Naive classifiers and zero counts

Marco Cattaneo
Department of Statistics, LMU Munich
cattaneo@stat.uni-muenchen.de

WPMSIIP 2010, Durham, UK
9 September 2010

naive classification

- naive assumption: the features $F_1, \ldots, F_k$ are **independent** given the class C;

naive classification

- naive assumption: the features $F_1, \dots, F_k$ are **independent** given the class C; let $\{P_\theta : \theta \in \Theta\}$ be the set of **all** probability distributions for $(C, F_1, \dots, F_k)$ satisfying the naive assumption

naive classification

- naive assumption: the features $F_1, \ldots, F_k$ are **independent** given the class C; let $\{P_\theta : \theta \in \Theta\}$ be the set of **all** probability distributions for $(C, F_1, \ldots, F_k)$ satisfying the naive assumption
- a **training dataset** induces the likelihood function *lik* on Θ defined by

$$lik(\theta) \propto P_\theta(\text{dataset}) \qquad \text{for all } \theta \in \Theta$$

naive classification

- naive assumption: the features $F_1, \ldots, F_k$ are **independent** given the class C; let $\{P_\theta : \theta \in \Theta\}$ be the set of **all** probability distributions for $(C, F_1, \ldots, F_k)$ satisfying the naive assumption
- a **training dataset** induces the likelihood function *lik* on Θ defined by

$$lik(\theta) \propto P_\theta(\text{dataset}) \qquad \text{for all } \theta \in \Theta$$

- to study the **preference** between the classifications $C = a$ and $C = b$ for a new object with observed features $F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h}$, we can introduce the function g on Θ defined by

$$g(\theta) = \frac{P_\theta(C = a \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})}{P_\theta(C = b \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})} \qquad \text{for all } \theta \in \Theta$$

naive classification

- naive assumption: the features $F_1, \ldots, F_k$ are **independent** given the class C; let $\{P_\theta : \theta \in \Theta\}$ be the set of **all** probability distributions for $(C, F_1, \ldots, F_k)$ satisfying the naive assumption
- a **training dataset** induces the likelihood function *lik* on Θ defined by

$$lik(\theta) \propto P_\theta(\text{dataset}) \qquad \text{for all } \theta \in \Theta$$

- to study the **preference** between the classifications $C = a$ and $C = b$ for a new object with observed features $F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h}$, we can introduce the function g on Θ defined by

$$g(\theta) = \frac{P_\theta(C = a \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})}{P_\theta(C = b \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})} \qquad \text{for all } \theta \in \Theta$$

- naive classifiers differ in the way in which they use *lik* to evaluate g:

naive classification

- naive assumption: the features $F_1, \ldots, F_k$ are **independent** given the class C; let $\{P_\theta : \theta \in \Theta\}$ be the set of **all** probability distributions for $(C, F_1, \ldots, F_k)$ satisfying the naive assumption
- a **training dataset** induces the likelihood function *lik* on Θ defined by

$$lik(\theta) \propto P_\theta(\text{dataset}) \qquad \text{for all } \theta \in \Theta$$

- to study the **preference** between the classifications $C = a$ and $C = b$ for a new object with observed features $F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h}$, we can introduce the function g on Θ defined by

$$g(\theta) = \frac{P_\theta(C = a \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})}{P_\theta(C = b \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})} \qquad \text{for all } \theta \in \Theta$$

- naive classifiers differ in the way in which they use *lik* to evaluate g:
  - naive **precise** classifiers: $g(\widehat{\theta})$, where $\widehat{\theta} \in \Theta$ is a precise estimate of θ (for example: ML, Bayesian)

naive classification

- naive assumption: the features $F_1, \ldots, F_k$ are **independent** given the class C; let $\{P_\theta : \theta \in \Theta\}$ be the set of **all** probability distributions for $(C, F_1, \ldots, F_k)$ satisfying the naive assumption
- a **training dataset** induces the likelihood function *lik* on Θ defined by

$$lik(\theta) \propto P_\theta(\text{dataset}) \qquad \text{for all } \theta \in \Theta$$

- to study the **preference** between the classifications $C = a$ and $C = b$ for a new object with observed features $F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h}$, we can introduce the function g on Θ defined by

$$g(\theta) = \frac{P_\theta(C = a \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})}{P_\theta(C = b \,|\, F_{i_1} = f_{i_1}, \ldots, F_{i_h} = f_{i_h})} \qquad \text{for all } \theta \in \Theta$$

- naive classifiers differ in the way in which they use *lik* to evaluate g:
  - naive **precise** classifiers: $g(\widehat{\theta})$, where $\widehat{\theta} \in \Theta$ is a precise estimate of θ (for example: ML, Bayesian)
  - naive **credal** classifier: $\{g(\theta) : \theta \in \mathcal{C}\}$, where $\mathcal{C} \subseteq \Theta$ is the IDM imprecise estimate of θ

naive classification

- naive assumption: the features $F_1, \dots, F_k$ are **independent** given the class C; let $\{P_\theta : \theta \in \Theta\}$ be the set of **all** probability distributions for $(C, F_1, \dots, F_k)$ satisfying the naive assumption
- a **training dataset** induces the likelihood function *lik* on Θ defined by

$$lik(\theta) \propto P_\theta(\text{dataset}) \qquad \text{for all } \theta \in \Theta$$

