Generative Models for Classification

Introduction to Artificial Intelligence with Mathematics
Lecture Notes

Ganguk Hwang

Department of Mathematical Sciences
KAIST

Consider a pair $(\mathbf{x}, y)$ with an input $\mathbf{x}$ and its label y.

- Generative models vs. Discriminative models ⇝ for Classification
- Generative models
  - Assume some function forms for $p(y)$ and $p(\mathbf{x}|y)$. ⇝ probability distributions
  - Estimate the parameters of $p(y)$ and $p(\mathbf{x}|y)$ from the training data.
    ↳ parameters from prob. distributions
  - Compute $p(y|\mathbf{x})$ by Bayes Theorem.

$$p(y|\mathbf{x}) = \frac{p(y, \mathbf{x})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

- Discriminative models
  - Assume some function form for $p(y|\mathbf{x})$. ⇝ described by parameters
  - Estimate the parameters of $p(y|\mathbf{x})$ from the training data.
    ↳ directly compute

In a generative model, we classify $\mathbf{x}$ based on

$$\operatorname{argmax}_y p(y|\mathbf{x}) = \operatorname{argmax}_y \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} = \operatorname{argmax}_y p(\mathbf{x}|y)p(y).$$

We will study two popular models.

- Naive Bayes Classifier
- Gaussian Discriminative Analysis (Gaussian Bayes Classifier)

Naive Bayes Classifier

Let $\mathbf{x} = (x_1, x_2, \cdots, x_d)$. We assume that all features $x_j, 1 \leq i \leq d$, are conditionally independent given y, which is called the Naive Bayes (NB) assumption. $\hookrightarrow p(a,b|c) = p(a|c)p(b|c)$

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

For example, $\mathbf{x}$ denotes an email and x_j denote words in $\mathbf{x}$.

- Note that Naive Bayes Classifier does not assume a particular Condit. distribution.
- Even though the Naive Bayes assumption is an extremely strong assumption, the resulting algorithm works well on many problems.

The estimation of probabilities of interest

For a data set $\{(\mathbf{x}^{(k)}, y^{(k)}), 1 \leq k \leq N\}$ with $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \cdots, x_d^{(k)})$

counting technique

$$p(x_j = l | y = i) = \frac{\sum_{k=1}^{N} I\{x_j^{(k)} = l, y^{(k)} = i\}}{\sum_{k=1}^{N} I\{y^{(k)} = i\}}, \; 1 \leq l \leq L$$

indicator func

$$p(y = i) = \frac{\sum_{k=1}^{N} I\{y^{(k)} = i\}}{N}, \; 1 \leq i \leq C.$$

Decision Rule

The decision rule is given by

choose label y that maximizes expression

$$\hat{y} = \text{argmax}_y \prod_{j=1}^{d} p(x_j | y) p(y)$$

no data point feature x_i

in training dataset

Laplacian Correction

When there exists some feature x_i with $p(x_i|y) = 0$, even though there is a high probability that $\mathbf{x}$ is classified as y, the resulting probability becomes 0.

To resolve this problem we use Laplacian correction as follows:

\# of feature values

$$p(x_j = l|y = i) = \frac{\sum_{k=1}^{N} I\{x_j^{(k)} = l, y^{(k)} = i\} + 1}{\sum_{k=1}^{N} I\{y^{(k)} = i\} + L}, 1 \le l \le L$$

The idea behind it is that the resulting probability is not changed much by adding one (virtual) data to each type of feature x_i.

Gaussian Naive Bayes Classifier

Assume distribution

Gaussian Naive Bayes Classifier assumes that the likelihood functions are Gaussian, i.e.,

jth feature

cond. probability

$$p(x_j|y=i) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left(-\frac{(x_j-\mu_{ij})^2}{2\sigma_{ij}^2}\right)$$

where μ_{ij} and σ_{ij}^2 are estimated by

data having label i

$$\mu_{ij} = \frac{\sum_{k=1}^{N} I\{y^{(k)}=i\}x_j^{(k)}}{\sum_{k=1}^{N} I\{y^{(k)}=i\}},$$

$$\sigma_{ij}^2 = \frac{\sum_{k=1}^{N} I\{y^{(k)}=i\}(x_j^{(k)}-\mu_{ij})^2}{\sum_{k=1}^{N} I\{y^{(k)}=i\}}$$

Gaussian Discriminative Analysis

Gaussian Discriminative Analysis in its general form assumes that $p(\mathbf{x}|y)$ is distributed according to a multivariate normal distribution

$$p(\mathbf{x}|y=i) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_i)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}.$$

- Each class i has associated mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$.
- We usually assume that all classes share a single covariance matrix $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_C$

