Naive Bayes Classifier

朱新梅

1. *Inputs*: $X = (X_1, \ldots, X_p)^T \in \mathbb{R}^p$
2. *Output*: $Y \in \{1, \ldots, J\}$
3. *Training set*: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$
4. *Assumption*: Given a class $Y = j$, the features are independent 特征的条件独立性

$$\Pr(X|Y = j) = \prod_{i=1}^{p} \Pr(X_i|Y = j)$$

5. *Parameters*:

- *class probabilities:* $\pi_j = \Pr(Y = j), j = 1, \ldots, J,$
- *class conditional probabilties:* $\theta_{ji}^a = \Pr(X_i = a|Y = j), X_i \in A_i, i = 1, \ldots, p.$
- *total number of parameters:* $J(\sum_{i=1}^p |A_i| + 1)$

6. *Generalized additive model (GAM)*: log-transform, using class J as the base (基类)

$$\begin{aligned}
\log \frac{\Pr(Y = j|X)}{\Pr(Y = J|X)} =& \log \frac{\Pr(Y = j)\Pr(X|Y = j)}{\Pr(Y = J)\Pr(X|Y = J)} \\
=& \log \frac{\pi_j \Pr(X|Y = j)}{\pi_J \Pr(X|Y = J)} \\
=& \log \frac{\pi_j \prod_{i=1}^p \Pr(X_i|Y = j)}{\pi_J \prod_{i=1}^p \Pr(X_i|Y = J)} \\
=& \log \frac{\pi_j}{\pi_J} + \sum_{i=1}^{p} \log \frac{\Pr(X_i|Y = j)}{\Pr(X_i|Y = J)} \\
=& \alpha_j + \sum_{i=1}^{p} g_{ji}(X_i)
\end{aligned}$$

7. *Maiximum a posteriori, MAP*: Given new input x^*,

$$\begin{aligned}
y \leftarrow& \arg \max_{j=1,\ldots,J} \frac{\pi_j \prod_{i=1}^p \Pr(x_i^*|Y = j)}{\sum_{l=1}^J \pi_l \prod_{i=1}^p \Pr(x_i^*|Y = l)} \\
=& \arg \max_{j=1,\ldots,J} \pi_j \prod_{i=1}^{p} \Pr(x_i^*|Y = j)
\end{aligned}$$

8. *Why MAP?*:

- *0-1 loss function*: $L(Y, f(X)) = I(Y \neq f(X))$
- *risk function/expected prediction error*:

$$\begin{aligned}
EPE(f) &= E[L(Y, f(X))] \\
&= E_X E_{Y|X}[L(Y, f(X))] \\
&= E_X \sum_{j=1}^{J} L(j, f(X)) \Pr(Y = j|X)
\end{aligned}$$

- $\hat{f} = \arg\min_f EPE(f)$, it suffices to minimize EPE pointwise

$$\begin{aligned}\hat{f}(x) &= \arg\min_{f(x)\in\{1,\dots,J\}} \sum_{j=1}^{J} L(j, f(x)) \Pr(Y=j|X=x)\\ &= \arg\min_{g\in\{1,\dots,J\}} \sum_{j=1}^{J} L(j, g) \Pr(Y=j|X=x)\\ &= \arg\min_{g\in\{1,\dots,J\}} \sum_{j\neq g} \Pr(Y=j|X=x)\\ &= \arg\min_{g\in\{1,\dots,J\}} \Pr(Y\neq g|X=x)\\ &= \arg\min_{g\in\{1,\dots,J\}} [1-\Pr(Y=g|X=x)]\\ &= \arg\max_{g\in\{1,\dots,J\}} \Pr(Y=g|X=x).\end{aligned}$$

9. *Estimation of parameters: MLE*

- *log-likelihood*:

$$\begin{aligned}l(\pi,\theta) &= \sum_{k=1}^{N} \log\left(\Pr(Y=y^{(k)})\prod_{i=1}^{p}\Pr(X_i=x_i^{(k)}|Y=y^{(k)})\right)\\ &= \sum_{k=1}^{N}\log\Pr(Y=y^{(k)}) + \sum_{k=1}^{N}\sum_{i=1}^{p}\log\Pr(X_i=x_i^{(k)}|Y=y^{(k)}).\end{aligned}$$

