Linear classification (线性分类)

Linear regression

$\downarrow$ connection

Linear classification

(Transform the linear function of **w** using a nonlinear function f(·), f(·) is known as an activation function, whereas its inverse is called a link function)

Y = $f(\boldsymbol{w^T X} + \boldsymbol{b}) \longrightarrow$

- $\boldsymbol{y} \in \{\mathbf{0},\mathbf{1}\}$
  - **Linear Discriminant Analysis (LDA 线性判别分析)**
  - **Perceptron (感知机)**
- $\boldsymbol{y} \in [\mathbf{0},\mathbf{1}]$
  - **Generative Model（生成式模型）：GDA/Naïve Bayes …**
  - **Discriminative Model (判别式模型)：logistic regression…**

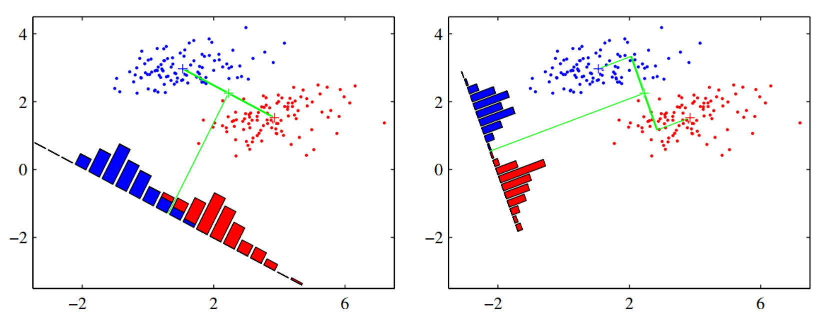
1. Linear Discriminant Analysis (LDA 线性判别分析)

Source(prml chapter4)

View a linear classification model is in terms of dimensionality reduction

Idea: give a large separation between the projected class means while also giving a small variance within each class

（类内距离小，类间距离大）



The second is better than the first one since the left one exists some overlaps between classes and the mean in the right one is also smaller.

$$X_{c1} = \{X_i \mid y_i = 1\} \quad X_{c2} = \{X_i \mid y_i = -1\}$$
$$|X_{c1}| = N_1, \ |X_{c2}| = N_2 \qquad N_1 + N_2 = N$$

$$\begin{aligned}
C_1:\ \bar{Z}_1 &= \frac{1}{N_1}\sum_{i=1}^{N_1} w^T X_i \\
S_1 &= \frac{1}{N_1}\sum_{i=1}^{N} (w^T X_i - \bar{Z}_1)(w^T X_i - \bar{Z}_1)^T \\
C_2:\ \bar{Z}_2 &= \frac{1}{N_2}\sum_{i=1}^{N_2} w^T X_i \\
S_2 &= \frac{1}{N_2}\sum_{i=1}^{N} (w^T X_i - \bar{Z}_2)(w^T X_i - \bar{Z}_2)^T
\end{aligned}$$

Model:

From the idea above , our goal function will be the ratio of the between-class variance to the within-class variance

$$J(w) = \frac{(\bar{Z}_1 - \bar{Z}_2)^2}{S_1 + S_2}$$
$$\hat{w} = \arg\max_{w} J(w)$$

the between-class variance = $(\bar{Z}_1 - \bar{Z}_2)^2$

the within-class variance = $S_1 + S_2$

$$\begin{aligned}
&(\bar{Z}_1 - \bar{Z}_2)^2 \\
&= \frac{1}{N_1}\sum_{i=1}^{N} w^T X_i - \frac{1}{N_2}\sum_{i=1}^{N_2} w^T X_i \\
&= w^T\left(\frac{1}{N_1}\sum_{i=1}^{N_1} X_i - \frac{1}{N_2}\sum_{i=1}^{N_2} X_i\right) \\
&= \{w^T(\bar{X}_{c_1} - \bar{X}_{c_2})\}^T
\end{aligned}$$

