Week6 - Probabilistic view of classification: Discriminative v.s. Generative classifiers

1. Probabilistic Generative classifiers

Generative approach model the the class-conditional densities $p(\boldsymbol{x}|C_k)$ and the class prior $p(C_k)$, then use these to compute posterior probabilities $p(C_k|\boldsymbol{x})$ through Bayes' theorem.

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k) \cdot p(C_k)}{p(\boldsymbol{x})}$$

Note that If we calculate $p(C_k|\boldsymbol{x})$ in order to make prediction, we don't need to actually calculate the denominator $p(\boldsymbol{x})$.

$$\arg\max_{C_k} p(C_k|\boldsymbol{x}) = \arg\max_{C_k} p(C_k|\boldsymbol{w}) = \frac{p(\boldsymbol{x}|C_k) \cdot p(C_k)}{p(\boldsymbol{x})} = \arg\max_{C_k} p(\boldsymbol{x}|C_k) \cdot p(C_k)$$

Hence, the model used for classification problem via Bayes' rule is also call **Bayes classifier**.

Class-Conditional Probability Density Function (PDF)

Class-Conditional Probability Density Function (PDF) is the probability of x given that the state of nature of C_k, which indicates how much probability will be for $\boldsymbol{x}$ belonging to class C_k .

Univariate Inputs (Binary class problem)

- **Assumption**: Generative learning model assuming that class-conditional PDF $p(x|C_k)$ is distributed according to a **normal (Gaussian) distribution**. (**Gaussian Bayes classifier**)

$$p(x|C_k) = \frac{1}{\sigma_{C_k}\sqrt{2\pi}} exp\left(-\frac{(x-\mu_{C_k})^2}{2\sigma_{C_k}^2}\right)$$

where parameters: $\mu_{C_k}, \sigma_{C_k}^2$ is the distribution mean and variance of class C_k

- Step by step
    - Estimate the model parameters $\{(\mu_{C_1}, \sigma_{C_1}^2), (\mu_{C_2}, \sigma_{C_2}^2)\}$
        - Divide the training set D into two class: D_1 for class C_1 and D_2 for class C_2
        - For each class C_k, we fit a Gaussian to model $p(x|C_k)$ on class C_k.
    - Using **Maximum Likelihood Estimation (MLE)** for a Gaussian

    $$p(\boldsymbol{x}|C_k) = \prod_{n=1}^{N} p(\boldsymbol{x}_n)|C_k = \prod_{n=1}^{N} \left\{\frac{1}{\sigma_{C_k}\sqrt{2\pi}} exp\left(-\frac{(x-\mu_{C_k})^2}{2\sigma_{C_k}^2}\right)\right\}$$

        - Minimise the negative logarithm of the class-conditional PDF:

        $$E(\boldsymbol{x}) = -\ln p(\boldsymbol{x}|C_k) = -\left(\sum_{n=1}^{N}\left\{\frac{1}{\sigma_{C_k}\sqrt{2\pi}}\right\} + \sum_{n=1}^{N}\left\{-\frac{(x-\mu_{C_k})^2}{2\sigma_{C_k}^2}\right\}\right)$$

$\Rightarrow \sum_{n=1}^{N} ln(\sigma_{C_k}\sqrt{2\pi}) + \sum_{n=1}^{N} \left\{ \frac{(x-\mu_{C_k})^2}{2\sigma_{C_k}^2} \right\}$

Partial differentiate with respect to μ_{C_k} and $\sigma_{C_k}^2$, respectively. let the derivates equal to zero:

$\frac{\partial(-\ln p(x|C_k))}{\partial \mu_{C_k}} = 0 \Rightarrow \mu_{C_k} = \frac{1}{N}\sum_{n=1}^{N} x_n$

$\frac{\partial(-\ln p(x|C_k))}{\partial \sigma_{C_k}^2} = 0 \Rightarrow \sigma_{C_k}^2 = \frac{1}{N}\sum_{n=1}^{N} (x_n - \mu_{C_k})^2$

- Note: $p(x|C_k) \approx p(x|\mu_{C_k}, \sigma_{C_k}^2)$

- **Prediction rule**: $p(C_1|x) > p(C_2|x) \Rightarrow p(x|C_1)p(C_1) > p(x|C_2)p(C_2)$
  - Class prior $p(C_k)$:

    $p = \frac{\sum_{n=1}^{N} 1}{N}$ (If $t_n = C_k$)

