k) \cdot x_j$$

hence:

$$\nabla_{w_k} L = (\vec{t} - \vec{y}) \cdot \vec{x}^T$$

Gradient descent

$$\vec{w}_k \leftarrow \vec{w}_k - \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} (y_k^{(i)} - t_k^{(i)}) \vec{x}^{(i)}$$

Q&A

Q: Minimum possible loss?

A: min: 0 max: ∞

Q: At weight init, W can be small, resulting small s (≈ 0). What's the loss?

A: $-\log(\frac{1}{K})$

Support Vector Machine

2022年1月1日 星期六　　下午1:19

- maximizes the margin
- $t \in \{-1, 1\}$

Linear model

$\vec{y} = W\vec{x} + \vec{b}$

Hinge loss

$L = \sum_{i=1}^{N} \max\{0, 1 - t^{(i)} y^{(i)}\}$

- 0 if $y^{(i)}$ outside of the margin
- $1 - t^{(i)} y^{(i)}$ if $y^{(i)}$ inside of the margin or wrongly classified

Soft-margin SVM loss

$L = \sum_{i=1}^{N} \max\{0, 1 - t^{(i)} y^{(i)}\} + \frac{1}{2\gamma}\|W\|_2^2$

- $\max\{0, 1 - t^{(i)} y^{(i)}\}$: hinge loss
- $\|W\|_2^2$: L_2-norm, $\sum_{i=1}^{K}\sum_{j=1}^{D}(W_{j,i})^2$

Multiclass SVM loss

$L = \sum_{j \neq y_i} \max\{0, s_j - s_{y_i} + 1\}$

- $\sum_{j \neq y_i}$: all classes except the true class
- y_i: integer label
- s_j: false class score
- s_{y_i}: true class score
- $\max\{0, s_j - s_{y_i} + 1\}$:
  - 0 if true class score larger than the others by at lease 1
  - $s_j - s_{y_i}$ if false class score close to (within 1) or larger than true class score

example

Cat 3.2

Car 5.1

Frog -1.7

$L = \max\{0, 5.1 - 3.2 + 1\} + \max\{0, -1.7 - 3.2 + 1\}$

$= \max\{0, 2.9\} + \max\{0, -3.9\}$

$= 2.9 + 0$

$= 2.9$

Q&A

Q: What if sum include the correct class?

A: L increase by 1.

Q: What if we use mean instead of sum?

A: No effect.

Q: What if $L_{new} = L^2$?

A: The penalty scale is changed

Q: Minimum possible loss?

A: min: 0 max: ∞

Q: At weight init, W can be small, resulting small s (≈ 0). What's the loss?

A: N-1

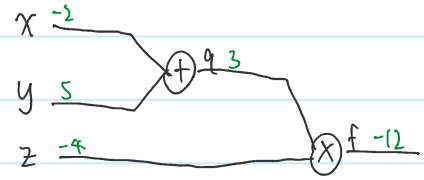
Q: What if we jiggle the score for frog a bit?

A: Does not change as it is smaller than the correct class score by more than 1

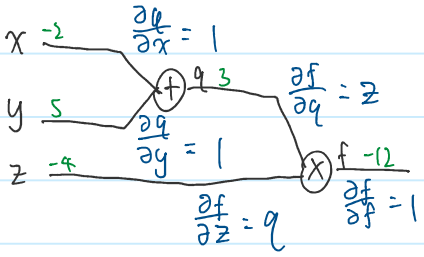
Back Propagation

2022年1月1日 星期六　　下午4:59

Example



$f = (x+y)\cdot z$



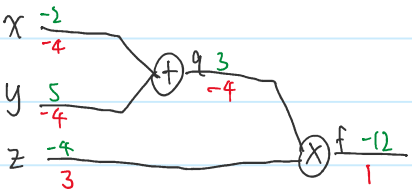
$\frac{\partial f}{\partial f} = 1$

$\frac{\partial f}{\partial q} = -4$

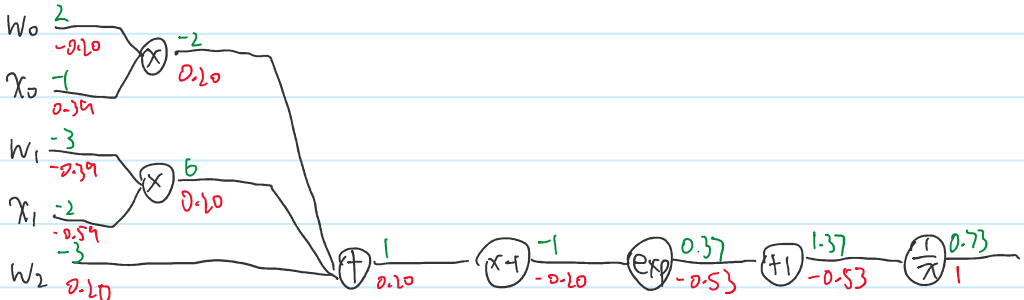
$\frac{\partial f}{\partial z} = 3$

$\frac{\partial f}{\partial x} = -4 \cdot \frac{\partial q}{\partial x} = -4$

$\frac{\partial f}{\partial y} = -4 \cdot \frac{\partial q}{\partial y} = -4$



Example



$f = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

$\frac{\partial f}{\partial f} = 1$

$\frac{\partial f}{\partial x} = 1 \cdot \frac{\partial}{\partial x}\frac{1}{x} = -1 \cdot \frac{1}{x^2} = -\frac{1}{1.37^2} = -0.53$

$\frac{\partial f}{\partial x} = -0.53 \frac{\partial}{\partial x}(x+1) = -0.53 \cdot 1 = -0.53$

$\frac{\partial f}{\partial x} = -0.53 \frac{\partial}{\partial x}(e^x) = -0.53 \cdot e^x = -0.53 \cdot e^{-1} = -0.20$

$\frac{\partial f}{\partial x} = -0.20 \frac{\partial}{\partial x}(-1 \cdot x) = 0.20 \cdot (-1) = 0.20$

$\frac{\partial f}{\partial w_2} = 0.20 \frac{\partial}{\partial w_2}(-2+b+tw_2) = 0.20 \cdot 1 = 0.20$

