Probabilistic Classifiers

Procheta Sen

UNIVERSITY OF LIVERPOOL

Probabilistic vs "ordinary" classifier

- **"Ordinary" classifier** is a function f that assigns to an input object $\bar{X}$ a predicted class c from a fixed set of classes $\{c_1, c_2, \ldots, c_k\}$, i.e.

$$c = f(\bar{X}).$$

- **Probabilistic classifier** is a conditional distribution $P(C\,|\,\bar{X})$. For an input object $\bar{X}$ it gives probabilities $p_1, p_2, \ldots, p_k$, where

$$p_i = P(c_i \mid \bar{X})$$

and $p_1 + p_2 + \ldots + p_k = 1$.

Two types of models: discriminative vs generative

Discriminative

- Assume that the conditional distribution $P(C|X)$ (i.e. the probabilistic classifier) has specific form $P_\theta(C|X)$ depending on some parameters $\theta = (\theta_1, \ldots, \theta_k)$
- Use training data set to **find** / **learn** parameters $\theta_1, \ldots, \theta_k$ such that the resulting distribution is "best possible" among all distributions of the assumed form

Generative

- Assume that data come from specific distribution $P_\theta(X, C)$ depending on some parameters $\theta = (\theta_1, \ldots, \theta_k)$
- Use training data set to **find** / **learn** parameters $\theta_1, \ldots, \theta_k$ such that the resulting distribution is "best possible" among all distributions of the assumed form
- Use $P_\theta(X, C)$ to classify new objects

Generative models

Data generating distribution

- **Assumption**: data come from some unknown probability distribution P over object-class pairs $(\overline{X}, c) \in \mathscr{X} \times \mathscr{C}$, i.e. the classification problem is characterised by P.

- **Suppose we know** P: for any pair $(\overline{X}, c) \in \mathscr{X} \times \mathscr{C}$ we can compute $P(\overline{X}, c)$, i.e., the probability of the event that an object with feature vector $\overline{X}$ belongs to class c.

Then we can compute the Bayes optimal classifier

$$f^*(\overline{X}) = \arg\max_{c \in \mathscr{C}} P(\overline{X}, c)$$

Bayes optimal classifier

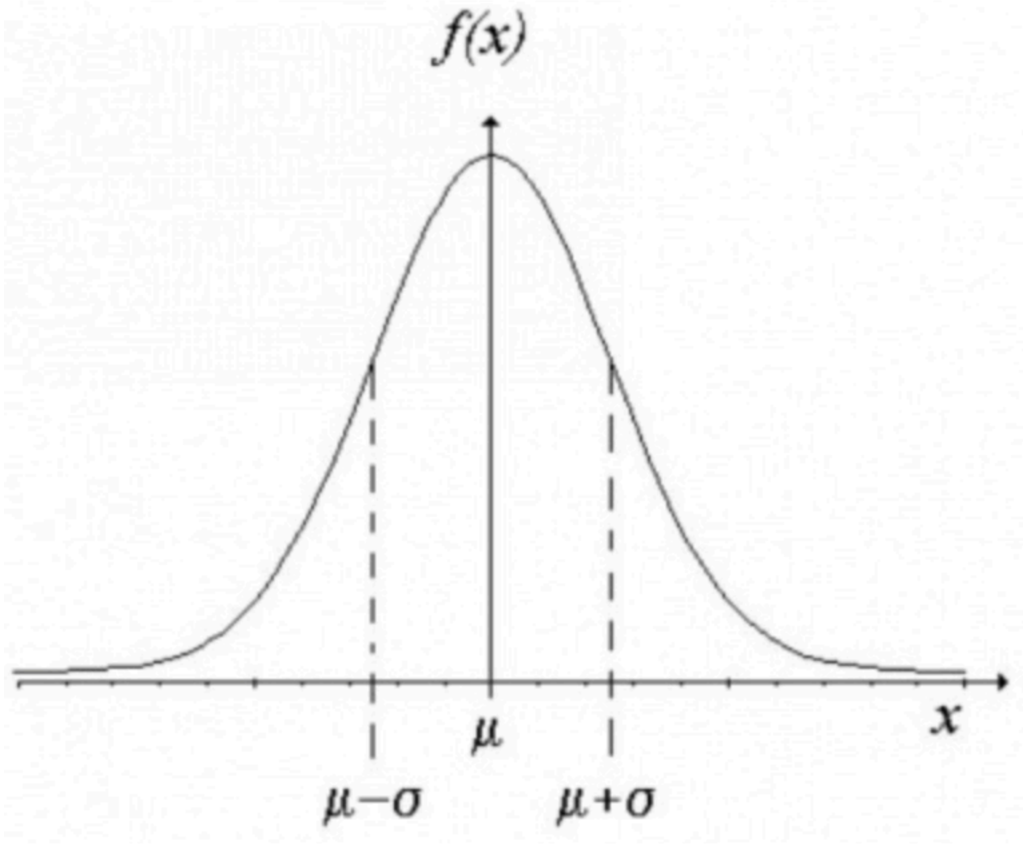
The Bayes optimal classifier

$$f^*(\overline{X}) = \arg\max_{c \in \mathscr{C}} P(\overline{X}, c)$$

The Bayes optimal classifier is best possible, in the sense that

if g is another classifier, then for any $\overline{X} \in \mathscr{X}$
the probability that f^* errs on $\overline{X}$ is smaller than the probability that g errs on $\overline{X}$

In practice…



- We do not now the data generating distribution P
- Instead, using training data, we can try to learn a distribution $\hat{P}$, which we hope to be very similar to P
- And then use $\hat{P}$ for classification

Normal distribution $\mathcal{N}(\mu, \sigma)$ is **parametric**: the two parameters are mean (μ) and standard deviation (σ)