S_1+S_2

$= \frac{1}{N_1}\sum_{i=1}^{N}(w^T x_i-\overline{z}_1)(w^T x_i-\overline{z}_2)^T + \frac{1}{N_2}\sum_{i=1}^{N}(w^T x_i-\overline{z}_1)(w^T x_i-\overline{z}_2)^T$

$= \frac{1}{N_1}\sum_{i=1}^{N_1}(w^T x_i-\frac{1}{N_1}\sum_{j=1}^{N_1} w^T x_j)(w^T x_i-\frac{1}{N_1}\sum_{j=1}^{N_1} w^T x_j)^T + \frac{1}{N_2}\sum_{i=1}^{N_2}(w^T x_i-\frac{1}{N_2}\sum_{j=1}^{N_2} w^T x_j)(w^T x_i-\frac{1}{N_2}\sum_{j=1}^{N_2} w^T x_j)^T$

$= \frac{1}{N_1}\sum_{i=1}^{N_1} w^T(x_i-\underbrace{\frac{1}{N_1}\sum_{j=1}^{N_1} x_j}_{=\overline{x}_{c_1}})(x_i-\underbrace{\frac{1}{N_1}\sum_{j=1}^{N_2} x_j}_{=\overline{x}_{c_1}})^T w + \frac{1}{N_2}\sum_{i=1}^{N_2} w^T(x_i-\overline{x}_{c_2})(x_i-\overline{x}_{c_2})^T w$

$= w^T\underbrace{\{\frac{1}{N_1}\sum_{i=1}^{N_1}(x_i-\overline{x}_{c_1})(x_i-\overline{x}_{c_1})^T\}}_{\text{方差矩阵 } S} w + w^T\{\frac{1}{N_2}\sum_{i=1}^{N_2}(x_i-\overline{x}_{c_2})(x_i-\overline{x}_{c_2})^T\} w$

$= w^T S_{c_1} w + w^T S_{c_2} w$

$= w^T(S_{c_1}+S_{c_2})w$

$$\therefore \boxed{J(w) = \frac{w^T(\overline{x}_{c_1}-\overline{x}_{c_2})(\overline{x}_{c_1}-\overline{x}_{c_2})^T w}{w^T(S_{c_1}+S_{c_2})w}}$$

Let $S_b = (\overline{x}_{c_1}-\overline{x}_{c_2})(\overline{x}_{c_1}-\overline{x}_{c_2})^T$ (between class variance matrix) $p\times p$ matrix

$S_w = (S_{c_1}+S_{c_2})$ (within class variance matrix) $p\times p$ matrix

$$\boxed{J(w) = \frac{w^T S_b w}{w^T S_w w}}$$

$J(w) = (w^T S_b w)(w^T S_w w)^{-1}$

$\frac{dJ(w)}{dw} = 2S_b w(w^T S_w w)^{-1} + (w^T S_b w)\cdot(-1)(w^T S_w w)^{-2}(2S_w w) = 0$

$S_b w(w^T S_w w) - w^T S_b w(S_w \cdot w) = 0$

$w^T S_b w S_w \cdot w = S_b w \cdot \underbrace{(w^T S_w \cdot w)}_{\text{实数} \in \mathbb{R}}$

$S_w \cdot w = \frac{w^T S_w \cdot w}{w^T S_b \cdot w} \cdot S_b \cdot w$ 关心 w 方向而非大小

$w = \frac{w^T S_w \cdot w}{w^T S_b \cdot w} S_w^{-1} S_b \cdot w \quad \propto S_w^{-1} \underline{S_b \cdot w}$

$(\overline{x}_{c_1}-\overline{x}_{c_2})\underbrace{(\overline{x}_{c_1}-\overline{x}_{c_2})^T w}_{(1\times p)\cdot(p\times 1)\text{ 一维}}$