$\frac{\partial f}{\partial x} = 0.20 \frac{\partial}{\partial x}(x+b-3) = 0.20 \cdot 1 = 0.20$

$\frac{\partial f}{\partial w_0} = 0.20 \frac{\partial}{\partial w_0}(-1 \cdot w_0) = 0.20 \cdot (-1) = -0.20$

$\frac{\partial f}{\partial x_0} = 0.20 \frac{\partial}{\partial x_0}(2 \cdot x_0) = 0.20 \cdot 2 = 0.39$

$\frac{\partial f}{\partial x} = 0.20 \frac{\partial}{\partial x}(-2+x-3) = 0.20 \cdot 1 = 0.20$

$\frac{\partial f}{\partial w_1} = 0.20 \frac{\partial}{\partial w_1}(-2 \cdot w_1) = 0.20 \cdot (-2) = -0.39$

$\frac{\partial f}{\partial x_1} = 0.20 \frac{\partial}{\partial x_1}(-3 \cdot x_1) = 0.20 \cdot (-3) = -0.59$

example $y = (\sum_i x_i^p)^{\frac{1}{p}}$, suppose $\frac{\partial L}{\partial y} = y'$, what is $\frac{\partial L}{\partial x_i}$?

$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_i}$

$\frac{\partial y}{\partial x_i} = \frac{\partial}{\partial x_i}(\sum_i x_i^p)^{\frac{1}{p}}$ ← $x_1^p + x_2^p + \cdots + x_i^p + \cdots + x_n^p$

$= \frac{1}{p}(\sum_i x_i^p)^{\frac{1-p}{p}} \frac{\partial}{\partial x_i}(\sum_i x_i^p)$

$= \frac{1}{p}(\sum_i x_i^p)^{\frac{1-p}{p}} \frac{\partial}{\partial x_i} x_i^p$

$= \frac{1}{p}(\sum_i x_i^p)^{\frac{1-p}{p}} \cdot p\, x_i^{p-1}$

$= (\sum_i x_i^p)^{\frac{1-p}{p}} x_i^{p-1}$

therefore, $\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_i}$

$= y' (\sum_i x_i^p)^{\frac{1-p}{p}} x_i^{p-1}$

example $y = \frac{1}{\beta} \ln(\frac{1}{n}\sum_i \exp(\beta x_i))$, suppose $\frac{\partial L}{\partial y} = y'$, what is $\frac{\partial L}{\partial x_i}$?

$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_i}$

$\frac{\partial y}{\partial x_i} = \frac{\partial}{\partial x_i}(\frac{1}{\beta} \ln(\frac{\sum_i e^{\beta x_i}}{n}))$ ← $e^{\beta x_1} + \cdots + e^{\beta x_i} + \cdots + e^{\beta x_n}$

$= \frac{1}{\beta} \cdot \frac{n}{\sum_i e^{\beta x_i}} \frac{\partial}{\partial x_i}(\frac{1}{n} \cdot \sum_i e^{\beta x_i})$

$= \frac{1}{\beta} \cdot \frac{n}{\sum_i e^{\beta x_i}} \cdot \frac{1}{n} \frac{\partial}{\partial x_i} e^{\beta x_i}$

$= \frac{1}{\beta} \cdot \frac{1}{\sum_i e^{\beta x_i}} \cdot e^{\beta x_i} \frac{\partial}{\partial x_i}(\beta x_i)$

$= \frac{1}{\beta} \cdot \frac{e^{\beta x_i}}{\sum_i e^{\beta x_i}} \cdot \beta$

$= \frac{e^{\beta x_i}}{\sum_i e^{\beta x_i}}$

therefore, $\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x_i}$

$= y' \frac{e^{\beta x_i}}{\sum_i e^{\beta x_i}}$

example $f(\vec{x}, W) = \|W\vec{x}\|^2 = \sum_{i=1}^{n} (W\vec{x})_i^2$, find $\nabla_W f$ and $\nabla_{\vec{x}} f$

$n\left[\begin{array}{c} \overset{m}{} \\ W \end{array}\right] m\left[\begin{array}{c} 1 \\ \vec{x} \end{array}\right] \rightarrow n\left[\begin{array}{c} 1 \\ \vec{q} \end{array}\right] \rightarrow q_1^2 + q_2^2 + \cdots + q_n^2$

$\begin{cases} f(q) = \|q\|^2 = q_1^2 + q_2^2 + \cdots + q_n^2 \\ \vec{q} = W \cdot \vec{x} \end{cases}$

$\frac{\partial f}{\partial W_{ij}} = \frac{\partial f}{\partial q_i} \cdot \frac{\partial q_i}{\partial W_{ij}}$

$\frac{\partial f}{\partial q_i} = \frac{\partial}{\partial q_i}(q_1^2 + q_2^2 + \cdots + q_i^2 + \cdots + q_n^2)$

$= 2q_i$

therefore, $\nabla_{\vec{x}} f = 2 \cdot n\left[\begin{array}{c} 1 \\ \vec{q} \end{array}\right] \rightarrow n\left[\begin{array}{c} 1 \\ \frac{\partial f}{\partial x_j} \end{array}\right]$

$= 2q_i$

therefore, $\nabla_{\vec{q}} f = 2q$ $\quad 2 \cdot n\left[\vec{q}\right]^1 \rightarrow n\left[\nabla_{\vec{q}} f\right]^1$

$\frac{\partial q_i}{\partial w_{i,j}} = \frac{\partial}{\partial w_{i,j}} (w_{i,1}x_1 + w_{i,2}x_2 + \cdots + w_{i,j}x_j + \cdots + w_{i,m}x_m)$

$= x_j$

hence $\frac{\partial f}{\partial w_{i,j}} = \frac{\partial f}{\partial q_i} \cdot \frac{\partial q_i}{\partial w_{i,j}} = 2q_i x_j$ $\quad 2 \cdot n\left[\vec{q}\right]^1 \times \underset{m}{\left[\vec{x}\right]} \rightarrow n\left[\nabla_w f\right]^m$

therefore, $\nabla_w f = 2\vec{q}\vec{x}^T$

$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{n} \frac{\partial f}{\partial q_i} \frac{\partial q_i}{\partial x_j}$

$\frac{\partial q_i}{\partial x_j} = \frac{\partial}{\partial x_j} (w_{i,1}x_1 + w_{i,2}x_2 + \cdots + w_{i,j}x_j + \cdots + w_{i,m}x_m)$