One of the ways to construct $\hat{P}$

- Assume that the distribution P belongs (or is “close” to) some family of parametric distributions (e.g. Normal distribution) (the **model assumption**)
- **Estimate parameters** which give specific distribution in the family that is most likely to be the generating distribution for the training data
- Key assumption: the training data is drawn from P independently, i.e. the training instances are $(\overline{X}, c)$ are **independently and identically distributed** (the **i.i.d. assumption**)

Estimation of parameters

A common approach to estimate parameters is

Maximum likelihood estimation method

Choose the parameters that **maximise**
the **probability** (i.e. **likelihood**) of the observed (i.e. training) data

Maximum likelihood estimation: single parameter

Example 1

- Suppose we want to model a **biased coin** (or a **class** in binary classification problem)
- and observe data $THHH$, where H — heads, T — tails.
- Assume further:
  - All the flips came from the same coin and each flip was independent (the **i.i.d. assumption**)
  - The coin has a fixed probability β of coming up heads and the probability $1 - \beta$ of coming up tails, i.e. we assume that the data generation distribution is from the family of **Bernoulli distributions parameterised** by the probability of heads $\beta \in [0,1]$ (the **model assumption**)

Maximum likelihood estimation: single parameter

Example 1

- Observed data: $THHH$
- For the heads probability β the probability of observed data:

$$P_\beta(THHH) = P_\beta(T) \cdot P_\beta(H) \cdot P_\beta(H) \cdot P_\beta(H)$$

$$= (1 - \beta)\beta\beta\beta = (1 - \beta)\beta^3 = \beta^3 - \beta^4$$

Maximum likelihood estimation: single parameter

Example 1

- To find β that maximises $P_\beta(THHH) = \beta^3 - \beta^4$
  - compute the derivative of $\beta^3 - \beta^4$
  - set it equal to 0
  - and solve for β

$$\frac{\partial}{\partial\beta}[\beta^3 - \beta^4] = 3\beta^2 - 4\beta^3$$

$$3\beta^2 - 4\beta^3 = 0 \Leftrightarrow 3\beta^2 = 4\beta^3$$

$\beta = \dfrac{3}{4}$ — the maximum likelihood estimate of β

Maximum likelihood estimation: single parameter

Example 2 (generalisation)

- Assume that the observed data consists of h heads and t tails
- Then the probability of the observed data is $\beta^h(1-\beta)^t$
- Instead of maximising the probability it is often more convenient to maximise its logarithm (this is called the **log likelihood** or **log probability**)
- In our example **log likelihood** = $\log(\beta^h(1-\beta)^t) = h\log\beta + t\log(1-\beta)$

Exercise

compute the maximum likelihood estimate of β.

Maximum likelihood estimation: multiple parameters

Example 3

- Suppose we want to model a K-sided die (or a **class** in **multiclass** classification problem)

- We can model this with parameters $\beta_1, \beta_2, \ldots, \beta_K$, where β_i is the probability of having the i-th side of the die. Since β's are probabilities, we should also assume:

  - $\beta_i \geq 0$ for every $i = 1, \ldots, K$, and

  - $\beta_1 + \beta_2 + \ldots + \beta_K = 1$

The model assumption is that the data generating distribution is from the parametric family of **generalised Bernoulli distribution**

Maximum likelihood estimation: multiple parameters

Example 3

- Suppose the observed data consists of x_1 rolls of 1, x_2 rolls of 2, and so on
- Then the probability of this data is $\beta_1^{x_1} \cdot \beta_2^{x_2} \cdot \ldots \cdot \beta_K^{x_K}$ and
- The log probability (i.e. log likelihood) is

$$\sum_{i=1}^{K} x_i \log \beta_i$$

- Thus to find the maximum likelihood estimate of β's we need to find β's that maximise the log likelihood subject to constraint $\beta_1 + \beta_2 + \ldots + \beta_K = 1$

Maximum likelihood estimation: multiple parameters

Example 3

$$\max_{\beta_1,\ldots,\beta_K} \sum_{i=1}^{K} x_i \log \beta_i$$

$$\text{subject to } \sum_{i=1}^{K} \beta_i - 1 = 0$$

Using the method of Lagrange multipliers we obtain the following Lagrangian

$$\mathscr{L}(\bar{\beta}, \lambda) = \mathscr{L}(\beta_1, \ldots, \beta_K, \lambda) = \sum_{i=1}^{K} x_i \log \beta_i - \lambda \left(\sum_{i=1}^{K} \beta_i - 1 \right)$$

Maximum likelihood estimation: multiple parameters

Example 3

- For fixed i, we have

$$\frac{\partial \mathscr{L}(\overline{\beta}, \lambda)}{\partial \beta_i} = \frac{x_i}{\beta_i} - \lambda$$

Setting $\frac{\partial \mathscr{L}(\overline{\beta}, \lambda)}{\partial \beta_i}$ to 0 and solving with respect to β_i we get

$$\beta_i = \frac{x_i}{\lambda}$$

Maximum likelihood estimation: multiple parameters

Example 3

From the constraint $\beta_1 + \beta_2 + \ldots + \beta_K = 1$, (which is equivalent to $\dfrac{\partial \mathcal{L}(\bar{\beta}, \lambda)}{\partial \lambda} = 0$) we have

$$\frac{x_1 + \ldots + x_K}{\lambda} = 1$$

and hence $\lambda = \displaystyle\sum_{i=1}^{K} x_i$.

Thus the maximum likelihood estimate of β's is

$$\beta_i = \frac{x_i}{\sum_{i=1}^{K} x_i} \text{ for every } i = 1, \ldots, K$$

Probabilistic classifiers

Generative

- Naive Bayes

…

Discriminative

- Logistic regression
- Multilayer perceptrons (neural networks)

…