For a given data set $\{(\mathbf{x}^{(k)}, y^{(k)}), 1 \leq k \leq N\}$ with $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \cdots, x_d^{(k)})$, we have the following estimation:

↳ k^{th} input vector

$$p(y = i) = \frac{\sum_{k=1}^{N} I\{y^{(k)} = i\}}{N}, 1 \leq i \leq C,$$

marginal probability of label y

$$\mu_{ij} = \frac{\sum_{k=1}^{N} I\{y^{(k)} = i\} x_j^{(k)}}{\sum_{k=1}^{N} I\{y^{(k)} = i\}}, 1 \leq j \leq d,$$

$$\boldsymbol{\mu}_i = (\mu_{i1}, \mu_{i2}, \cdots, \mu_{id})^\top,$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{k=1}^{N} (\mathbf{x}^{(k)} - \boldsymbol{\mu}_{\underbrace{y^{(k)}}_{x^{(k)} \text{ has label } y^{(k)}}})(\mathbf{x}^{(k)} - \boldsymbol{\mu}_{y^{(k)}})^\top.$$

covariance matrix

Decision Rule

The decision rule is given by

label y

$$\hat{y} = \operatorname{argmax}_y p(\mathbf{x}|y)p(y)$$

$p(y=i|x)$

Gaussian Discriminative Analysis vs Logistic Regression

For a binary classification, it is easy to show that, for $\phi = p(y = 1)$

input vector x

$$p(y = 1|\mathbf{x}; \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}, \phi) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})}$$

for some $\mathbf{w}$, which is exactly the same form as logistic regression.

Same form as logistic Regression

Discussions

- If $p(\mathbf{x}|y)$ is multivariate normal, $p(y|\mathbf{x})$ becomes a logistic function.
- However, the converse is not true, which means GDA makes stronger assumptions than logistic regression.
- If the Gaussian assumptions are correct, then GDA performs well.
- On the other hand, logistic regression is more robust and less sensitive to incorrect modeling assumptions.

Example:

concentric circled

We can generate data by using `make_circles` function. It makes two circles whose centers are the same. We use Gaussian naive Bayes classifier for binary classification.

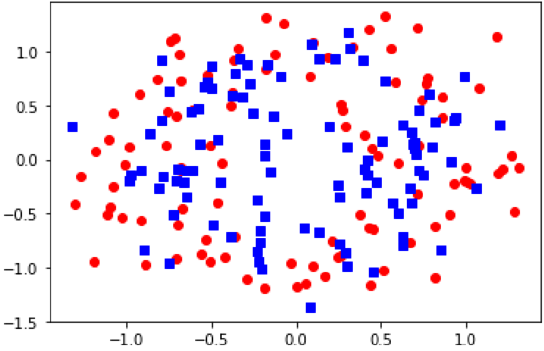


Figure: Randomly generated data ($N = 300$)

for classification

Decision boundary by Gaussian naive Bayes classifier is given below.

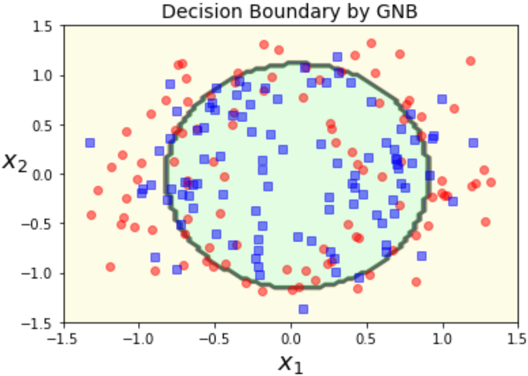
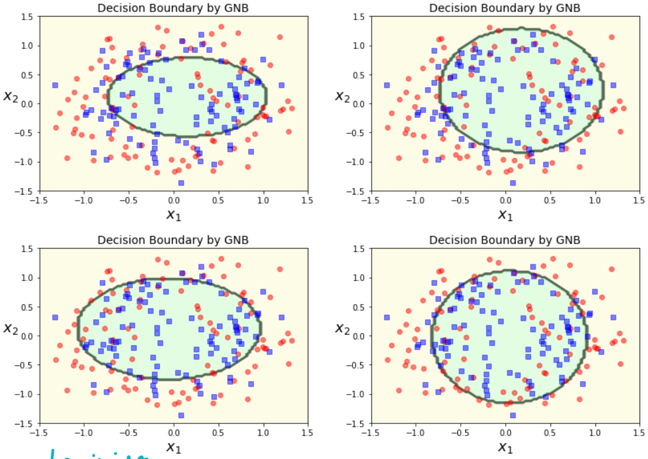


Figure: Decision boundary by Gaussian naive Bayes classifier

performs well

Comparison with different training sets

different decision boundaries



depends on training