PART I: The Decision Theory:

For the two classes case:

C_1, C_2, $\{x\}$

$\{x\}$ are divided into regions R_k ⇐ decision region

$$P\{\text{mistake}\} = P\{x \in R_1, C_2\} + P\{x \in R_2, C_1\}$$

$$= \int_{R_1} P\{x, C_2\}\, dx + \int_{R_2} P\{x, C_1\}\, dx$$

To minimize the mistake, we should rearrange the labels:

If $P\{x, C_1\} > P\{x, C_2\} \Rightarrow$ assign C_1 to x

$\because P\{x, C_1\} = P\{C_1|x\}\, P\{C_1\}$

$P\{x, C_2\} = P\{C_2|x\}\, P\{C_2\}$

$\therefore$ we only need to care about

$P\{C_1|x\}$ and $P\{C_2|x\}$.

Thus, we could broke down the classification problem into two stages:

I) Inference stage: use training data to learn a model for $P\{C_k|x\}$

II) Decision stage: use the estimated $P\{C_k|x\}$ to assign labels.

Otherwise, we could map x directly to the decision stage by

a discriminant function.

II Complexity

a) Generative models.

1. Assume or determine the class conditional density $p\{x|C_k\}$ for each class k conditionally.

2. Infer or assume $p\{C_k\}$ i.e. $p\{C_k\} = \frac{N_k}{\sum_i N_i}$

3. $p\{C_k|x\} = \frac{p\{x|C_k\}\, p\{C_k\}}{p\{x\}}$

4. Use the decision theory based on $p\{C_k|x\}$

Advantage: Could determine $p(x)$ used for outlier detection.

Disadvantage: Requires $p\{x|C_k\}$

b) Discriminant models.

1. Directly model the posterior distribution $p\{C_k|x\}$.

2. Use the decision theory to assign labels

c) Directly find a discriminant function $f(x)$ maps to the class labels.

PART II

Approach c: Discriminant functions.

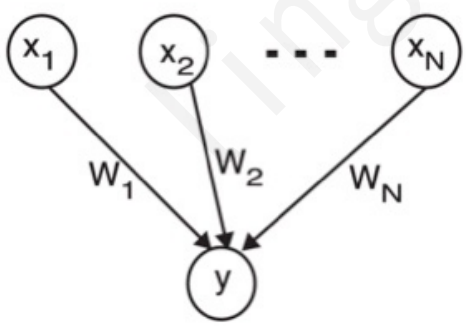
The perceptron algorithm:

Defn: $y = \text{sign}(w^T \phi(x))$

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

Error function:

$$E_p(w) = -\sum_{n \in M} w^T \phi_n t_n.$$



Since sign(a) is a piece-wise function, we can use Stochastic gradient descent to update w

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E_p(w) = w^{(\tau)} + \eta \phi_n t_n.$$

Substitute $w^{(\tau+1)}$ into $E_p(w)$:

$$E_p(w) = -w^{(\tau+1)} \phi_n t_n$$

$$= -w^{(\tau)T} \phi_n t_n - \eta (\phi_n t_n)^T \phi_n t_n.$$

$$< -w^{(\tau)T} \phi_n t_n.$$

Perceptron Convergence Theorem

For any finite set of linearly separable labeled examples, the Perceptron Learning Algorithm will halt after a finite number of iterations.

proof: Let w^* be the weight vector that perfectly seperates all samples.

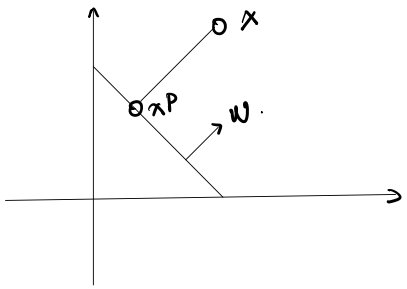
$$\Rightarrow \forall n. \quad w^{*T} \cdot x^n \cdot t^n > 0$$

Let w^n denote the weight vector in the nth step.

θ_{in}: the angle between w^* and w^n

$$C(n) = \cos\theta(n) = \frac{w(n)^T \cdot w^*}{\|w(n)\| \|w^*\|}$$

Margin of a point: (The distance from the point x to w^*)



d: the vector for x to w^* with the minimum length

$$x^P = x - d$$

$\because d$ is parallel to w

$\therefore d = \alpha \cdot w$

$\because x^P \in$ the seperating plane $w^{*T} x = 0$

$\therefore w^T \cdot x^P = 0$

$$w^T \cdot (x - d) = w^T (x - \alpha w) = 0$$

$$\Rightarrow \alpha = \frac{w^T \cdot x}{w^T \cdot w}$$

$$\Rightarrow \|d\|_2 = \frac{|w^T \cdot x|}{\|w\|_2}$$

$$\delta^n := \frac{t^n \cdot w^{*T} \cdot x^n}{\|w^*\|_2} \Rightarrow \text{the signed margin.}$$

$\because w^{*T}$ correctly seperates all samples

$\therefore \delta^n$ is strictly positive.