$\boxed{\propto S_w^{-1}(\overline{x}_{c_1}-\overline{x}_{c_2})}$

对角矩阵，各向同性 $S_w^{-1} \propto I$

进一步简化为 $w \propto (\overline{x}_{c_1}-\overline{x}_{c_2})$

2. Perceptron (感知机)

Idea: determine the parameters **w** of the perceptron can most be motivated by error function minimization.(错误驱动)

Model：

$$f(x) = sign(w^T x) \quad , x \in \mathbb{R}^p, w \in \mathbb{R}^p$$

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

Similar to the linear regression we want to minimize the loss function, use the idea above we know that if (xi,yi) is classified correctly:

$$\left.\begin{matrix} w^T x_i > 0 & y_i = 1 \\ w^T x_i < 0 & y_i = -1 \end{matrix}\right\} \Leftrightarrow y_i w^T x_i > 0$$

So for the misclassified samples, we want them to be as less as possible.

$$L(w) = \sum_{i=1}^{N} I\{y_i w^T x_i < 0\}$$

(the error is a piecewise constant function of **w** with discontinuities, thus hard to find gradient)

Therefore consider an alternative error function known as the ***perceptron criterion*.**

The new loss function becomes

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

$$\nabla_w L = -y_i x_i$$

Where D is the set that been separated wrong.

Then using the stochastic gradient descent, we got

$$w^{(t+1)} \leftarrow w^{(t)} - \lambda \nabla w L \quad (L: \text{learning rate})$$

$$w^{(t)} - \lambda y_i x_i$$

Note: for data sets that are not linearly separable, the perceptron learning algorithm will never converge.

3. Logistic regression (逻辑回归)

(Source: cs 229 note1 &PRML chapter 4)

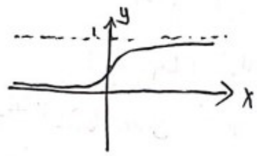
（1）binary classification

Logistic Regression

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

$\sigma : \mathbb{R} \mapsto (0,1)$

$w^T x \mapsto p$



$$\begin{cases} P(y=1 \mid x) = \sigma(w^T x) = \frac{1}{1+\exp\{-w^T x\}}, \quad y=1 & \text{let } P(y=1 \mid x) = \psi(x_i, w) \\ P(y=0 \mid x) = 1 - P(y=1 \mid x) = \frac{\exp\{-w^T x\}}{1+\exp\{-w^T x\}} \quad y=0 & \text{let } P(y=0 \mid x) = \log(1-\psi(x_i, w)) \end{cases}$$

$$\Rightarrow P(y \mid X) = P_1^{y} P_0^{1-y}$$

$$\begin{aligned} \text{MLE } \hat{w} &= \arg\max \log P(Y \mid X) \\ &= \arg\max \log \prod_{i=1}^{N} P(Y_i \mid X_i) \\ &= \arg\max \sum_{i=1}^{N} \log P(Y_i \mid X_i) \\ &= \arg\max \underbrace{\sum_{i=1}^{N} y_i \log \psi(x_i, w) + (1-y_i) \log(1-\psi(x_i, w))}_{-\text{cross entropy}} \end{aligned}$$

MLE (max) $\Rightarrow$ loss function (min cross entropy)

Maximize the likelihood：

$$P_1' = \left(\frac{1}{1+\exp(-w^T x)}\right)' = P_1(1-P_1)$$

$$\frac{\partial}{\partial w} \log P(Y \mid X) = \sum_{i=1}^{N} (y_i - P_1) X_i$$

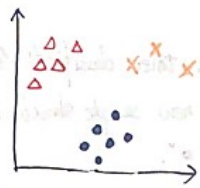
Using the stochastic gradient descent rule:

$$w^{(t+1)} \leftarrow w^{(t)} + \lambda \frac{\partial}{\partial w} \log P(Y \mid X)$$

(λ: learning rate)

（2）multi-class classification

(1) One - Vs - all

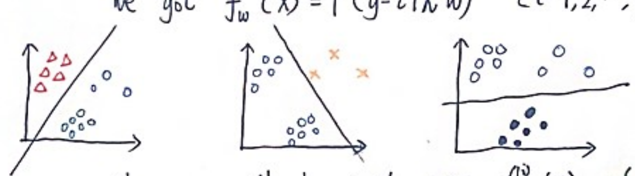


思想 idea: trans one multi-classification problem to many binary-classification problem.