$= w_{i,j}$

hence $\frac{\partial f}{\partial x_j} = \sum_{i=1}^{n} \frac{\partial f}{\partial q_i} \frac{\partial q_i}{\partial x_j}$

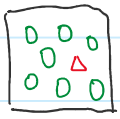
$= 2q_i w_{i,j}$

therefore, $\nabla_{\vec{x}} f = 2W^T\vec{q}$

Entropy

2021年12月31日 星期五　　下午2:52

Surprise

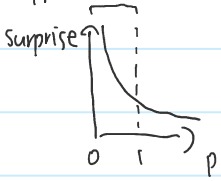


- randomly selected △ : very surprised
- randomly selected ○ : not surprised

↑ the higher the probability, the lower the surprise

Quantifying surprise

Approach 1　inverse of probability

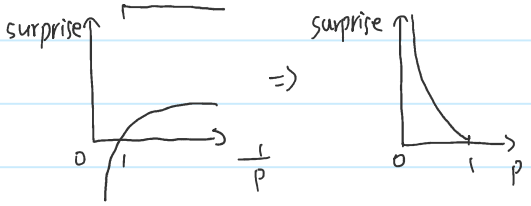


$\text{surprise} = \frac{1}{p}$

problem

want surprise = 0 when probability = 1

Approach 2　log of inverse of probability



$$\text{surprise} = \log_2\left(\frac{1}{p}\right)$$
$$= \log_2(1) - \log_2(p)$$
$$= 0 - \log_2(p)$$
$$= -\log_2(p)$$

Entropy H

- expectation of surprise
- indicator of uncertainty (the higher the entropy, the higher the uncertainty)

$$H(X) = -\sum_{x \in X} p(x) \log(p(x))$$

Example　coin toss

① coin #1
Ⓐ heads : $\frac{1}{9}$
Ⓑ tails : $\frac{8}{9}$

$$H(X) = -p(\text{heads}) \log_2(p(\text{heads})) - p(\text{tails}) \log_2(p(\text{tails}))$$
$$= -\frac{1}{9}\log_2\left(\frac{1}{9}\right) - \frac{8}{9}\log_2\left(\frac{8}{9}\right)$$
$$\approx 0.5$$

← less uncertain (since tails is more likely)
lower entropy

more uncertain (since heads and tails are evenly likely) →
higher entropy

② coin #2
Ⓐ heads : $\frac{1}{2}$
Ⓑ tails : $\frac{1}{2}$

$$H(X) = -p(\text{heads}) \log_2(p(\text{heads})) - p(\text{tails}) \log_2(p(\text{tails}))$$
$$= -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)$$
$$= 1$$

properties

- $H \geq 0$

- $H(X,Y) = H(X|Y) + H(Y) = H(Y|X) + H(X)$
- if X, Y independent, $H(Y|X) = H(Y)$ ← X informs nothing about Y
- $H(X|X) = 0$ ← X fully informs itself
- $H(Y|X) \geq H(Y)$ ← new information never decreases certainty

Entropy of a joint distribution

$$H(X,Y) = -\sum_x \sum_y p(x,y) \log_2 p(x,y)$$

Example

$$H(X,Y) = -\frac{24}{100}\log_2\left(\frac{24}{100}\right) - \frac{1}{100}\log_2\left(\frac{1}{100}\right) - \frac{25}{100}\log_2\left(\frac{25}{100}\right) - \frac{50}{100}\log\left(\frac{50}{100}\right)$$
$$\approx 1.56$$

Example weather

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

let $X = \{$Raining, Not Raining$\}$
$Y = \{$Cloudy, Not Cloudy$\}$

Specific conditional entropy

$$H(Y|X=x) = -\sum_y p(y|x) \log_2 p(y|x)$$

Example what is the probability of cloudiness Y, given that it is raining?

H(cloudiness | raining or not = raining)
$= -p(\text{cloudy}|\text{raining}) \log_2 p(\text{cloudy}|\text{raining}) - p(\text{not cloudy}|\text{raining}) \log_2 p(\text{not cloudy}|\text{raining})$
$= -\frac{24}{25}\log_2\frac{24}{25} - \frac{1}{25}\log_2\frac{1}{25}$
≈ 0.24

Conditional entropy

$$H(Y|X) = \sum_x p(x) H(Y|X=x)$$
$$= -\sum_x p(x) \sum_y p(y|x) \log_2 p(y|x)$$
$$= -\sum_x \sum_y p(x,y) \log_2 p(y|x)$$

Example what is the entropy of cloudiness, given the knowledge of whether it is raining or not?

$H(Y|X) = p(\text{raining})\, H(\text{cloudiness}|\text{raining}) + p(\text{not raining})\, H(\text{cloudiness}|\text{not raining})$
$= \frac{1}{4}\left(-\frac{24}{25}\log_2\frac{24}{25} - \frac{1}{25}\log_2\frac{1}{25}\right) + \frac{3}{4}\left(-\frac{25}{75}\log_2\frac{25}{75} - \frac{50}{75}\log_2\frac{50}{75}\right)$
≈ 0.75

Information Gain IG

- the amount of increase in the certainty of Y, when given knowledge about X

$$IG(Y|X) = H(Y) - H(Y|X)$$

↑ increase of

$H(Y)$ ← uncertainty of Y, before knowing X

$H(Y|X)$ ← uncertainty of Y, after knowing X

↑ increase of certainty

↖ uncertainty of Y, after knowing X

uncertainty of Y, before knowing X ↖

(example) how much more certain of the cloudiness am I, after having the information of whether it's raining or not?