$$\delta^* := \min_n \delta^n > 0.$$

$$w^n - w^{n-1} \equiv \Delta w^n = \eta t^n x^n.$$

∵ The update will happen if and only if there is an error. : $t^n \cdot w^{(n-1)T} \cdot x^n < 0.$

$$\Rightarrow w^{(n-1)T} \cdot \Delta w^n$$

$$= w^{(n-1)T} \cdot \eta t^n \cdot x^n < 0.$$

$$w^{nT} \cdot w^* = w^{(n-1)T} \cdot w^* + \Delta w^{nT} \cdot w^*$$

$$= w^{(n-1)T} w^* + \eta t^n \cdot w^{*T} \cdot x^n$$

$$= w^{(n-1)T} \cdot w^* + \eta \delta^n \|w^*\|$$

$$\geq w^{(n-1)T} \cdot w^* + \eta \delta^* \|w^*\|$$

$$\geq w^{(n-2)T} \cdot w^* + 2\eta \delta^* \|w^*\|$$

$$\geq \eta n \delta^* \|w^*\|$$

$$\|w^{(n)}\|^2 = \|w^{(n-1)} + \Delta w^{(n)}\|_2$$

$$= \|w^{(n-1)}\|_2 + \|\Delta w^{(n)}\|_2 + 2 w^{(n-1)T} \Delta w^{(n)} \quad \Leftarrow f^{(n)} = \begin{cases} +1 \\ -1 \end{cases}$$

$$= \|w^{(n-1)}\|_2 + \eta^2 \|x^{(n)}\|^2 + 2 w^{(n)T} \Delta w^{(n)}$$

$$< \|w^{(n-1)}\|_2 + \eta^2 D^2$$

$$D := \max_n \|x^{(n)}\|$$

By applying the inequality n times:

$$\|w^{(n)}\|^2 \leq \|w^{(0)}\|_2 + n\eta^2 D^2$$

$\Rightarrow$ for sufficiently large n:

$$\cos\theta^{(n)} \geq \frac{\eta n \delta^* \|w^{(*)}\|}{\sqrt{n}\, \eta D \|w^{(*)}\|} = \frac{\delta^*}{D} \sqrt{n}$$

Assume, for contradiction, if the algorithm will go on forever:

$\cos\theta^{(n)}$ will be greater than 1 for large n.

$\Rightarrow$ After a finite step, the algorithm will halt.

Convergence Time of the Perceptron Algorithm

Define: $\delta_{max} = \max_{w^*} \delta^*$

$\because \frac{\delta^*}{D}\sqrt{n} \leq 1$

$\therefore n \leq \frac{D^2}{\delta^{*2}} \leq \frac{D^2}{\delta_{max}^2}$

The convergence time depends on the margin.

If the two points with different labels are close to each other, the perceptron algorithm will take a long time.

Definition: Angular Margin: $\gamma := \min_{i \in [m]} |x_i \cdot w^*| = \delta^* \cdot \|w^*\|$

PART III: Probabilistic Generative Models: Bayes classifier and Naive Bayes

For the two classes cases:

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

$$= \frac{1}{1 + \frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}}$$

let $a := \ln \frac{p\{x|c_1\}\, p\{c_1\}}{p\{x|c_2\}\, p\{c_2\}}$

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

Defn: logistic sigmoid function:

$$\sigma(a) := \frac{1}{1+\exp\{-a\}}$$

properties: I) Symmetric: $\sigma(a) = 1 - \sigma(-a)$

II) Inverse: $a = \ln \frac{\sigma}{1-\sigma}$ (logic function)

$$= \ln \frac{p\{c_1|x\}}{p\{c_2|x\}}$$

For $k > 2$ classes:

$$p\{c_k|x\} = \frac{p\{x|c_k\}\, p\{c_k\}}{\sum_j p\{x|c_j\}\, p\{c_j\}}$$

$$= \frac{\exp\{a_k\}}{\sum_j \exp\{a_j\}}$$ "softmax function"

$a_i := \ln p\{x|c_i\}\, p\{c_i\}$

Softmax function is a smoothed version of the max function. When $a_k \gg a_j \quad k \neq j$: $p\{c_k|x\} \simeq 1$

1° If the inputs are all continuous variables

Assume, the class conditional densities are Gaussian.

$$p\{x|C_k\} = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\right\}.$$

For the two-classes cases:

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

$$\ln\frac{p\{x|C_1\}}{p\{x|C_2\}} = \ln\frac{\exp\{-(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\}}{\exp\{-(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\}}$$

$$= \Sigma^{-1}\left[(\mu_1-\mu_2)^T x - \frac{1}{2}\mu_1^T\mu_1 + \frac{1}{2}\mu_2^T\mu_2\right]$$

$$\Rightarrow p\{C_1|x\} = \sigma(w^T x + w_0)$$

$$w := \Sigma^{-1}(\mu_1 - \mu_2)$$

$$w_0 := -\frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \ln\frac{p(C_1)}{p(C_2)}$$

For the $k \geq 2$ classes cases:

$$a_k(x) = \ln p\{x|C_k\}p\{C_k\}$$

$$\Rightarrow p\{C_k|x\} = \sigma(w_k^T x + w_0)$$

$$w_k := \Sigma^{-1} \mu_k$$

$$w_0 := -\frac{1}{2} \mu_k \Sigma^{-1} \mu_k + \ln p(C_k)$$

Once we have determined the distribution of $p(x|C_k)$, together with $p(C_k)$, we can get the maximum likelihood solution of the parameters.

For two-classes case:

let $p(C_1) = \pi$, $p(C_2) = 1-\pi$

$$\Rightarrow p(x_n, C_1) = p(C_1)\, p(x|C_1) = \pi N(x_n|\mu_1, \Sigma)$$

$$p(x_n, C_2) = p(C_2)\, p(x|C_2) = (1-\pi) N(x_n|\mu_2, \Sigma)$$

$$\Rightarrow p(t|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^{N} [\pi N(x_n|\mu_1, \Sigma)]^{t_n} [(1-\pi) N(x_n|\mu_2, \Sigma)]^{1-t_n}$$

$$\log p(t|\pi, \mu_1, \mu_2, \Sigma) = \sum_{n=1}^{N} t_n \log \pi N(x_n|\mu_1, \Sigma) + (1-t_n) \log (1-\pi) N(x_n|\mu_2, \Sigma)$$

w.r.t. π:

The terms containing π are:

$$\sum_{n=1}^{N} (t_n \ln \pi + (1-t_n) \ln (1-\pi))$$

Take derivative, get:

$$\hat{\pi} = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1+N_2} \quad \left(\text{Note that } t_i = \begin{cases} 1 & \text{if } i \in C_1 \\ 0 & \text{otherwise} \end{cases}\right)$$

w.r.t. μ_1:

$$\sum_{n=1}^{N} t_n \ln N\{x_n|\mu_1,\Sigma\} = -\frac{1}{2}\sum_{n=1}^{N} t_n (x_n-\mu_1)^T\Sigma^{-1}(x_n-\mu_1) + \text{const.}$$

Taking derivative:

$$\sum_{n=1}^{N} t_n (x_n-\mu_1)^T\Sigma^{-1} = 0.$$

$$\Rightarrow \hat{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n x_n.$$

Similarly for μ_2:

$$\hat{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1-t_n) x_n.$$

w.r.t. Σ:

$$-\frac{1}{2}\sum_{n=1}^{N} t_n \ln|\Sigma| - \frac{1}{2}\sum_{n=1}^{N} t_n (x_n-\mu_1)^T\Sigma^{-1}(x_n-\mu_1)$$

$$-\frac{1}{2}\sum_{n=1}^{N} (1-t_n)\ln|\Sigma| - \frac{1}{2}\sum_{n=1}^{N} (1-t_n)(x_n-\mu_2)^T\Sigma^{-1}(x_n-\mu_2)$$

$$= -\frac{N}{2}\ln|\Sigma| - \frac{N}{2}\mathrm{Tr}\{\Sigma^{-1}S\}.$$

$$S := \frac{N_1}{N}S_1 + \frac{N_2}{N}S_2$$

$$S_1 = N_1^{-1}\sum_{n\in C_1}(x_n-\mu_1)(x_n-\mu_1)^T$$

$$S_2 = N_2^{-1}\sum_{n\in C_2}(x_n-\mu_2)(x_n-\mu_2)^T$$

For multiple classes.

$$P\{C_1\} = \pi_1, \quad P\{C_2\} = \pi_2 \cdots \quad P\{C_K\} = \pi_K = 1 - \pi_1 - \pi_2 - \cdots - \pi_{K-1}$$

$$P\{x_n, C_j\} = P\{C_j\} P\{x_n | C_j\} = \pi_j N(x_n | \mu_j, \Sigma)$$

Similarly, the likelihood function is given by:

$$p(\{x_n, t_n\} | \{\pi_k\}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \{p(\phi_n | C_k) \pi_k\}^{t_{nk}}$$

$$\ln p(\{x_n, t_n\} | \{\pi_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \{\ln P\{\phi_n | C_k\} + \ln \pi_k\}$$

The problem is now a convex optimization with constraints.

$$\max \ln p(\{x_n, t_n\} | \{\pi_k\}) \quad \text{s.t.} \quad \sum_k \pi_k = 1$$

$\Rightarrow$ By Lagrange:

$$\ln p(\{x_n, t_n\} | \{\pi_k\}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

Set the derivative to zero:

$$\sum_{n=1}^{N} \frac{t_{nk}}{\pi_k} + \lambda = 0$$

$$\Rightarrow -\pi_k \cdot \lambda = \sum_{n=1}^{N} t_{nk} = N_k.$$

$$\sum_k -\pi_k \cdot \lambda = -\lambda = \sum_k N_k = N$$

$$\Rightarrow \lambda = -N$$

$$\Rightarrow \hat{\pi}_k = \frac{N_k}{N}$$

$$\ln p(\{x_n, t_n\} | \{\pi_k\}) = -\frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \{\ln|\Sigma| + (x_n - \mu_k)^T \Sigma^{-1} (x_n - \mu_k)\}$$

w.r.t. μ_k:

$$\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \Sigma^{-1} (x_n - \mu_k) = 0$$

For $k = j$:

$$\sum_{n=1}^{N} t_{nj} \Sigma^{-1} (x_n - \mu_j) = 0$$

$$\Rightarrow \hat{\mu}_j = \frac{1}{N_j} \sum_{n=1}^{N} t_{nj} x_n.$$

For Σ:

Rewriting the log likelihood function as:

$$-\frac{1}{2} b \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \{\ln|\Sigma| + Tr[\Sigma^{-1} (x_n - \mu_k)(\phi_n - \mu_k)^T]\}$$

Taking derivative:

$$\frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \{\Sigma - (x_n - \mu_k)(x_n - \mu_k)^T\} = 0.$$

$$\Rightarrow \quad \hat{\Sigma} = \sum_{k=1}^{K} \frac{N_k}{N} S_k$$

$$S_k = \frac{1}{N_k} \sum_{n=1}^{N} t_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

If we have multiple features. $X \in \mathbb{R}^{N \times D}$ is a matrix.

we may use the Naive Bayes Assumption.

$$P\{x|y\} = \prod_{\alpha=1}^{D} P\{x_\alpha | y\} = \prod_{i=1}^{N} \prod_{\alpha=1}^{D} P\{x_{i\alpha} | y_i\}$$

which means: the feature values are independent of the given label.

For the continuous features. we just need to repeat the above procedures:

$$P\{x_\alpha | C_k\} = N\{\mu_{\alpha k}, \sigma^2_{\alpha k}\}.$$

Taking derivative to the log likelihood w.r.t. $\mu_{\alpha k}$ and $\sigma^2_{\alpha k}$.

$$\Rightarrow \mu_{\alpha k} = \frac{1}{N_k} \sum_{i=1}^{N} I(t_i = C_k) x_{i\alpha}.$$

$$\sigma^2_{\alpha k} = \frac{1}{N_k} \sum_{i=1}^{N} I(t_i = C_k)(x_{i\alpha} - \mu_{\alpha k})^2$$

2° Dealing with Discrete Features

If $[x]_\alpha \in \{f_1, f_2, \dots, f_{k_\alpha}\}$.

i.e. male / female single / married / widowed.