Choose one of the classes and labeled as "y = 1", the others are all labeled as "y=0"
then similarly, choose one of the classes and labeled as "y = 2", the others are all labeled as "y=0"

⋮

we got $f_w^{(i)}(x) = P(y=i|x, w)$ $(i = 1, 2, \cdots, k)$ k: # of classes



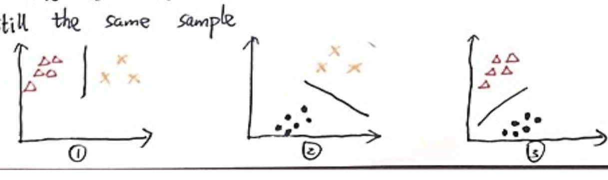
then we will choose the max $f_w^{(i)}(x)$ of sample x that represent x belongs to the ith class.

advantages: big range to use

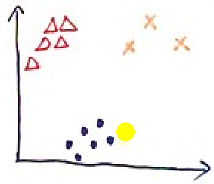
disadvantages: the binary classification is unbalanced.

(2) One - Vs - One

still the same sample



So for a new sample (the yellow point)



the ① first classification model will considered it as "△"
② second - - - - - - - "•"
③ third - - - - - - - "•"

after three classification's vote, the new sample should be a "•".

(3) softmax

$$P(C_k|x) = \frac{P(X|C_k)P(C_k)}{\Sigma_j P(X|C_k)P(C_k)}$$

$P(X|C_k)$: class-conditional densities

$P(C_k)$: priors

$$= \frac{\exp(a_k)}{\Sigma_j \exp(a_j)}$$

(normalized exponential / softmax)

4. Gaussian Discriminant Analysis(GDA 高斯判别分析)

（source： http://cs229.stanford.edu/notes/cs229-notes2.pdf）

Idea： judge p(y=0|x) and p(y=1|x) which is bigger.

For binary classification

$$\begin{aligned} y &\sim \text{Bernoulli}(\phi) \\ x|y=0 &\sim \mathcal{N}(\mu_0, \Sigma) \\ x|y=1 &\sim \mathcal{N}(\mu_1, \Sigma) \end{aligned}$$

$$\begin{aligned} p(y) &= \phi^y(1-\phi)^{1-y} \\ p(x|y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right) \\ p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right) \end{aligned}$$

log-likelihood : $\log(\theta) = \log \prod_{i=1}^{N} P(X_i, y_i)$ ← 联合概率

$\theta = (\mu_1, \mu_2, \Sigma, \phi)$

$\hat{\theta} = \underset{\theta}{\arg\max}\, l(\theta)$

$$
\begin{aligned}
&= \sum_{i=1}^{N} \log\{P(X_i|y_i)\, P(y_i)\} \\
&= \sum_{i=1}^{N} \{\log P(X_i|y_i) + \log P(y_i)\} \\
&= \sum_{i=1}^{N} \{\log N(\mu_1, \Sigma)^{y_i} \cdot N(\mu_2, \Sigma)^{1-y_i} + \log \phi^{y_i}(1-\phi)^{1-y_i}\} \\
&= \sum_{i=1}^{N} \{\underbrace{\log N(\mu_1, \Sigma)^{y_i}}_{①} + \underbrace{\log N(\mu_2, \Sigma)^{1-y_i}}_{②} + \underbrace{\log \phi^{y_i}(1-\phi)^{1-y_i}}_{③}\}
\end{aligned}
$$