    $p(C) = \begin{cases} p & C = C_k \\ 1-p & otherwise \end{cases}$

    using Bernoulli distribution: $p(C_k) = p^{C_k}(1-p)^{1-C_k}$
- Summary:
  - Given the training set D , learn the parameters to fully specify the joint distribution $p(\boldsymbol{x}, C_k)$
  - Model parameters: $\boldsymbol{w} = (p, \{\mu_{C_k}\}, \{\sigma_{C_k}^2\})$
  - Likelihood function of the joint distribution for each class C_k

    $\prod_{n=1}^{N} p(x_n, \{C_k, \mu_{C_k}, \sigma_{C_k}^2, p\}) = \prod_{n=1}^{N} p(x_n|C_k, \mu_{C_k}, \sigma_{C_k}^2)p(C_k, p)$
    - $p(x_n|C_k, \mu_{C_k}, \sigma_{C_k}^2)$: use MLE to find μ_{C_k} and $\sigma_{C_k}^2$, then calculate
    - $p(C_k, p)$: count the data point to find p, then use Bernoulli distribution to find $p(C_k)$

Multivariate Inputs (Binary class problem)

- **Assumption**: assume that class-conditional PDF $p(\boldsymbol{x}|C_k)$ is distributed according to a **Multivariate normal (Gaussian) distribution**.

$p(\boldsymbol{x}|C_k) \approx p(\boldsymbol{x}|\boldsymbol{\mu_{C_k}}, \boldsymbol{\Sigma})\frac{1}{|\sigma|^{1/2}(2\pi)^{D/2}} exp\left(-\frac{1}{2}(x-\boldsymbol{\mu_{C_k}})^T \cdot |\Sigma|^{-1} \cdot (x-\boldsymbol{\mu_{C_k}})\right)$

  - $\boldsymbol{x}$: a vector-valued random variable $\boldsymbol{x} = (x_1, \ldots, x_D)^T$. ($D \times 1$ column vector)
  - $\boldsymbol{\mu_{C_k}}$: D-dimensional mean vector ($D \times 1$) for class C_k
  - $\boldsymbol{\Sigma}$: $D \times D$ covariance matrix, $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$.
  - Each class has a different $\boldsymbol{\mu_{C_k}}$, but all share the same $\boldsymbol{\Sigma}$.
- **Logistic sigmoid**: a 'squashing function' maos the whole real axis into a finite interval.

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

we can make use of the property of logistic sigmoid to model the posterior probability for class C_k.

- For **real-valued** x:

$$p(C_1|x) = \frac{p(x|C_1) \cdot p(C_1)}{p(x|C_1) \cdot p(C_1) + p(x|C_2) \cdot p(C_2)} = \frac{1}{1 + exp(-a)}$$

where:

$$a = \ln \frac{p(x|C_1) \cdot p(C_1)}{p(x|C_2) \cdot p(C_2)} \text{ (\textbf{the log odds})}$$

- Then, **for vector $\boldsymbol{x}$** :

$$p(C_1|\boldsymbol{x}) = \frac{1}{1 + exp(-a)} = \sigma(\boldsymbol{w} \cdot \boldsymbol{x} + w_0)$$

where:

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_{C_1} - \boldsymbol{\mu}_{C_2})$$

$$w_0 = -\tfrac{1}{2}\boldsymbol{\mu}_{C_1}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_{C_1} + \tfrac{1}{2}\boldsymbol{\mu}_{C_2}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_{C_2} + \ln \frac{p(C_1)}{pC_2}$$

- Step by step:

  Given training example $\{\boldsymbol{x}_n, t_n\}_{n=1,...,N}$ with $t_n \in \{0, 1\}$. (t_n=1 if $\boldsymbol{x}_n$ is in class C_1)

  - Calculate class prior:

    $$p = \frac{N_1}{N_1 + N_2}$$

  - Calculate the mean vector:

    $\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \boldsymbol{x}_n$

    $\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \boldsymbol{x}_n$

  - Calculate the variance vector:

    $\boldsymbol{S}_1 = \frac{1}{N_1} \sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T (\boldsymbol{x}_n - \boldsymbol{\mu}_1)$

    $\boldsymbol{S}_2 = \frac{1}{N_2} \sum_{n \in C_2} (\boldsymbol{x}_n - \boldsymbol{\mu}_2)^T (\boldsymbol{x}_n - \boldsymbol{\mu}_2)$

  - The covariance:

    $\boldsymbol{\Sigma} = \frac{N_1}{N} \boldsymbol{S}_1 + \frac{N_2}{N} \boldsymbol{S}_2$

  - Plug the Gaussian class densities into the variable a

    $a = \boldsymbol{w}\boldsymbol{x} + w_0$

    $$\Rightarrow -\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_2) + \tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_1) + \ln \frac{p(C_1)}{pC_2}$$

    > Note that a **takes a simple linear form**, which means the induced **decision boundary is linear**.