Under the Naive Bayes assumption, the log-likelihood function could be written as:

$$L(\theta) = \sum_{i=1}^{n} \log P\{x^{(i)}, y^{(i)}\}$$

$$= \sum_{i=1}^{n} \log \left[\pi_{y_i} \prod_{j=1}^{D} P\{x_j^{(i)} \mid y^{(i)}\} \right]$$

$$= \sum_{i=1}^{n} \log \pi_{y_i} + \sum_{i=1}^{n} \sum_{j=1}^{D} \log P\{x_j^{(i)} \mid y^{(i)}\}$$

$$= \sum_{c_i \in \{1,\dots,k\}} \text{count}(c_i) \cdot \log \pi_{c_i} +$$

$$\sum_{j=1}^{D} \sum_{c_i \in \{1,\dots,j\}} \sum_{x_j \in \{f_1,\dots,f_{k_j}\}} \text{count}_j(x \mid y) \log P\{x_j \mid c_i\}.$$

By the previous derivation, we have verified that the MLE of π_{c_i} is $\frac{\text{count}(c_i)}{N}$

Thus, to maximize the log likelihood, we need to

maximize.

$$\sum_{x_j \in \{f_1, \dots, f_{|j|}\}} \text{count}_j \log P\{x_j \mid c_i\}.$$

Note that, x_j is a categorical variable, thus, $P\{x_j \mid c_i\}$ is a multinomial distribution.

We simplify the above notations.

we want to find a distribution q, s.t.

$$q^* = \arg\max_{q \in P_Y} \sum_{y \in Y} c_y \log q_y$$

$$P_Y := \left\{ q \in \mathbb{R}^{|Y|} : \forall y \in Y,\ q_y \geq 0;\ \sum_{y \in Y} q_y = 1 \right\}.$$

$$\Leftrightarrow:$$

$$\max \sum_{y \in Y} c_y \log q_y \quad \text{s.t.} \sum_{y \in Y} q_y = 1$$

By Lagrange:

$$g(\lambda, q) = \sum_{y \in Y} c_y \log q_y - \lambda \left[\sum_{y \in Y} q_y - 1 \right]$$

$$\frac{d}{dq_y} g(\lambda, q) = \frac{c_y}{q_y} - \lambda = 0$$

$$\Rightarrow \theta_y = \frac{c_y}{\lambda}$$

By $\sum_y \theta_y = 1$: $\theta_y = \frac{c_y}{\sum_y c_y}$

Substituting $c_y = \text{count}\{x_j \mid c_i\}$.

$$\Rightarrow P_j(x|y) = \frac{\text{count}(x_j \mid c_i)}{\sum_{x_j \in \{f_1, \dots, f_D\}} \text{count}(x_j \mid c_i)}$$

$$\Leftrightarrow P\{x_\alpha = j \mid y = c\} = \frac{\sum_{i=1}^{n} I[y_i = c \wedge x_{i\alpha} = j]}{\sum_{i=1}^{n} I(y_i = c)}$$

3° Dealing with multinomial inputs.

$x_\alpha \in \{0, 1, 2, \dots, m\}$ and $m = \sum_{\alpha=1}^{D} x_\alpha$.

By multinomial distribution:

$$P\{x \mid m, y = c\} = \frac{m!}{x_1! \, x_2! \cdots x_d!} \prod_{\alpha=1}^{D} [\theta_{\alpha c}]^{x_\alpha}$$

$\theta_{\alpha c}$: the parameter of the multinomial distribution

the probability of selecting x_α.

→ The likelihood function can be written as:

$$P\{X \mid y=c\} = \prod_{i=1}^{N} \prod_{\alpha=1}^{D} \frac{M_i!}{x_{i1}! \cdots x_{iD}!} [\theta_{\alpha c}]^{x_\alpha}.$$

By the above derivation, to obtain the MLE of $\theta_{\alpha c}$, we just need to maximize:

$$\log P\{x \mid y=c\} = \sum_{i=1}^{N} \sum_{\alpha=1}^{D} x_{i\alpha} \log(\theta_{\alpha c}) + \log(M_i) - \log(x_{i1}! \cdots x_{iD}!)$$

$$\text{s.t.} \sum_{i=1}^{D} \theta_{ic} = 1$$

By Lagrange:

$$f(\theta, \lambda) = \sum_{i=1}^{N} \sum_{\alpha=1}^{D} x_{i\alpha} \log(\theta_{kc}) + \lambda \left(1 - \sum_{j=1}^{D} \theta_{jc}\right)$$

$$\frac{d}{d\theta_k} f(\theta, \lambda) = \sum_{i=1}^{N} \frac{x_{ik}}{\theta_k} - \lambda = 0$$

$$\Rightarrow \sum_{i=1}^{N} x_{ik} = \lambda \theta_{kc}$$

$$\sum_{\alpha=1}^{D} \sum_{i=1}^{N} x_{i\alpha} = \sum_{\alpha=1}^{D} \lambda \theta_{\alpha c} = \lambda.$$

$$\Rightarrow \hat{\theta}_{kc} = \frac{\sum_{i=1}^{N} x_{ik}}{\sum_{\alpha=1}^{D} \sum_{i=1}^{N} x_{i\alpha}}.$$

Note that the above derivation under the known y label.

$$\hat{\theta}_{kc} = \frac{\sum_{i=1}^{n} I\{y_i = c\} x_{ik}}{\sum_{i=1}^{N} \sum_{\alpha=1}^{D} I\{y_i = c\} x_{i\alpha}}.$$

Theorem: Under some certain cases, the naive bayes classifier leads to a linear combination.

