Bayes decision rule

Victor Kitov

Yandex School of Data Analysis

Table of Contents

Costs

Classification

- supervised learning
- $y \in \{1, 2, ...C\}$ takes finite discrete set of values

- λ_{yf} is the cost of predicting true class y with forecasted class f.
- Examples with costs: diagnosis prediction, fraud detection, spam filtering, intrusion detection.

Costs

- Matrix of outcomes:

	$f=1$	$f=2$	$\cdots$	$f=C$
$y=1$	λ_{11}	λ_{12}	$\cdots$	λ_{1C}
$y=2$	λ_{21}	λ_{22}	$\cdots$	λ_{2C}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$y=C$	λ_{C1}	λ_{C2}	$\cdots$	λ_{CC}

- Expected cost of solution $\widehat{y}(x)=f$:

$$\mathcal{L}(f)=\sum_{y}p(y|x)\lambda_{yf}$$

Decision rule

- Which best prediction $\widehat{y}(x)$ for object x to select?

Decision rule

- Which best prediction $\widehat{y}(x)$ for object x to select?

Bayes minimum risk decision rule

Assign class, yielding minimum expected cost:

$$\widehat{y}(x) = \arg\min_f \mathcal{L}(f) \tag{1}$$

Decision rule

- Which best prediction $\widehat{y}(x)$ for object x to select?

Bayes minimum risk decision rule

Assign class, yielding minimum expected cost:

$$\widehat{y}(x) = \arg\min_f \mathcal{L}(f) \tag{1}$$

- This rule minimizes expected cost among all rules.
  - if $p(y|x)$ are known

Simplifications

- $\lambda_{yf} \equiv \lambda_y \mathbb{I}[y \neq f]$: constant within class cost of misclassification.

Simplifications

- $\lambda_{yf} \equiv \lambda_y \mathbb{I}[y \neq f]$: constant within class cost of misclassification.

Matrix of outcomes:

	$f=\omega_1$	$f=\omega_1$	$\cdots$	$f=\omega_1$
$y=\omega_1$	0	λ_1	$\cdots$	λ_1
$y=\omega_2$	λ_2	0	$\cdots$	λ_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$y=\omega_C$	λ_C	λ_C	$\cdots$	0

Simplifications

- $\lambda_{yf} \equiv \lambda_y \mathbb{I}[y \neq f]$: constant within class cost of misclassification.

Matrix of outcomes:

	$f=\omega_1$	$f=\omega_1$	$\cdots$	$f=\omega_1$
$y=\omega_1$	0	λ_1	$\cdots$	λ_1
$y=\omega_2$	λ_2	0	$\cdots$	λ_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$y=\omega_C$	λ_C	λ_C	$\cdots$	0

- Expected cost of solution $\widehat{y}(x)=f$:
  $\mathcal{L}(f)=\sum_y p(y|x)\lambda_y \mathbb{I}[f \neq y]$

Equal misclassification costs

- Suppose further $\lambda_y \equiv \lambda \, \forall y$.

Equal misclassification costs

- Suppose further $\lambda_y \equiv \lambda \, \forall y$.
- Then cost of prediction equals:

$$\mathcal{L}(f) = \sum_y p(y|x)\lambda\mathbb{I}[f \neq y] = \sum_y p(y|x)\lambda - p(f|x)\lambda = \lambda(1-p(f|x))$$

- And (1) becomes:

$$\widehat{y}(x) = \arg\min_f \lambda(1 - p(f|x)) = \arg\max_f p(f|x) \qquad (2)$$

- This is termed **maximum posterior probability rule** or **Bayes minimum error rule.**

Equal misclassification costs

- This rule minimizes expected error rate.
    - if $p(y|x)$ are known

Equal misclassification costs

- This rule minimizes expected error rate.
  - if $p(y|x)$ are known
- If x and y are independent, then (2) reduces to

$$\widehat{y}(x) = \arg\max_f p(f|x) = \arg\max_f p(f)$$

Table of Contents

Generative and discriminative models

Generative model

Full distribution $p(x, y)$ is modeled.

- Can generate new observations (x, y)

$$
\begin{aligned}
\widehat{y}(x) &= \arg\max_y p(y|x) = \arg\max_y \frac{p(x,y)}{p(x)} = \arg\max_y p(y)p(x|y) \\
&= \arg\max_y \{\log p(y) + \log p(x|y)\}
\end{aligned}
$$

Generative and discriminative models

Generative model

Full distribution $p(x, y)$ is modeled.

- Can generate new observations (x, y)

$$
\begin{aligned}
\widehat{y}(x) &= \arg\max_y p(y|x) = \arg\max_y \frac{p(x,y)}{p(x)} = \arg\max_y p(y)p(x|y) \\
&= \arg\max_y \{\log p(y) + \log p(x|y)\}
\end{aligned}
$$

Discriminative model

- **Discriminative with probability**: only $p(y|x)$ is modeled
- **Reduced discriminative**: only $y = f(x)$ is modeled.