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

$$IG(Y|X) = H(Y) - H(Y|X)$$

$$= (-p(\text{cloudy})\log_2 p(\text{cloudy}) - p(\text{not cloudy})\log_2 p(\text{not cloudy})$$

$$- p(\text{raining})\, H(\text{cloudiness} \mid \text{raining}) + p(\text{not raining})\, H(\text{cloudiness} \mid \text{not raining})$$

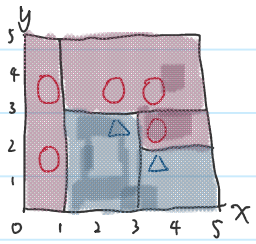
$$= \left(-\frac{49}{100}\log_2\frac{49}{100} - \frac{51}{100}\log_2\frac{51}{100}\right) - \left(\frac{1}{4}\left(-\frac{24}{25}\log_2\frac{24}{25} - \frac{1}{25}\log_2\frac{1}{25}\right) + \frac{3}{4}\left(-\frac{25}{75}\log_2\frac{25}{75} - \frac{50}{75}\log_2\frac{50}{75}\right)\right)$$

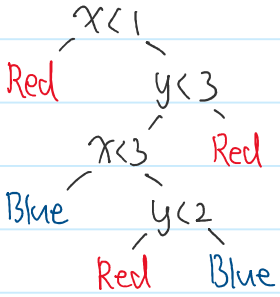
$$\approx 1.00 - 0.75$$

$$\approx 0.25$$

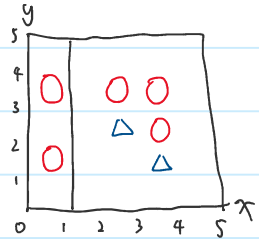
Decision Tree

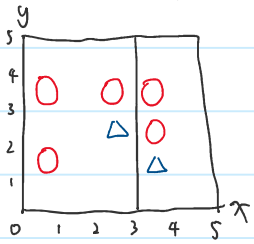
2021年12月31日 星期五　　下午5:32





Problem How to quantify the quality of a split?

① 

② 

Solution: use information gain!

$Y = \{\text{Red}, \text{Blue}\}$

$X = \{\text{Left}, \text{Right}\}$

initial entropy:

$$H(Y) = -p(\text{Red})\log_2 p(\text{Red}) - p(\text{Blue})\log_2(\text{Blue})$$
$$= -\frac{5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7}$$
$$\approx 0.86$$

Split ①:

$$H(Y|X) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(y|x)$$
$$= -p(\text{Left},\text{Red})\log_2 p(\text{Red}|\text{Left}) - p(\text{Left},\text{Blue})\log_2 p(\text{Left}|\text{Blue})$$
$$- p(\text{Right},\text{Red})\log_2 p(\text{Red}|\text{Right}) - p(\text{Right},\text{Blue})\log_2 p(\text{Right}|\text{Blue})$$
$$= -\frac{2}{7}\log_2\frac{2}{2} - \frac{0}{7}\log_2\frac{0}{2} - \frac{3}{7}\log_2\frac{3}{5} - \frac{2}{7}\log\frac{2}{5}$$
$$\approx 0.69$$

$$IG(Y,X) = H(Y) - H(Y|X) \approx 0.17$$

split ②:

$$H(Y|X) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(y|x)$$
$$= -p(\text{Left},\text{Red})\log_2 p(\text{Red}|\text{Left}) - p(\text{Left},\text{Blue})\log_2 p(\text{Left}|\text{Blue})$$
$$- p(\text{Right},\text{Red})\log_2 p(\text{Red}|\text{Right}) - p(\text{Right},\text{Blue})\log_2 p(\text{Right}|\text{Blue})$$
$$= -\frac{3}{7}\log_2\frac{3}{4} - \frac{1}{7}\log_2\frac{1}{4} - \frac{2}{7}\log_2\frac{2}{3} - \frac{1}{7}\log_2\frac{1}{3}$$

≈ 0.86

$IG(Y,X) = H(Y) - H(Y|X) \approx 0.006$

The IG of split ① is higher, it is thus a better split

Algorithm

1. pick a non-terminal node to split
2. pick a feature to split and choose where to split, then split the node
3. for each group in the split:
   - if no examples: return majority from parent
   - else if all examples in the same class: return class
   - else jump to step 1

Properties of a good tree

- not too small
- not too big
  - efficiency
  - overfitting
  - interpretability
- "Occam's Razor" find the simplest hypothesis that fits the observations
- small tree with informative nodes near the root preferred

Problems

- exponentially less data at lower levels
- overfit when tree too big
- greedy algorithm does not necessarily yield global optimum

Handling numeric features

1. sort the dataset
2. calculate IG between each split and split at the row with the highest IG

Advantages over KNN and NN

- simple to deal with discrete features, missing values, and poorly scaled data

- simple to deal with discrete features, missing values, and poorly scaled data
- Fast at test time
- More interpretable

Disadvantages

○ compared to KNN

- More hyperparameters
- Not incorporated with interesting distance measures

△ compared to NN

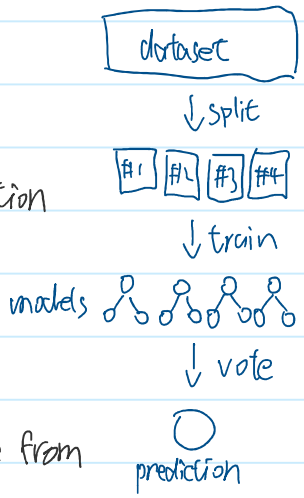
- Unable to handle features that interact in complex ways (e.g., pixels)

Random Forest

2022年1月2日 星期日　　下午3:59

Lowering the variance of a model

- train multiple different models with different datasets sampled from the sampling distribution
- aggregating the models to vote the prediction



analysis:

Bayes error - unchanged since we have no control over it

Bias - unchanged since the averaged prediction has the same expectation

$$\mathbb{E}[y] = \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} y_i\right] = \mathbb{E}[y_i]$$

Variance - reduced since independent samples are averaged

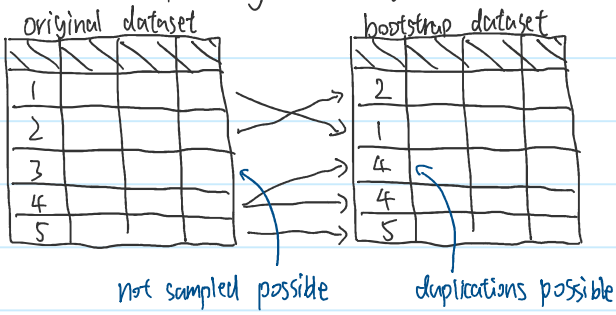
$$Var[y] = Var\left[\frac{1}{m}\sum_{i=1}^{m} y_i\right] = \frac{1}{m^2}\sum_{i=1}^{m} Var[y_i] = \frac{1}{m} Var[y_i]$$