① Suppose that $y_i \in \{-1, +1\}$, and features are multinomial.

To prove the linearity, we need to show that:

$$h(x) = \arg\max_{y} \prod_{\alpha=1}^{D} P\{x_\alpha | y\} = \text{sign}(w^T x + b)$$

which means:

$$w^T x + b > 0 \iff h(x) = +1$$

Define:

$$P\{x_\alpha | y = +1\} \propto \theta_{\alpha+}^{x_\alpha}$$

$$P\{Y = +1\} = \pi_+$$

let $[w]_\alpha = \log(\theta_{\alpha+}) - \log(\theta_{\alpha-})$

$$b = \log(\pi_+) - \log(\pi_-)$$

Then:

$$w^T x + b > 0$$

$$\Leftrightarrow \sum_{\alpha=1}^{D} [x]_\alpha \left(\log(\theta_{\alpha+}) - \log(\theta_{\alpha-})\right) + \log(\pi_+) - \log(\pi_-)$$

$$\Leftrightarrow \exp\left\{\sum_{\alpha=1}^{D} [x]_\alpha \left(\log(\theta_{\alpha+}) - \log(\theta_{\alpha-})\right) + \log(\pi_+) - \log(\pi_-)\right\} > 1$$

$$\Leftrightarrow \prod_{\alpha=1}^{D} \frac{\exp\{\log \theta_{\alpha+}^{[x]_\alpha} + \log(\pi_+)\}}{\exp\{\log \theta_{\alpha-}^{[x]_\alpha} + \log(\pi_-)\}} > 1$$

$$\Leftrightarrow \prod_{\alpha=1}^{D} \frac{\theta_{\alpha+}^{[x]_\alpha} \pi_+}{\theta_{\alpha-}^{[x]_\alpha} \pi_-} > 1$$

$$\Leftrightarrow \frac{\prod_{\alpha=1}^{D} P\{|x|_\alpha \mid Y=+1\} \pi_+}{\prod_{\alpha=1}^{D} P\{|x|_\alpha \mid Y=-1\} \pi_-} > 1$$

By NB assumption $\Leftarrow$

$$\frac{P\{x \mid Y=+1\} \pi_+}{P\{x \mid Y=-1\} \pi_-} > 1$$

$$\Leftrightarrow P\{Y=+1 \mid x\} > P\{Y=-1 \mid x\}$$

$$\Leftrightarrow \quad \arg\max_y P\{Y=y|x\} = +1$$

Thus: $w^T x + b > 0 \quad \Leftrightarrow \quad +1 = \arg\max_y P\{y\} \prod_{\alpha=1}^{D} P\{x_\alpha | y\}$

② Gaussian Naive Bayes.

If the covariance is shared between classes

$$h(x) = \arg\max_y P(y) \prod_{\alpha=1}^{D} P\{x_\alpha | y\} = 1 .$$

$$\Leftrightarrow \quad P\{x | Y=1\} > P\{x | y=2\}.$$

$$\Leftrightarrow \log \pi_1 - \frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1) > \log \pi_0 - \frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)$$

$$\Leftrightarrow x^T \Sigma^{-1} x - 2\mu_1^T \Sigma^{-1} x + \mu_1^T \Sigma^{-1} \mu_1 + C > x^T \Sigma^{-1} x - 2\mu_0^T \Sigma^{-1} x + \mu_0^T \Sigma^{-1} \mu_0$$

$$\Leftrightarrow \left[2(\mu_0 - \mu_1)^T \Sigma^{-1}\right] x - (\mu_0 - \mu_1)^T \Sigma^{-1} (\mu_0 - \mu_1) > C$$

$$\Rightarrow \quad w^T \cdot x + b > 0$$

$$w := 2(\mu_0 - \mu_1)^T \Sigma^{-1})^T$$

$$b := -(\mu_0 - \mu_1)^T \Sigma^{-1} (\mu_0 - \mu_1) - C$$

PART IV: Probalistic Discriminative Models: Logistic Regression.

By the discussion above, we see that

$$p\{C_1 | x\} = y(\phi) = \sigma(w^T x)$$

In this section, we directly model the conditional distribution $p\{C_1|x\}$ rather than assuming the class conditional function $p\{x|C_1\}$ first.

Given a dataset: $\{\phi_i, t_i\}$ $t_i \in \{0,1\}$:

$$\text{let } y_n = p\{C_1|\phi_n\}$$

$$\Rightarrow p\{t|w\} = \prod_{n=1}^{N} y_n^{t_n} \{1-y_n\}^{1-t_n}$$

Define the error function as:

$$E(w) = -\ln p\{t|w\} = -\sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\} \quad \text{"cross entropy"}$$

where $y_n = \sigma(w^T \phi_n)$

For $\sigma(\cdot)$:

$$\frac{\partial \sigma}{\partial a} = -\frac{-\exp(-a)}{(1+\exp(-a))^2} = \frac{\exp(-a)}{\exp(-a)^2 + 2\exp(-a) + 1}$$

$$= \frac{\exp(-a)}{1+\exp(-a)} \cdot \frac{1}{1+\exp(-a)} = \sigma(1-\sigma)$$

$$\Rightarrow \nabla E(w) = \sum_{n=1}^{N} \left[-t_n \frac{1}{y_n} \cdot (1-y_n) \cdot y_n + \frac{1-t_n}{1-y_n} \cdot y_n (1-y_n) \right] \phi_n$$

$$= \sum_{n=1}^{N} \left[-t_n (1-y_n) + y_n (1-t_n) \right] \phi_n$$

$$= \sum_{n=1}^{N} \left[-t_n + y_n + t_n y_n - y_n t_n \right] \phi_n$$

$$= \sum_{n=1}^{N} (y_n - t_n) \phi_n .$$

For $k>2$ classes:

Having obtained above.

$$p\{C_k | \phi\} = y_k(\phi) = \frac{\exp\{a_k\}}{\sum_j \exp\{a_j\}}$$

$$a_k := w_k^T \phi$$

$$\frac{\partial y_k}{\partial a_j} :$$

If $j=k$:
$$\frac{\exp\{a_k\} \sum_j \exp\{a_j\} - \exp\{a_k\} \cdot \exp\{a_k\}}{\left| \sum_j \exp\{a_j\} \right|^2}$$

$$= \frac{\exp\{a_k\}}{\sum_j \exp\{a_j\}} - \frac{\exp\{a_k\}}{\sum_j \exp\{a_j\}} \cdot \frac{\exp\{a_k\}}{\sum_j \exp\{a_j\}}$$

$$= \quad y_k - y_k^2 = y_k(1 - y_k)$$

对 $j \neq k$:

$$- \frac{\exp\{a_k\} \cdot \exp\{a_j\}}{(\sum_j \exp\{a_j\})^2}$$

$$= \quad - y_k \cdot y_j$$

$$\Rightarrow \quad \frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$$

The likelihood fuction: By one-hot encoding, with N samples and k classes, our target label matrix is $T \in \mathbb{R}^{N \times k}$

$$p\{T | w_1, \ldots, w_k\} = \prod_{n=1}^{N} \prod_{a=1}^{k} p\{C_k | \phi_n\}^{t_{nk}}$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{k} y_{nk}^{t_{nk}}$$

$$\Rightarrow E(w_1, \ldots, w_k) = -\ln p\{T | w_1, \ldots, w_k\}$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{k} t_{nk} \ln y_{nk}$$

$$\nabla_{w_j} E(w_1, \dots, w_K) = \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\partial E}{\partial y_{nk}} \cdot \frac{\partial y_{nk}}{\partial a_{nj}} \cdot \frac{\partial a_{nj}}{\partial w_j}$$

$$= \sum_{n=1}^{N} - \sum_{k=1}^{C} \frac{t_{nk}}{y_{nk}} y_{nk} (I_{kj} - y_{nj}) \cdot \phi_n$$

$$\because \sum_k t_{nk} = 1 \Rightarrow$$

$$= \sum_{n=1}^{N} (y_{nj} - t_{nj}) \cdot \phi_n.$$

⇒ Logistic Regression does not have a closed-form solution, we need a sequential algorithm to approximate it.

Iterative reweighted least squares.

The Newton Raphson iteration:

$f(r)$ takes root at r, we want to find r.

given a good estimate of r : x_0, $r = x_0 + h$, h is "small"

$$0 = f(r) = f(x_0 + h) \approx f(x_0) + h f'(x_0)$$

$$\Rightarrow h \approx -\frac{f(x_0)}{f'(x_0)}$$

$$\Rightarrow r = x_0 + h \approx x_0 - \frac{f(x_0)}{f'(x_0)}$$

let $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

To our problem, since $E(w)$ is clearly concave, we want to find the root of $\nabla E(w)$. Thus:

$$w^{(n+1)} = w^{(n)} - H^{-1} \nabla E(w)$$

For $k=2$:

$$\nabla E(w) = \sum_{n=1}^{N} (y_n - t_n)\, \phi_n = \Phi^T (y - t)$$

$$H = \nabla\nabla E(w) = \sum_{n=1}^{N} y_n(1-y_n)\, \phi_n \phi_n^T = \Phi^T R \Phi$$

R is a diagonal matrix: $R_{nn} = y_n(1-y_n)$

$$w^{(n+1)} = w^{(n)} - (\Phi^T R \Phi)^{-1} \Phi^T (y - t)$$

$$= (\Phi^T R \Phi)^{-1} \{ \Phi^T R \Phi w^{(n)} - \Phi^T (y - t) \}$$

$$= (\Phi^T R \Phi)^{-1} \Phi^T R z$$ ← the normal equation in the least squares regression.

where $z := \Phi w^{(n)} - R^{-1}(y - t)$

$$\Rightarrow w^{(n+1)} = \arg\min_w (z - w^T \phi)^T R (z - w^T \phi)$$

For $K>2$:

$$H_{ij} = \nabla_{w_j} \nabla_{w_i} E(w) = -\sum_{n=1}^{N} y_{nj}(I_{ij} - y_{ij}) \phi_n \phi_n^T$$

Ridge Logistic Regression.

Why we need Ridge estimation?

1° n: number of samples ; p: number of features.

If $p > n$: our discriminative boundary:

$$w^T \phi = 0.$$

$\because p > n \therefore$ the null space of w is non-trivial

$\therefore$ for some $\gamma > 0$, there exists:

$$w^T \phi = 0$$

$$w^T(\phi + \gamma) = 0$$

$\Rightarrow$ an infinite number of estimators of the logistic regression exists.

2° When the dataset is perfectly linear separable.

$\because$ there exists an optimal hyperplane that seperates correctly all the samples

$\therefore$ For the optimal fit: we have $p\{y_{i=1} | \phi_i\} \in \{0, 1\}$.

⇒ The Loss function becomes:

$$L = -\sum_{i=1}^{N} t_i \log\{P(y_i=1 \mid x_i)\} + (1-t_i)\log\{P(y_i=0 \mid x_i)\} = 0$$

which does not contain any logistic parameters.

⇒ $P(y_i=1 \mid \phi_i) = \frac{1}{1+\exp(-w^T\phi_i)} \in \{0,1\}$.

this will only happen when w approaches negative infinity or positive infinity.

⇒ The logistic regression cannot be learned from the seperable data.