求ϕ：③ $\sum_{i=1}^{N} \log \phi^{y_i}(1-\phi)^{1-y_i}$

$$
\frac{d③}{d\phi} = \sum_{i=1}^{N} \{y_i \frac{1}{\phi} - (1-y_i)\frac{1}{1-\phi}\} = 0
$$

$$
\Rightarrow \sum_{i=1}^{N} y_i(1-\phi) - \phi(1-y_i) = 0
$$

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

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

$$
\therefore \boxed{\hat{\phi} = \frac{1}{N}\sum_{i=1}^{N} y_i = \frac{1}{N_1}\sum_{i=1}^{N_1} y_i + \underbrace{\frac{1}{N_0}\sum_{i=1}^{N_2} y_i}_{y_i=0} = \frac{1}{N_1}\sum_{i=1}^{N_1} y_i = \frac{N_1}{N}}
$$

求$\hat{\mu}_1$：① $\sum_{i=1}^{N} \log N(\mu_1, \Sigma)^{y_i}$

$$
= \sum_{i=1}^{N} y_i \log\left(\frac{1}{(2\pi)^{p/2}|\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(X_i-\mu_1)^T \Sigma^{-1}(X_i-\mu_1)\}\right)
$$

可去掉：与Σ无关只与μ有关

$$
\mu_1 = \underset{\mu_1}{\arg\max}\, ① = \arg\max \sum_{i=1}^{N} y_i(-\frac{1}{2}(X_i-\mu_1)^T \Sigma^{-1}(X_i-\mu_1))
$$

$$
\begin{aligned}
&\sum_{i=1}^{N} y_i(-\frac{1}{2}(X_i-\mu_1)^T \Sigma^{-1}(X_i-\mu_1)) \\
&= -\frac{1}{2}\sum_{i=1}^{N} y_i(X_i^T\Sigma^{-1} - \mu_1^T\Sigma^{-1})(X_i-\mu_1) \\
&= -\frac{1}{2}\sum_{i=1}^{N} y_i(\underbrace{X_i^T\Sigma^{-1}X_i}_{constant} - \underbrace{X_i^T\Sigma^{-1}\mu_1}_{\substack{(1\times p)(p\times p)(p\times 1) \\ \in \mathbb{R}}} - \underbrace{\mu_1^T\Sigma^{-1}X_i}_{\substack{(1\times p)(p\times p)(p\times 1) \\ \in \mathbb{R}}} + \mu_1^T\Sigma^{-1}\mu_1) \\
&= -\frac{1}{2}\sum_{i=1}^{N} y_i(X_i^T\Sigma^{-1}X_i - 2X_i^T\Sigma^{-1}\mu_1 + \mu_1^T\Sigma\mu_1)
\end{aligned}
$$

$$
\begin{aligned}
\frac{d\Delta}{d\mu} &= -\frac{1}{2}\sum_{i=1}^{N} y_i(-2\Sigma^{-1}X_i + 2\Sigma^{-1}\mu_1) = 0 \\
&= \sum_{i=1}^{N} y_i(\Sigma^{-1}\mu_1 - \Sigma^{-1}X_i) = 0 \\
&= \sum_{i=1}^{N} y_i(\mu_1 - X_i) = 0 \\
&= \sum_{i=1}^{N} (y_i\mu_1 - y_iX_i) = 0
\end{aligned}
$$

$$
\boxed{\hat{\mu}_1 = \frac{\sum_{i=1}^{N} y_i x_i}{\sum_{i=1}^{N} y_i} = \frac{1}{N_1}\sum_{i=1}^{N} y_i x_i}
$$

求 μ_2：② $\sum_{i=1}^{N} \log N(\mu_2, \Sigma)^{1-y_i}$

$$= \sum_{i=1}^{N} (1-y_i) \log \left(\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(x_i-\mu_2)^T \Sigma^{-1}(x_i-\mu_2)\}\right)$$

$$\hat{\mu}_2 = \underset{\mu_2}{\arg\max}\ ② = \underset{\mu_2}{\arg\max} \sum_{i=1}^{N} (1-y_i) \left(-\frac{1}{2}(x_i-\mu_2)^T \Sigma^{-1}(x_i-\mu_2)\right)$$