Discussion

- **Disadvantages of generative models:**
  - Discriminative models are more general
  - $p(x|y)$ may be inaccurate in high dimensional spaces

Discussion

- **Disadvantages of generative models:**
  - Discriminative models are more general
  - $p(x|y)$ may be inaccurate in high dimensional spaces
- **Advantages of generative models:**
  - Generative models can be adjusted to varying $p(y)$
  - Naturally adjust to missing features (by marginalization)
  - Easily detect outliers (small $p(x)$)

Table of Contents

Gaussian classifier

- In Gaussian classifier
$$p(x|y) = \frac{1}{(2\pi)^{D/2}|\Sigma_y|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)\right\}$$

Gaussian classifier

- In Gaussian classifier
$$p(x|y) = \frac{1}{(2\pi)^{D/2}|\Sigma_y|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)\right\}$$
- It follows that
$$\begin{aligned}\log p(y|x) &= \log p(x|y) + \log p(y) - \log p(x) \\ &= -\frac{1}{2}(x-\mu_y^T)\Sigma_y^{-1}(x-\mu_y) - \frac{1}{2}\log|\Sigma_y| \\ &\quad -\frac{D}{2}\log(2\pi) + \log p(y) - \log p(x)\end{aligned}$$

Gaussian classifier

- In Gaussian classifier

$$p(x|y) = \frac{1}{(2\pi)^{D/2}|\Sigma_y|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)\right\}$$

- It follows that

$$\begin{aligned}\log p(y|x) &= \log p(x|y) + \log p(y) - \log p(x)\\ &= -\frac{1}{2}(x-\mu_y^T)\Sigma_y^{-1}(x-\mu_y) - \frac{1}{2}\log|\Sigma_y|\\ &\quad -\frac{D}{2}\log(2\pi) + \log p(y) - \log p(x)\end{aligned}$$

- Removing common additive terms, we obtain discriminant functions:

$$g_y(x) = \log p(y) - \frac{1}{2}\log|\Sigma_y| - \frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y) \quad (3)$$

Practical application

- In practice we replace theoretical terms μ_y, Σ_y with their sample estimates $\widehat{\mu}_y$, $\widehat{\Sigma}_y$.
- $\widehat{p}(y) = \frac{N_y}{N}$.

$$g_y(x) = \log \widehat{p}(y) - \frac{1}{2} \log |\widehat{\Sigma}_y| - \frac{1}{2}(x - \widehat{\mu}_y)^T \widehat{\Sigma}_y^{-1}(x - \widehat{\mu}_y)$$

- Analysis:
  - depends on normality assumptions (in particular - on unimodality)
  - needs to specify:
    - CD parameters to estimate $\widehat{\mu}_y$, $y = 1, 2, ...C$.
    - $CD(D+1)/2$ parameters to estimate $\widehat{\Sigma}_y$, $j = 1, 2, ...C$.

Simplifying assumptions

- $CD(D+3)/2$ may be too large for multidimensional tasks with small training sets.
- Simplifying assumptions:
    - **Naive Bayes**: assume that $\Sigma_1, \Sigma_2, ...\Sigma_C$ are diagonal.
    - **Project data onto a subspace**: for example on first few principal components.
    - **Proportional covariance matrices**: assume that $\Sigma_1 = \alpha_1\Sigma,\ \Sigma_2 = \alpha_2\Sigma,\ ...\ \Sigma_C = \alpha_C\Sigma$.
    - **Fisher's linear discriminant analysis**: assume that $\Sigma_1 = \Sigma_2 = ... = \Sigma_C$.

Table of Contents

Naive Bayes assumption

$$p(x^1, x^2, ...x^D) = p(x^1)p(x^2|x^1)...p(x^D|x^1, x^2, ...x^{D-1})$$

Naive Bayes assumption

$$p(x^1, x^2, \ldots x^D) = p(x^1)p(x^2|x^1)\ldots p(x^D|x^1, x^2, \ldots x^{D-1})$$

Cure: make simplifying assumptions.

Naive Bayes assumption

$p(x^1, x^2, ...x^D) = p(x^1)p(x^2|x^1)...p(x^D|x^1, x^2, ...x^{D-1})$

Cure: make simplifying assumptions.

Independence assumption

Individual features are independent: $p(x) = p(x^1)p(x^2)...p(x^D)$

Naive Bayes assumption

$$p(x^1, x^2, ...x^D) = p(x^1)p(x^2|x^1)...p(x^D|x^1, x^2, ...x^{D-1})$$

Cure: make simplifying assumptions.

Independence assumption

Individual features are independent: $p(x) = p(x^1)p(x^2)...p(x^D)$

Naive Bayes assumption in classification

Individual features are **class conditionally** independent:
$p(x|y) = p(x^1|y)p(x^2|y)...p(x^D|y)$

Under Naive Bayes assumption max-posterior probability rule becomes:

$$\widehat{y}(x) = \arg\max_y p(y)p(x^1|y)p(x^2|y)...p(x^D|y)$$

Table of Contents

Text models

- Restrict attention to M words $w_1, w_2, ...w_M$
    - all unique words
    - possibly with stop words removal
    - possibly only most frequent words
    - or only words relevant to the topic of study
- Two major models:
    - Bernoulli
    - Multinomial

Bernoulli model

- Document is represented with feature vector $x \in \mathbb{R}^M$
- $x^d = \mathbb{I}[w_d \text{ appeared in document}]$
- $\theta_y^d = p(x^d = 1|y)$
- $p(x|y) = \prod_{d=1}^{M} \left(\theta_y^d\right)^{x^d} \left(1 - \theta_y^d\right)^{1-x^d}$
- $p(y) = \frac{N_y}{N}$
- $\theta_y^d = \frac{N_{yx^d}}{N_y}$
- Smoothed variant: $\theta_y^d = \frac{N_{yx^d}+c}{N_y+2c}$

Multinomial model

- Document is represented with feature vector $x \in \mathbb{R}^M$
- x^d =number of times w_i appeared in document d
- θ_y^d =probability of w_i on word position
- $p(x|y) = \frac{\left(\sum_d x^d\right)!}{\prod_d (x^d)!} \prod_{d=1}^M \left(\theta_y^d\right)^{x^d}$
- $p(y) = \frac{N_y}{N}$
- θ_y^d =#{w_i in sequence of words from class y}/#{of words in sequence of class y}
- Smoothing also used.