Thus, we need want to add a ridge penalty to the loss function.

$$E(w,\lambda) = -\sum_{i=1}^{N} \left[t_i \log(y_i) + (1-t_i)\log(1-y_i) \right] + \frac{\lambda}{2} \|w\|_2^2$$

$$\nabla_{w_j} E(w,\lambda) = \nabla_{w_j} E(w) + \lambda w_j$$

$$\nabla_{w_i} \nabla_{w_j} E(w,\lambda) = \nabla_{w_i} \nabla_{w_j} E(w) + \lambda I_{ij}$$

For $K=2$ case, we reformulate the Newton-Raphson algorithm:

$$\nabla E(w,\lambda) = \Phi^T (y-t) + \lambda w$$

$$\nabla\nabla E(w,\lambda) = \Phi^T R \Phi + \lambda I_{pp}, \quad R_{nn} = y_n(1-y_n)$$

$$\Rightarrow \quad w^{(new)} = w^{(old)} - (\Phi^T R \Phi + \lambda I_{pp})^{-1} \left[\Phi^T (y - t) + \lambda w \right]$$

let $V = \Phi^T R \Phi + \lambda I_{pp}$

$$* = w^{(old)} - V^{-1} \left[\Phi^T (y - t) + \lambda w^{(old)} \right]$$

$$= V^{-1} V w^{(old)} - \lambda V^{-1} w^{(old)} - V^{-1} \Phi^T R R^{-1} (y - t)$$

$$= V^{-1} \Phi^T R \left[\Phi w^{(old)} - R^{-1} (y - t) \right]$$

let $z := \Phi w^{(old)} - R^{-1} (y - t)$

$$* = (\Phi^T R \Phi + \lambda I_{pp})^{-1} \Phi^T R \cdot z$$

the Ridge estimation of the weighted least square.

Bayesian Logistic Regression.

The bayesian treatment of the logistic regression is more complex than the linear regression, since the posterior is no longer Gaussian.

Laplace Approximation.

Laplace Approximation is used to find a Gaussian approximation to a probability density over a set of continuous variables.

Suppose:

$$p(z) = \frac{1}{Z} f(z)$$

$$Z := \int f(z)\, dz \qquad \text{"normalization coefficient"}$$

The Laplace method wants to find a Gaussian approximation whose mean is the mode of $p(z)$

To find the mode of $p(z)$:

we want to find a z_0 s.t.:

$$\frac{d}{dz} p(z)\Big|_{z=z_0} = 0$$

$$\Leftrightarrow \quad \frac{d}{dz} f(z)\Big|_{z=z_0} = 0$$

By Taylor expansion:

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2} A (z - z_0)^2$$

where: $$A = -\frac{d}{dz^2} \ln f(z)\Big|_{z=z_0}$$

$$\Rightarrow f(z) \approx f(z_0) \exp\left\{-\frac{A}{2}(z - z_0)^2\right\}$$

$$Z = \int f(z)\,dz$$

$$= f(z_0) \int \exp\left\{-\frac{A}{2}(z-z_0)^2\right\} dz$$

$$= f(z_0) \cdot \sqrt{\frac{2\pi}{A}} \quad \Leftarrow \text{Gaussian Integral}$$

⇒ The approximation $q(z)$:

$$q(z) = \frac{1}{Z} f(z) = \left(\frac{A}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{A}{2}(z-z_0)^2\right\}.$$

For the multi-dimensional case:

The Taylor expansion step:

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2}(z-z_0)^T A (z-z_0)$$

$$A = -\nabla\nabla \ln f(z)\Big|_{z=z_0}$$

$$\Rightarrow f(z) \propto f(z_0) \exp\left\{-\frac{1}{2}(z-z_0)^T A (z-z_0)\right\}.$$

⇒ The approximation $q(z)$:

$$q(z) = \frac{|A|^{\frac{1}{2}}}{(2\pi)^{\frac{M}{2}}} \exp\left\{-\frac{1}{2}(z-z_0)^T A (z-z_0)\right\}.$$

For the logistic regression, we firstly assume the prior is Gaussian:

$$p(w) = N\{w|m_0, S_0\}$$

⇒ The posterior is:

$$p\{w|t\} \propto p\{w\}\, p\{t|w\}$$

$$\Rightarrow \ln p\{w|t\} = -\frac{1}{2}(w-m_0)^T S_0^{-1}(w-m_0) + \sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\} + \text{const}$$

and $y_n = \sigma(w^T\phi)$

To perform the Laplace approximation, we need firstly to compute the w_{MAP} by maximizing the posterior distribution.

By Laplace Approximation, the variance is:

$$S_N = -\nabla\nabla \ln p(w|t) = S_0^{-1} + \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T$$

⇒The Gaussian Approximation:

$$q(w) = N\{w|w_{MAP}, S_N\}$$

The predictive distribution is then given by:

$$p\{C_1 \mid \phi, t\} = \int p\{C_1 \mid \phi, w\}\, p\{w \mid t\}\, dw$$

$$\simeq \int \sigma(w^T \phi)\, q(w)\, dw.$$

Let $a = w^T \phi$

$$\Rightarrow \quad \sigma(w^T \phi) = \int \delta(a - w^T \phi)\, \sigma(a)\, da$$

$\delta(a)$ is a Dirac delta function

$$\delta(a) = \begin{cases} 1, & a = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Thus:

$$\int \sigma(w^T \phi)\, q(w)\, dw = \int \sigma(a)\, p(a)\, da$$

where $p(a) = \int \delta(a - w^T \phi)\, q(w)\, dw.$

$$\mu_a = E[a] = \int p(a) \cdot a\, da = \int q(w) \cdot w^T \phi\, dw = w_{MAP}^T \cdot \phi.$$

$$\sigma_a^2 = \mathrm{Var}[a] = \int p(a) \{a^2 - E[a]^2\}\, da$$

$$= \int q(w) \{(w^T \phi)^2 - (m_N^T \phi)^2\}\, dw$$

$$= \phi^T S_N \phi$$

Thus:

$$p(a|\phi, t) = \int \sigma(a)\, p(a)\, da$$

$$= \int \sigma(a)\, N(a|\mu_a, \sigma_a^2)\, da.$$ "A convolution of the sigmoid and Gaussian"

The above does not have an analytical solution.