Similarly $\frac{\partial \Delta}{\partial \mu_2} = -\frac{1}{2}\sum_{i=1}^{N}(1-y_i)\,(-2\Sigma^{-1}x_i + 2\Sigma^{-1}\mu_2) = 0$

$$\sum_{i=1}^{N}(1-y_i)(\mu_2 - x_i) = 0$$

$$\sum_{i=1}^{N}(1-y_i)\mu_2 - (1-y_i)x_i = 0$$

$$\boxed{\hat{\mu}_2 = \frac{\sum_{i=1}^{N}(1-y_i)x_i}{\sum_{i=1}^{N}(1-y_i)} = \frac{1}{N_2}\sum_{i=1}^{N}(1-y_i)x_i}$$

求 Σ　$\hat{\Sigma} = \underset{\Sigma}{\arg\max}\ ①+②$

let $C_1 = \{x_i \mid y_i=1,\ i=1,2,\cdots,N\}$　$|C_1| = N_1$

$C_2 = \{x_i \mid y_i=0,\ i=1,2,\cdots,N\}$　$|C_2| = N_2$

thus $①+② = \sum_{x_i\in C_1} \log N(\mu_1,\Sigma) + \sum_{x_i \in C_2} \log N(\mu_2, \Sigma)$

$$\sum_{i=1}^{N}\log N(\mu,\Sigma) = \sum_{i=1}^{N}\log \frac{1}{(2\pi)^{p/2}|\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\}$$

$$= \sum_{i=1}^{N}\left(\log\frac{1}{(2\pi)^{p/2}} + \log|\Sigma|^{-\frac{1}{2}} - \frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right)$$

$$= \sum_{i=1}^{N}\left[C - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right]$$

$$= \tilde{C} - \frac{1}{2}\sum_{i=1}^{N}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$$

$$= \tilde{C} - \frac{1}{2}N\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$$

$x_i \in p\times 1$

$(x_i-\mu)^T \in 1\times p$

$(1\times p)(p\times p)(p\times 1) \in \mathbb{R}$

for real number this equals to it's trace

$\frac{\partial\, tr(AB)}{\partial A} = B^T$

$\frac{\partial |A|}{\partial A} = |A|\cdot A^{-1}$

$tr(AB) = tr(BA)$

$tr(ABC) = tr(CAB) = tr(BCA)$

for $-\frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$

$$= -\frac{1}{2}\sum_{i=1}^{N} tr\{(x_i-\mu)(x_i-\mu)^T\Sigma^{-1}\}$$

$$= -\frac{1}{2}\, tr\{\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T\Sigma^{-1}\}$$

$$= -\frac{1}{2}N\, tr(S\Sigma^{-1})$$

$$= \tilde{C} - \frac{1}{2}N\log|\Sigma| - \frac{1}{2}N\, tr(S\Sigma^{-1})$$

thus $①+② = -\frac{1}{2}N_1\log|\Sigma| - \frac{1}{2}N_1 tr(S_1\Sigma^{-1}) - \left(\frac{1}{2}N_2\log|\Sigma| + \frac{1}{2}N_2 tr(S_2\Sigma^{-1})\right) + C$

$$= -\frac{1}{2}N\log|\Sigma| - \frac{1}{2}N_1 tr(S_1\Sigma^{-1}) - \frac{1}{2}N_2 tr(S_2\Sigma^{-1}) + C.$$