- Summary

  - Likelihood function of the joint distribution for all the data points: Bernoulli distribution

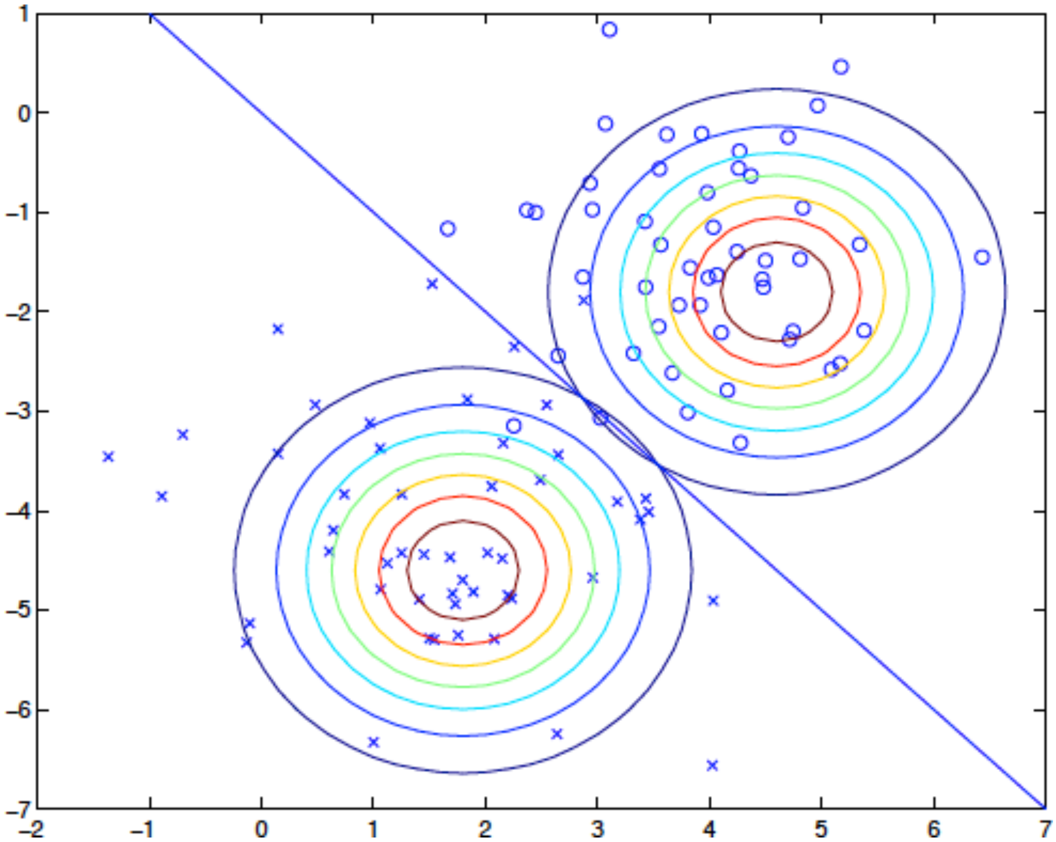
$\prod_{n=1}^{N}[p\cdot p(\boldsymbol{x}_n|\boldsymbol{\mu}_1,\boldsymbol{\Sigma})]^{t_n}[(1-p)\cdot p(\boldsymbol{x}_n|\boldsymbol{\mu}_2,\boldsymbol{\Sigma})]^{1-t_n}$

- Error function: the log-likelihood function

  $\ell(x_n,\mu_1,\mu_2,\boldsymbol{\Sigma})=\log\{likelihood\}$

  $\Rightarrow\sum_{n=1}^{N}\{t_n\cdot\ln[p\cdot p(\boldsymbol{x}_n|\boldsymbol{\mu}_1,\boldsymbol{\Sigma})]^{t_n}+(1-t_n)\cdot\ln[(1-p)\cdot p(\boldsymbol{x}_n|\boldsymbol{\mu}_2,\boldsymbol{\Sigma})]^{1-t_n}\}$

- **Visualization**:



- **Prediction Rule**:
  - Compare the posterior: $p(C_1|\boldsymbol{x})>p(C_2|\boldsymbol{x})\Rightarrow p(\boldsymbol{x}|C_1)p(C_1)>p(\boldsymbol{x}|C_2)p(C_2)$
  - Or: $a=\ln\dfrac{p(\boldsymbol{x}|C_1)\cdot p(C_1)}{p(\boldsymbol{x}|C_2)\cdot p(C_2)}$, If $a>0$ then predict class C_1, C_2 otherwise.