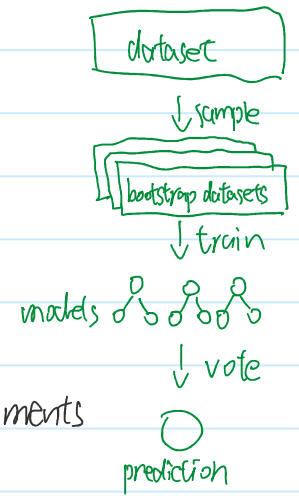
problem: the sampling distribution is often finite or expensive to sample from

(not enough data to be split into multiple datasets)

Bagging

- instead of splitting the dataset, make bootstrap datasets from the dataset





take dataset D with n examples

generate m new datasets with n examples by sampling from D with replacements

train m models from the m bootstrap datasets and average the predictions

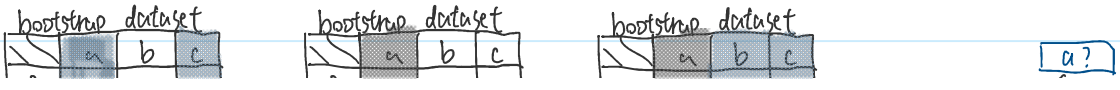
analysis:

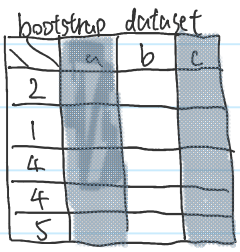
Variance - can't get $\frac{1}{m}$ variance reduction as the datasets are not independent, but still works

Ensembled hypothesis not in the original hypothesis space (e.g. $\frac{5}{8}$ cat, $\frac{3}{8}$ car, $\frac{0}{8}$ frog), solution: voting

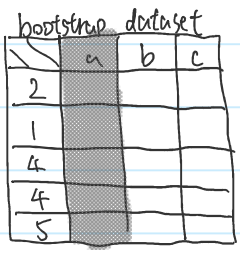
Random forest

one extra trick to decorrelate predictions: when training each tree, at each split, instead of considering all columns, only consider k randomly chosen columns that are not previously used

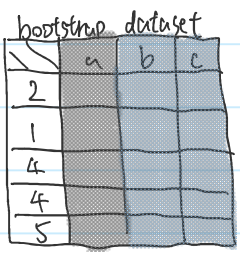




choose

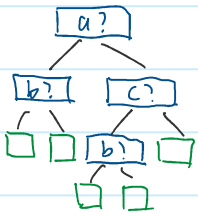


grey out chosen



choose

...



AdaBoost

2022年1月2日 星期日　　下午4:48

train classifiers sequentially, each time focusing on training examples that it previously got wrong

input: data $D_{n=1} = \{\vec{x}^{(n)}, t^{(n)}\}_{n=1}^{N}$, where

$t^{(n)} \in \{-1, 1\}$

classifier (hypothesis): $h(\vec{x}) \to \{-1, 1\}$

0-1 loss: $\mathbb{I}[h(\vec{x}^{(n)}) \neq t^{(n)}] = \frac{1}{2}(1 - h(\vec{x}^{(n)}) \cdot t^{(n)})$

0 if $h(\vec{x}^{(n)}) = t^{(n)}$

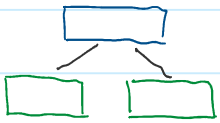
1 if $h(\vec{x}^{(n)}) \neq t^{(n)}$

Stump

tree with one node and two leaves

weak classifier

most commonly used in AdaBoost



Initialization

at the start, the weight of each sample is the same

$w^{(n)} = \frac{1}{N}$ for $n = 1, 2, \cdots, N$

Fit a classifier

in each iteration, fit a classifier $h(x)$ using weighted GINI impurity

$$h(x) \leftarrow \underset{h}{\operatorname{argmin}} \sum_{n=1}^{N} w^{(n)} \mathbb{I}\{h(x^{(n)}) \neq t^{(n)}\}$$

Calculate weighted error

$$err = \frac{\sum_{n=1}^{N} w^{(n)} \mathbb{I}\{h(x^{(n)}) \neq t^{(n)}\}}{\sum_{n=1}^{N} w^{(n)}}$$

sum of weights of misclassified samples

total weight

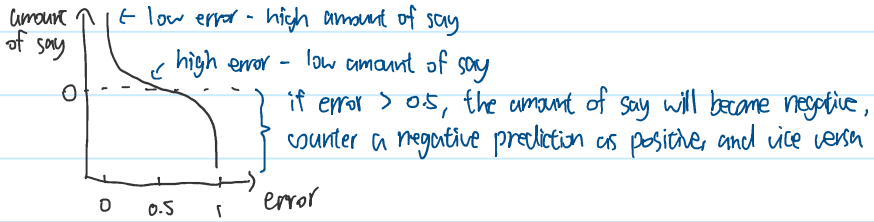
example

correctly classified?	sample weight
✓	1/8
✓	2/8
✗	3/8
✓	2/8

$$err = \frac{3/8}{1/8 + 2/8 + 3/8 + 2/8} = \frac{3}{8}$$

Determine the amount of say of the classifier (classifier coefficient)

$$\alpha = \frac{1}{2} \log\left(\frac{1 - err}{err}\right)$$

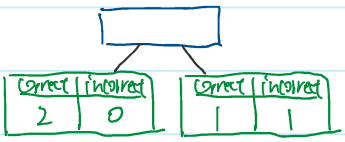


example

correctly | sample

Example

correctly classified?	sample weight
✓	1/8
✓	2/8
✗	3/8
✓	2/8



$err = \frac{3}{8}$

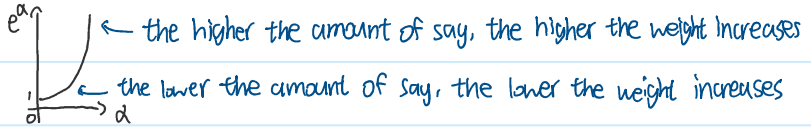
$\alpha = \frac{1}{2}\log_{10}\left(\frac{1-\frac{3}{8}}{\frac{3}{8}}\right) = 0.26$

we want to increase the weights of the samples that are incorrectly classified, and decrease the weights of the samples that are correctly classified