- *notation*:

$$u_j = \#(y^{(k)}=j), j=1,\dots,J$$

$$v_{ji}^{a} = \#(y^{(k)}=j, x_i^{(k)}=a), a\in A_i; j=1,\dots,J; i=1,\dots,p$$

- *re-write log-likelihood*:

$$l(\pi,\theta) = \sum_{j=1}^{J} u_j\log\pi_j + \sum_{j=1}^{J}\sum_{i=1}^{p}\sum_{a\in A_i} v_{ji}^{a}\log\theta_{ji}^{a}$$

- *goal*:

$$\text{maximize}_{\pi,\theta} \quad l(\pi,\theta)$$

$$s.t. \quad \sum_{j=1}^{J}\pi_j = 1, \sum_{a\in A_i}\theta_{ji}^{a} = 1$$

- *Lagrange multiplier:*

$$f(\pi,\theta,\lambda,\mu) = \sum_{j=1}^{J} u_j\log\pi_j + \sum_{j=1}^{J}\sum_{i=1}^{p}\sum_{a\in A_i} v_{ji}^{a}\log\theta_{ji}^{a} - \lambda\left(\sum_{j=1}^{J}\pi_j - 1\right) - \sum_{j=1}^{J}\sum_{i=1}^{p}\mu_{ji}\left(\theta_{ji}^{a}-1\right)$$

$$\begin{aligned}\frac{\partial f}{\partial\pi_j} &= \frac{u_j}{\pi_j} - \lambda \overset{let}{=} 0,\\ \Longleftrightarrow & u_j = \lambda\pi_j,\\ \Longleftrightarrow & N = \sum_{j=1}^{J} u_j = \lambda\sum_{i=1}^{J}\pi_j = \lambda\\ \Longleftrightarrow & \hat{\pi}_j = \frac{1}{N}\#(y^{(k)}=j), \quad j=1,\dots,J\end{aligned}$$

$$\begin{aligned}
&\frac{\partial f}{\partial \theta_{ji}^a} = \frac{v_{ji}^a}{\theta_{ji}^a} - \mu_{ji} \overset{let}{=} 0, \\
\Longleftrightarrow & v_{ji}^a = \mu_{ji}\theta_{ji}^a, \\
\Longleftrightarrow & u_j = \sum_{a\in A_i} v_{ji}^a = \mu_{ji}\sum_{a\in A_i}\theta_{ji}^a = \mu_{ji} \\
\Longleftrightarrow & \hat\theta_{ji}^a = \frac{v_{ji}^a}{u_j} = \frac{\#(y^{(k)} = j, x_i^{(k)} = a)}{\#(y^{(k)} = j)}, \quad i \in A_i; j = 1, \dots, J; i = 1, \dots, p
\end{aligned}$$

10. *smoothing*:

$$\begin{aligned}
\hat\theta_{ji}^a &= \frac{\#(y^{(k)} = j, x_i^{(k)} = a) + \tau}{\#(y^{(k)} = j) + |A_i|\tau}, \quad i \in A_i; j = 1, \dots, J; i = 1, \dots, p \\
\hat\pi_j &= \frac{\#(y^{(k)} = j) + \tau}{N + J\tau}, \quad j = 1, \dots, J
\end{aligned}$$

- $\tau = 0$, MLE
- $\tau = 1$, Laplace smoothing (also referred to as *add-one smoothing*)
- $\tau = 0.5$, Jeffrey's smoothing
- $0 < \tau < 1$, Lidstone smoothing

11. *continuous input?* **Gaussian Naive Bayes, GNB**

$$\Pr(X_i = a|Y = j) = \frac{1}{\sqrt{2\pi\sigma_{ji}^2}} \exp\left(-\frac{(a - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

where variance is sometimes assumed to be

- independent of Y: i.e. $\sigma_{ij}^2 = \sigma_i^2$
- independent of X: i.e. $\sigma_{ij}^2 = \sigma_j^2$
- or both: i.e. $\sigma_{ij}^2 = \sigma^2$

12. *pros*:

- appropriate when the dimensionality p of the feature space is high
- highly scalable, fast to train and classify
- not sensitive to irrelevant features
- handles real and discrete inputs
- handles streaming data well

13. *cons*:

- zero frequency → use *smoothing*
- assumes independence of features → remove correlated feature, data pre-processing and *feature selection*
- sensitivity to training size
- sensitivity to smoothing

14. R packages and functions:

- package `e1041`, function `naiveBayes`
- package `klaR`, function `NaiveBayes`
- package `caret`, function `train`, argument `method = "nb"`