---

2. Probabilistic Discriminative Model: Logistic Regression

- Directly model the prediction of target variable y on input x as a conditional probability $p(y|x)$.
- Compare with non-probabilistic linear regression
  - Map input to a continuous target value: $\boldsymbol{w}\cdot\boldsymbol{x}$
  - The continuous target value is constrained to $[0,1]$ by applying logistic function as the activation function: $\sigma(\boldsymbol{w}\cdot\boldsymbol{x})$
- Error function:

  $\ell(\boldsymbol{w})=\log\{likelihood\}=\log\prod_{n=1}^{N}y_n^{t_n}(1-y_n)^{1-t_n}$

  where:
  - $y(\boldsymbol{x})=p(C_1|\boldsymbol{x})=\sigma(\boldsymbol{w}\cdot\boldsymbol{x})$

- Optimising the likelihood by using MLE:

  $\nabla\ell(\boldsymbol{w})=\dfrac{\partial\ell(\boldsymbol{w})}{\partial\boldsymbol{w}}=0\Rightarrow\sum_{n=1}^{N}(t_n-\sigma(\boldsymbol{w}\cdot\boldsymbol{x}))\boldsymbol{x}=0$

  No analytical solution $\rightarrow$ **Iterative algorithm**

- Gradient of the error function of $\boldsymbol{w}_n$:

  $\ell(\boldsymbol{w}_n)=\log\{likelihood\}=\log y_n^{t_n}(1-y_n)^{1-t_n}=t_n\log y_n+(1-t_n)\log(1-y_n)$

  (1) $\dfrac{\partial\ell(\boldsymbol{w}_n)}{\partial\boldsymbol{w}}=\dfrac{t_n}{y_n}\cdot\dfrac{\partial y_n}{\partial\boldsymbol{w}}+\dfrac{1-t_n}{1-y_n}\cdot\dfrac{-\partial y_n}{\partial\boldsymbol{w}}$

  $\Rightarrow\dfrac{\partial y_n}{\partial\boldsymbol{w}}\cdot(\dfrac{t_n}{y_n}-\dfrac{1-t_n}{1-y_n})=\dfrac{\partial y_n}{\partial\boldsymbol{w}}\cdot\dfrac{t_n-y_n}{y_n(1-y_n)}$

  (2) let $u=-\boldsymbol{w}^T\boldsymbol{x}$, $y_n =\cfrac{1}{1+exp(-\pmb{w}^T\pmb{x})}=\cfrac{1}{1+exp(u)}$

  $\cfrac{\partial{y_n}}{\partial{\pmb{w}}}=\cfrac{\partial{y_n}}{\partial{u}}\cdot\cfrac{\partial{u}}{\partial{\pmb{w}}}=\cfrac{\partial{y_n}}{\partial{u}}\cdot (-\pmb{x})$

  (3) $\cfrac{\partial{y_n}}{\partial{u}}=\cfrac{\partial\left\{\cfrac{1}{1+exp(u)}\right\}}{\partial{u}}=\cfrac{exp(u)}{[1+exp(u)]^2} =\cfrac{1}{1+exp(u)}\cdot \cfrac{exp(u)}{1+exp(u)} = y_n \cdot (1-y_n)$

  Hence:

  $\cfrac{\partial{\ell(\pmb{w}_n)}}{\partial \pmb{w}}=\cfrac{\partial{y_n}}{\partial{\pmb{w}}}\cdot \cfrac{t_n-y_n}{y_n(1-y_n)}=\cfrac{\partial{y_n}}{\partial{u}}\cdot (-\pmb{x})\cdot \cfrac{t_n-y_n}{y_n(1-y_n)}$

  $=y_n\cdot(1-y_n)\cdot(-\boldsymbol{x})\cdot\dfrac{t_n-y_n}{y_n(1-y_n)}$

  $=-(t_n-y_n)\cdot\boldsymbol{x}$

- Using Gradient descent: **SGD**

  $\nabla\ell(\boldsymbol{w})=-\eta(t_n-y(\boldsymbol{w}))\boldsymbol{x}_n$

  $\boldsymbol{w}^{(t+1)}:=\boldsymbol{w}^{(t)}+\eta^{(t)}(t_n-y_n)\boldsymbol{x}_n$

  Pseudocode:

```
Initialise the parameters to zero
While {E(w_new)-E(w_old)} > epsilon
do {
    for each training data point {x_n, t_n}
        update w
}
```

- Advantages
  - Quick to train and Faster than probabilistic generative model in classification with Good accuracy.
  - Resistant to overfitting
  - Model parameters can be used as indicators for feature importance