Update weights

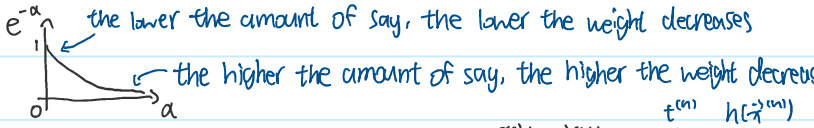
for incorrectly classified samples, we want to increase their weights

$w^{(n)} \leftarrow w^{(n)} e^{\alpha}$



for correctly classified samples, we want to decrease their weights

$w^{(n)} \leftarrow w^{(n)} e^{-\alpha}$



aggregated form: $w^{(n)} \leftarrow w^{(n)} e^{-\alpha t^{(n)} h(\vec{x}^{(n)})}$

$t^{(n)}$	$h(\vec{x}^{(n)})$	result
1	1	$-\alpha$
-1	-1	$-\alpha$
1	-1	α
-1	1	α

AdaBoost algorithm

initialize sample weights $w^{(n)} = \frac{1}{N}$ for $n = 1, 2, \ldots, N$

for $t = 1, 2, \ldots, T$:

fit a classifier $h(x) \leftarrow \underset{h}{\operatorname{argmin}} \sum_{n=1}^{N} w^{(n)} \mathbb{I}\{h(x^{(n)}) \neq t^{(n)}\}$

calculate weighted error $err = \frac{\sum_{n=1}^{N} w^{(n)} \mathbb{I}\{h(x^{(n)}) \neq t^{(n)}\}}{\sum_{n=1}^{N} w^{(n)}}$

determine amount of say $\alpha = \frac{1}{2}\log\left(\frac{1-err}{err}\right)$

update weights $w^{(n)} \leftarrow w^{(n)} e^{-\alpha t^{(n)} h(\vec{x}^{(n)})}$

return $H(\vec{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\vec{x})\right)$

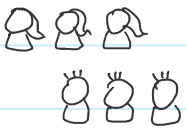
Maximum Likelihood Estimation

2022年1月5日 星期三 下午2:18

Example

Suppose we want to know the probability θ for a person in the world to have long hair, since surveying everyone in the world is not practical, we sampled N people and we have:

$x_i \begin{cases} 1 & \text{if the person has long hair} \\ 0 & \text{if the person has short hair} \end{cases}$ ← p(happen) = θ, p(not happen) = 1-θ



note that here x is a Bernoulli random variable for probability $\theta \in [0,1]$

hence:

$$p(x|\theta) = \theta^x (1-\theta)^{1-x}$$ ← p(1|θ) = θ, p(0|θ) = 0

therefore, the likelihood ← the probability of x given that θ is true is modeled as follows:

$$L = p(x_1, x_2, \cdots, x_N|\theta) = \theta^{x_1}(1-\theta)^{1-x_1}\theta^{x_2}(1-\theta)^{1-x_2}\cdots\theta^{x_N}(1-\theta)^{1-x_N} = \prod_{i=1}^{N}\theta^{x_i}(1-\theta^{1-x_i})$$

Since a good estimation of θ should assign high probability to the observed data (i.e., the likelihood should be high), our goal here is to find a $\hat{\theta}$ that maximizes the likelihood. We start by taking the derivative of L with respect to θ:

$$\frac{\partial L}{\partial \theta} = \frac{\partial}{\partial \theta}\left(\prod_{i=1}^{N}\theta^{x_i}(1-\theta^{1-x_i})\right)$$

maximum likelihood

Since when the derivative is 0 when L reaches maximum, we obtain $\hat{\theta}_{ML}$ when:

$$\frac{\partial L}{\partial \theta} = 0$$

however, since $\frac{\partial L}{\partial \theta}$ is very convoluted to work with thanks to all the multiplications, we transform the likelihood to log-likelihood:

$$\begin{aligned} l &= \log(L) \\ &= \log\left(\theta^{x_1}(1-\theta)^{1-x_1}\theta^{x_2}(1-\theta)^{1-x_2}\cdots\theta^{x_N}(1-\theta)^{1-x_N}\right) \\ &= \log\theta^{x_1} + \log(1-\theta)^{1-x_1} + \log\theta^{x_2} + \log(1-\theta)^{1-x_2} + \cdots + \log\theta^{x_N} + \log(1-\theta)^{1-x_N} \\ &= x_1\log\theta + (1-x_1)\log(1-\theta) + x_2\log\theta + (1-x_2)\log\theta + \cdots + x_N\log\theta + (1-x_N)\log(1-\theta) \\ &= \sum_{i=1}^{N} x_i\log\theta + (1-x_i)\log(1-\theta) \end{aligned}$$

now we can easily take $\frac{\partial l}{\partial \theta}$ as follows:

$$\begin{aligned} \frac{\partial l}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(x_1\log\theta + (1-x_1)\log(1-\theta) + x_2\log\theta + (1-x_2)\log\theta + \cdots + x_N\log\theta + (1-x_N)\log(1-\theta)\right) \\ &= \frac{x_1}{\theta} - \frac{1-x_1}{1-\theta} + \frac{x_2}{\theta} - \frac{1-x_2}{1-\theta} + \cdots + \frac{x_N}{\theta} - \frac{1-x_N}{1-\theta} \\ &= \frac{\sum_{i=1}^{N} x_i}{\theta} - \frac{N - \sum_{i=1}^{N} x_i}{1-\theta} \\ &= \frac{N\bar{x}}{\theta} - \frac{N(1-\bar{x})}{1-\theta} \end{aligned}$$

setting the derivative to 0 to get $\hat{\theta}_{ML}$ corresponding to the maximum likelihood, we have:

$$0 = \frac{N\bar{x}}{\hat{\theta}_{ML}} - \frac{N(1-\bar{x})}{1-\hat{\theta}_{ML}}$$