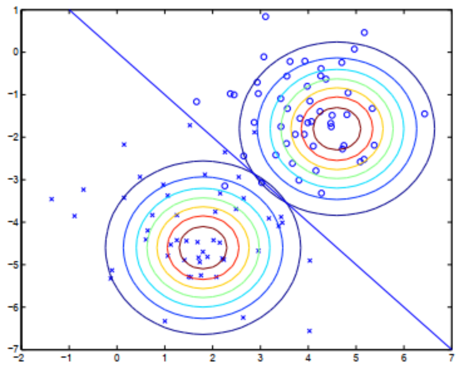
$$\frac{\partial(①+②)}{\partial\Sigma} = -\frac{1}{2}N\frac{1}{|\Sigma|}|\Sigma|\cdot\Sigma^{-1} - \frac{1}{2}N_1 S_1^T\cdot(-1)\Sigma^{-2} - \frac{1}{2}N_2 S_2^T\cdot(-1)\Sigma^{-2})$$

$$= -\frac{1}{2}N\Sigma^{-1} - \frac{1}{2}N_1 S_1\Sigma^{-2} - \frac{1}{2}N_2 S_2\Sigma^{-2} = 0$$

$$= N\Sigma - N_1S_1 - N_2S_2 = 0$$

$$\boxed{\hat{\Sigma} = \frac{1}{N}(N_1S_1 + N_2S_2)}$$

Pictorially, what the algorithm is doing can be seen in as follows:



In the figure is the straight line giving the decision boundary at which p(y = 1|x) = 0.5.

Since we already calculate all parameters we need ,we can calculate p(y = 1|x) and p(y = 0|x).

If p(y = 1|x) > p(y = 0|x), we'll predict y = 1 to be the most likely outcome, otherwise, we'll predict y = 0.

Discussion: GDA and logistic regression

(1)

If p(x|y) is multivariate Gaussian (with shared Σ), then p(y|x) necessarily follows a logistic function. The converse, however, is not true. GDA makes stronger modeling assumptions, and is more data efficient.

(2)

Logistic regression is also more robust and less sensitive to incorrect modeling assumptions.

5. Naïve Bayes (朴素贝叶斯)

Source (cs229 note2)

(1) **Naive Bayes (NB) assumption**: xi's are conditionally independent given y.

$$P(x|y) = \prod_{j=1}^{n} p(x_j|y)$$

(2) **Procedure:**

The idea is the find the argmax posterior probability

Data : $\{(x_i, y_i)\}_{i=1}^{N}$

$x_i \in \mathbb{R}^p$, y_i={1,2,...,k}

$$\hat{y} = \arg\max_y P(y|x) \qquad P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(x|y)\cdot P(y)}{P(x)} \propto P(y)\,P(x|y)$$

$$= \arg\max_y P(y)\,P(x|y) \longrightarrow \text{MLE 求解}$$

Step1: calculate the prior probability and condition probability

$$P(Y=c_k) = \frac{\sum_{i=1}^{N} I(y_i = c_k)}{N}, \quad k=1,2,\cdots,K$$

$$P(X^{(j)} = a_{jl}|Y=c_k) = \frac{\sum_{i=1}^{N} I(x_i^{(j)} = a_{jl}, y_i = c_k)}{\sum_{i=1}^{N} I(y_i = c_k)}$$

$$j=1,2,\cdots,n; \quad l=1,2,\cdots,S_j; \quad k=1,2,\cdots,K$$

Step2: calculate and decide the category of x(j)

$$P(Y=c_k)\prod_{j=1}^{n} P(X^{(j)} = x^{(j)}|Y=c_k), \quad k=1,2,\cdots,K$$

$$y = \arg\max_{c_k} P(Y=c_k)\prod_{j} P(X^{(j)} = x^{(j)}|Y=c_k)$$

(3) Bayes estimation

Since that the probability might be 0,we need to use the Bayes estimation. The formula looks like this:

$$P_\lambda(X^{(j)} = a_{jl} | Y = c_k) = \frac{\sum_{i=1}^{N} I(x_i^{(j)} = a_{jl}, y_i = c_k) + \lambda}{\sum_{i=1}^{N} I(y_i = c_k) + S_j \lambda} \qquad \lambda \geqslant 0$$

$$P_\lambda(Y = c_k) = \frac{\sum_{i=1}^{N} I(y_i = c_k) + \lambda}{N + K\lambda}$$

When $\lambda = 1$, we have Laplace smoothing

When $\lambda = 0$, we have MLE