1° We could use MCMC to get an approximation of the analytical solution

2° We could approximate the sigmoid fuction with probit fuction to get an analytical expression.

probit fuction: $\Phi(a) = \int_{-\infty}^{a} N(\theta|0,1)\, d\theta = \frac{1}{2}\{1 + \frac{1}{\sqrt{2}} \text{erf}(a)\}$.

For the method 2.

To obtain the best approximation, we need firstly to rescale the horizontal axis.

$$\left.\frac{d\sigma}{da}\right|_{a=0} = \sigma(0)(1-\sigma(0))$$

$$= \frac{1}{2}(1-\frac{1}{2}) = \frac{1}{4}$$

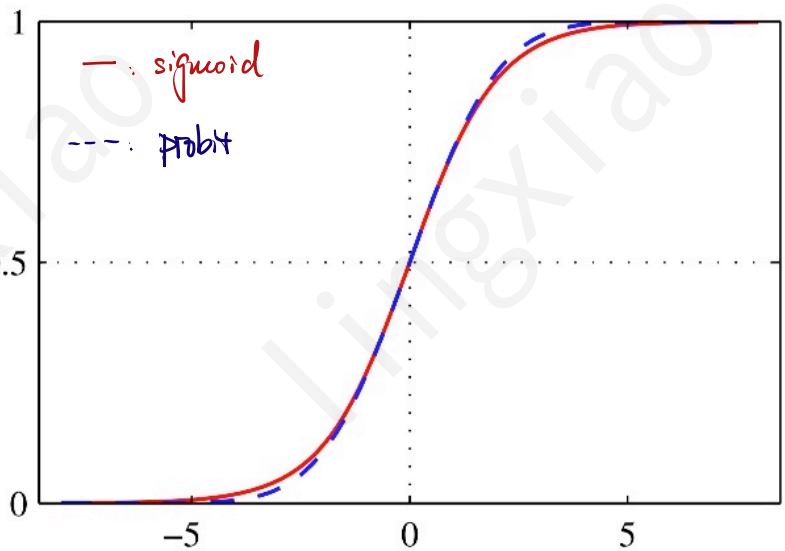
$$\frac{d\Phi}{da}\Big|_{a=0} = \lambda\, N(0|0,1)$$

$$= \frac{\lambda}{\sqrt{2\pi}}$$

let $\frac{\lambda}{\sqrt{2\pi}} = \frac{1}{4}$

$$\lambda = \frac{\sqrt{2\pi}}{4} \qquad \lambda^2 = \frac{\pi}{8}$$

we approximate $\sigma(a)$ by $\Phi(\lambda a)$.



With the probit function, we can show that:

$$\int \Phi(\lambda a)\, N(a|\mu,\sigma^2)\, da = \Phi\left(\frac{\mu}{(\lambda^{-2}+\sigma^2)^{\frac{1}{2}}}\right)$$

proof:

Statement: Suppose we have a random variable satisfied normal distribution denoted as $X \sim N(x|\mu,\sigma^2)$, the probability of $X \le x$ is $P\{X \le x\} = \Phi\left(\frac{x-\mu}{\sigma}\right)$

$$P\{X \le x\} = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right] dx$$

$$= \int_{-\infty}^{\frac{x-\mu}{\sigma}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}v^2\right\} \cdot \sigma\, dv$$

$$= \int_{-\infty}^{\frac{x-\mu}{\sigma}} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}v^2\right\} dv$$

$$= \Phi\left(\frac{x-\mu}{\sigma}\right)$$

For two random variables $X \sim N(0,\lambda^{-2})$ and $Y \sim N(\mu,\sigma^2)$:

$$P\{X \le Y \mid Y=a\} = P\{X \le a\} = \Phi\left(\frac{a-0}{\lambda^{-1}}\right) = \Phi(\lambda a)$$

$$P\{X \le Y\} = \int_{-\infty}^{+\infty} P\{X \le Y \mid Y=a\}\, P\{Y=a\}\, da$$

$$= \int_{-\infty}^{+\infty} \Phi(\lambda a)\, N(a|\mu, \sigma^2)\, da.$$

$\because$ $x - Y$ also follows a Gaussian

$\therefore$ $E[x - Y] = E[x] - E[Y] = 0 - \mu = -\mu$

$$Var(x - Y) = Var(x) + Var(Y) = \lambda^{-2} + \sigma^2$$

$\therefore$ $$P\{x - Y \le 0\} = \Phi\left\{\frac{0 - (-\mu)}{\sqrt{\lambda^{-2} + \sigma^2}}\right\} = \Phi\left(\frac{\mu}{\sqrt{\lambda^{-2} + \sigma^2}}\right)$$

Thus. $$\int \Phi(\lambda a)\, N(a|\mu, \sigma^2)\, da = \Phi\left(\frac{\mu}{\sqrt{\lambda^{-2} + \sigma^2}}\right)$$

Having approximated the convolution by the probit function, we can approximate the probit again by sigmoid:

$$\int \sigma(a)\, N(a|\mu, \sigma^2)\, da \simeq \sigma(k(\sigma^2)\mu)$$

$$k(\sigma^2) = \sqrt{1 + \frac{\pi\sigma^2}{8}}$$

Thus, the predictive distribution now becomes:

$$p\{C_1 | \phi, t\} = \sigma(k(\sigma_a^2)\mu_a)$$