$N\bar{x}$ $N(1-\bar{x})$

$$0 = \frac{N\bar{x}}{\hat{\theta}_{ML}} - \frac{N(1-\bar{x})}{1-\hat{\theta}_{ML}}$$

$$\frac{N\bar{x}}{\hat{\theta}_{ML}} = \frac{N(1-\bar{x})}{1-\hat{\theta}_{ML}}$$

$$N(\bar{x} - \bar{x}\hat{\theta}_{ML}) = N(\hat{\theta}_{ML} - \bar{x}\hat{\theta}_{ML})$$

$$\bar{x} - \bar{x}\hat{\theta}_{ML} = \hat{\theta}_{ML} - \bar{x}\hat{\theta}_{ML}$$

$$\hat{\theta}_{ML} = \bar{x} = \frac{\#\text{long hair}}{\#\text{long hair} + \#\text{short hair}}$$

Bayes Theorem

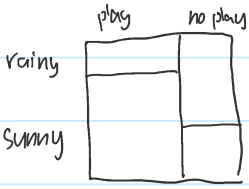
2022年1月5日 星期三　　下午1:14

$$p(\text{hypothesis} \mid \text{evidence}) = \frac{p(\text{hypothesis and evidence})}{p(\text{evidence})} = \frac{p(\text{evidence} \mid \text{hypothesis})\,p(\text{hypothesis})}{p(\text{evidence})}$$

or in short,

$$p(H \mid E) = \frac{p(E \mid H)\,p(H)}{p(E)}$$

(example) $p(\text{play} \mid \text{rainy}) = \frac{p(\text{rainy} \mid \text{play})\,p(\text{play})}{p(\text{rainy})}$



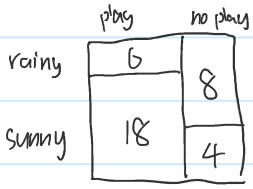
helps us to update our hypothesis when given new evidence

(example) what is the probability of play?

$$p(\text{play}) = \frac{24}{36} = \frac{2}{3}$$

(rainy | play) play

now if we have new evidence about weather and it's now rainy, what's the prob of play?



$$p(\text{play} \mid \text{rainy}) = \frac{p(\text{rainy} \mid \text{play})\,p(\text{play})}{p(\text{rainy})}$$

we have:

$$p(\text{rainy} \mid \text{play}) = \frac{6}{6+18} = \frac{1}{4}$$

$$p(\text{play}) = \frac{2}{3}$$

$p(E) = p(E \mid H)p(H) + p(E \mid \neg H)p(\neg H)$

$$p(\text{rainy}) = p(\text{rainy} \mid \text{play})\,p(\text{play}) + p(\text{rainy} \mid \text{no play})\,p(\text{no play}) = \frac{6}{6+8} \cdot \frac{2}{3} + \frac{8}{8+4} \cdot \frac{1}{3} = \frac{7}{18}$$

$$\text{therefore, } p(\text{play} \mid \text{rainy}) = \frac{1/4 \cdot 2/3}{7/18} = \frac{6}{14}$$

note that we can keep updating our hypothesis when new evidence becomes available:

$$p(\text{play} \mid \text{rainy}, \text{cold}) = \frac{p(\text{cold} \mid \text{play})\,p(\text{play} \mid \text{rainy})}{p(\text{cold})} = \frac{p(\text{cold} \mid \text{play})\,p(\text{rainy} \mid \text{play})\,p(\text{play})}{p(\text{cold} \mid)\,p(\text{rainy})}$$

or in general, $p(H \mid E) = p(H \mid e_1, e_2, \cdots, e_n) = \frac{p(e_1 \mid H)\,p(e_2 \mid H) \cdots p(e_n \mid H)\,p(H)}{p(e_1)\,p(e_2) \cdots p(e_n)}$

TERMINOLOGY ALERT!!!

$$p(H \mid E) = \frac{p(E \mid H)\,p(H)}{p(E)}$$

- $p(E \mid H)$: likelihood: the probability that the evidence is seen given that the hypothesis is true
- $p(H)$: prior: the probability that the hypothesis holds before considering the new evidence
- $p(E)$: the total probability of seeing the evidence
- $p(H \mid E)$: posterior: belief of the hypothesis after seeing the new evidence

Naïve Bayes

2022年1月3日 星期一 下午11:23

Example spam filtering

given an email with words $\vec{x} = \{x_1, x_2, \cdots, x_D\}$, is it a spam or a normal email?

suppose the prior probability of spam is $p(c)$ and we know $p(x_j|c)$ for $j \in [1, D]$ and $c = \{\text{spam}, \text{normal}\}$, we have:

$$p(c|\vec{x}) = \frac{p(\vec{x}|c)\,p(c)}{p(\vec{x})} = \frac{p(c)\,p(x_1|c)\,p(x_2|c)\cdots p(x_D|c)}{p(\vec{x})}$$

e.g., spam; each word

$$= \frac{p(c)\prod_{j=1}^{D} p(x_j|c)}{\sum_{c'} p(c')\prod_{j=1}^{D} p(x_j|c')}$$

c': spam and normal

note that for $c = \text{spam}$, $p(c|\vec{x}) = \frac{p(\vec{x}|\text{spam})\,p(\text{spam})}{p(\vec{x})}$

for $c = \text{normal}$, $p(c|\vec{x}) = \frac{p(\vec{x}|\text{normal})\,p(\text{normal})}{p(\vec{x})}$

the denominators are always the same for each class given the same email. In other words:

$$p(c|\vec{x}) \propto p(\vec{x}|c)\,p(c) = p(c)\prod_{j=1}^{D} p(x_j|c)$$

since we're doing classification, we can simply calculate $p(c)\prod_{j=1}^{D} p(x_j|c)$ for each class c and choose the largest, namely:

if $p(\text{spam})\prod_{j=1}^{D} p(x_j|\text{spam}) > p(\text{normal})\prod_{j=1}^{D} p(x_j|\text{normal})$, we predict that it is spam,

if $p(\text{spam})\prod_{j=1}^{D} p(x_j|\text{spam}) < p(\text{normal})\prod_{j=1}^{D} p(x_j|\text{normal})$, we predict that it is normal.

but how do we get $p(c)$ and $p(x_j|c)$?

use maximum likelihood!

$$\begin{aligned}
\ell(\theta) &= \sum_{i=1}^{N} (c^{(i)}, x^{(i)}) \\
&= \sum_{i=1}^{N} \log\left(p(\vec{x}^{(i)}|c^{(i)})\,p(c^{(i)})\right) \\
&= \sum_{i=1}^{N} \log\left(p(c^{(i)})\prod_{j=1}^{D} p(x_j^{(i)}|c^{(i)})\right) \\
&= \sum_{i=1}^{N} \left(\log p(c^{(i)}) + \sum_{j=1}^{D} \log p(x_j^{(i)}|c^{(i)})\right) \\
&= \sum_{i=1}^{N} \log p(c^{(i)}) + \sum_{i=1}^{N}\sum_{j=1}^{D} \log p(x_j^{(i)}|c^{(i)})
\end{aligned}$$

↑ Bernoulli log-likelihood for labels

↑ Bernoulli log-likelihood for feature x_j

for labels

we maximize $\sum_{i=1}^{N} \log p(c^{(i)})$

since here an email is either a spam or a normal, we let:

$p(c^{(i)} = 1) = \pi$ ← spam

$p(c^{(i)} = 0) = 1 - \pi$ ← not spam

$(c^{(i)} \quad 1-c^{(i)}$

$p(c^{(i)}=0) = 1-\pi$ ← not spam

hence $p(c^{(i)}) = \pi^{c^{(i)}}(1-\pi)^{1-c^{(i)}}$

$\sum_{i=1}^{N} = \sum_{i=1}^{N} c^{(i)} \log \pi + \sum_{i=1}^{N} (1-c^{(i)}) \log(1-\pi)$

by setting the derivative to zero, we get:

$\hat{\pi} = \frac{\sum_{i=1}^{N} \mathbb{1}[c^{(i)}=1]}{N}$ ← $\frac{\#\text{ spam in samples}}{\text{total }\#\text{ samples}}$

for features

we maximize $\sum_{i=1}^{N} \log p(x_j^{(i)} | c^{(i)})$

$p(x_j^{(i)} | c) = \theta_{jc}^{x_j^{(i)}} (1-\theta_{jc})^{1-x_j^{(i)}}$

$\sum_{i=1}^{N} \log p(x_j^{(i)} | c^{(i)}) = \sum_{i=1}^{N} c^{(i)} (x_j^{(i)} \log \theta_{j1} + (1-x_j^{(i)}) \log(1-\theta_{j1})) + \sum_{i=1}^{N} (1-c^{(i)}) \{x_j^{(i)} \log \theta_{j0} + (1-x_j^{(i)}) \log(1-\theta_{j0})$

by setting the derivative to zero, we get:

$\hat{\theta}_{jc} = \frac{\sum_i \mathbb{1}[x_j^{(i)}=1 \,\&\, c=c]}{\sum_i \mathbb{1}[c^{(i)}=c]}$

$\begin{cases} \frac{\#\text{ word } j \text{ in spams}}{\#\text{ spams in dataset}} & \text{if } c=1 \\ \frac{\#\text{ word } j \text{ in normal emails}}{\#\text{ normal emails in dataset}} & \text{if } c=0 \end{cases}$

K-Means

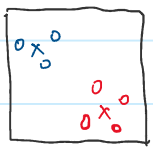
2022年1月1日 星期六　　下午7:11

unsupervised clustering algorithm

N data samples $\{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$

K cluster centers $\{m_1, m_2, \cdots, m_K\}$

for each data sample, produce a one-hot assignment $r^{(n)}$



o data sample
x cluster center

goal: find cluster centers and assignments that minimizes the total squared distance from data samples to their assigned cluster centers

$r_k^{(n)}$: n-th assignment (1 if assigned to k, 0 otherwise)

$$\min_{\{m_k\},\{r^{(n)}\}} J(\{m_k\},\{r^{(n)}\}) = \min_{\{m_k\},\{r^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} \|m_k - x^{(n)}\|^2$$

$\sum_{n=1}^{N}$: sum across all data points

$\sum_{k=1}^{K} r_k^{(n)} \|m_k - x^{(n)}\|^2$: distance between $x^{(n)}$ and its assigned cluster center

problem: need to optimize both $\{m_k\}$ and $\{r^{(n)}\}$, NP-hard!

workaround: use alternating approach, fix one and optimize the other

(can't guarantee global minimum, but good enough)

K-means

randomly initialize cluster centers $m_1, m_2, \cdots, m_K$

While assignments still changing:

① Assignment - assign each data point to the closest cluster center

$\hat{k}^{(n)} = \arg\min_k \|m_k - x^{(n)}\|^2$ ← $\hat{k}^{(n)}$-th cluster with the smallest distance

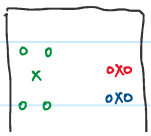
$r_k^{(n)} = \mathbb{I}(\hat{k}^{(n)} = k)$ for $k = 1, 2, \cdots, K$ ← 1 if is smallest else 0

② Refitting - move each cluster center to the mean of the data assigned to it

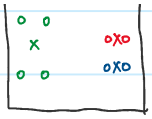
$$m_k = \frac{\sum_n r_k^{(n)} x^{(n)}}{\sum_n r_k^{(n)}}$$

k-means will always converge, as there are only finite possible clusters

problem: local minima



workaround: try many random starting points



workaround: try many random starting points

Soft k-means

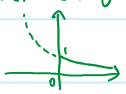
randomly initialize cluster centers $\{m_k\}$

while the change of J larger than threshold:

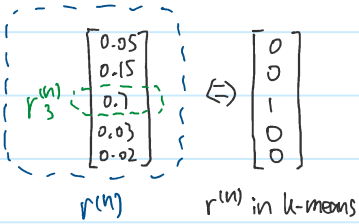
① Assignment - give each data point a soft "degree of assignment" based on softmax

the smaller the distance, the larger the weight

$$r_k^{(n)} = \frac{\exp(-\beta \|m_k - x^{(n)}\|^2)}{\sum_{j=1}^{K} \exp(-\beta \|m_j - x^{(n)}\|^2)}$$



β adjusts the sparsity of the weight



hence $r^{(n)} = \text{softmax}(-\beta \{\|m_k - x^{(n)}\|^2\}_{k=1}^{K})$

② Refitting - move each cluster center according to the weighted assignment

$$m_k = \frac{\sum_n r_k^{(n)} x^{(n)}}{\sum_n r_k^{(n)}}$$

weight ← $r_k^{(n)}$; data sample ← $x^{(n)}$; } weighted average

$\sum_n r_k^{(n)}$: sum of weight