- to study the **preference** between the classifications $C = a$ and $C = b$ for a new object with observed features $F_{i_1} = f_{i_1}, \dots, F_{i_h} = f_{i_h}$, we can introduce the function g on Θ defined by

$$g(\theta) = \frac{P_\theta(C = a \,|\, F_{i_1} = f_{i_1}, \dots, F_{i_h} = f_{i_h})}{P_\theta(C = b \,|\, F_{i_1} = f_{i_1}, \dots, F_{i_h} = f_{i_h})} \qquad \text{for all } \theta \in \Theta$$

- naive classifiers differ in the way in which they use *lik* to evaluate g:
  - naive **precise** classifiers: $g(\widehat{\theta})$, where $\widehat{\theta} \in \Theta$ is a precise estimate of θ (for example: ML, Bayesian)
  - naive **credal** classifier: $\{g(\theta) : \theta \in \mathcal{C}\}$, where $\mathcal{C} \subseteq \Theta$ is the IDM imprecise estimate of θ
  - naive **hierarchical** classifier: $\{g(\theta) : \theta \in \Theta,\ lik(\theta) > \beta\}$, where $\beta \in [0, 1[$

naive hierarchical classifier

- likelihood-based confidence region for $g(\theta)$ with cutoff point $\beta \in [0, 1[$:

$$\{g(\theta) : \theta \in \Theta,\ lik(\theta) > \beta\}$$

naive hierarchical classifier

- likelihood-based confidence region for $g(\theta)$ with cutoff point $\beta \in [0, 1[$:

  $$\{g(\theta) : \theta \in \Theta,\ lik(\theta) > \beta\} = \{x \in [0, +\infty] : lik_g(x) > \beta\},$$

  where lik_g is the **profile likelihood** function on $[0, +\infty]$ induced by *lik* and g:

  $$lik_g(x) = \sup_{\theta \in \Theta \,:\, g(\theta) = x} lik(\theta)$$

naive hierarchical classifier

- likelihood-based confidence region for $g(\theta)$ with cutoff point $\beta \in [0, 1[$:

$$\{g(\theta) : \theta \in \Theta,\ lik(\theta) > \beta\} = \{x \in [0, +\infty] : lik_g(x) > \beta\},$$

  where lik_g is the **profile likelihood** function on $[0, +\infty]$ induced by lik and g:

$$lik_g(x) = \sup_{\theta \in \Theta \,:\, g(\theta) = x} lik(\theta)$$

- basic idea for calculating lik_g: if $\theta = \theta_\alpha$ maximizes $[g(\theta)]^\alpha\, lik(\theta)$ over all $\theta \in \Theta$ for some $\alpha \in \mathbb{R}$, then it also maximizes $lik(\theta)$ over all $\theta \in \Theta$ such that $g(\theta) = g(\theta_\alpha)$, and therefore $(g(\theta_\alpha), lik(\theta_\alpha))$ is a point in the **graph** of lik_g;

naive hierarchical classifier

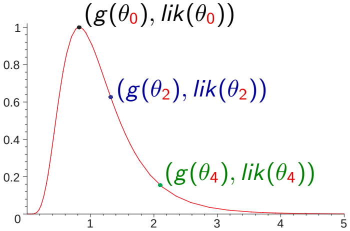
- likelihood-based confidence region for $g(\theta)$ with cutoff point $\beta \in [0, 1[$:

$$\{g(\theta) : \theta \in \Theta,\ lik(\theta) > \beta\} = \{x \in [0, +\infty] : lik_g(x) > \beta\},$$

  where lik_g is the **profile likelihood** function on $[0, +\infty]$ induced by lik and g:

$$lik_g(x) = \sup_{\theta \in \Theta \,:\, g(\theta) = x} lik(\theta)$$

- basic idea for calculating lik_g: if $\theta = \theta_\alpha$ maximizes $[g(\theta)]^\alpha\, lik(\theta)$ over all $\theta \in \Theta$ for some $\alpha \in \mathbb{R}$, then it also maximizes $lik(\theta)$ over all $\theta \in \Theta$ such that $g(\theta) = g(\theta_\alpha)$, and therefore $(g(\theta_\alpha), lik(\theta_\alpha))$ is a point in the **graph** of lik_g; in fact, θ_α is the ML estimate of θ with α-modified data, and by varying α, the whole graph of lik_g is obtained



using all the available information

- observed data:

$$\underbrace{(C^{(1)}, F^{(1)}), \ldots, (C^{(n)}, F^{(n)})}_{\text{training dataset}}, \underbrace{(C^{(n+1)}, F^{(n+1)}), \ldots, (C^{(n+m)}, F^{(n+m)})}_{\text{objects to be classified}}$$

using all the available information

- observed data:

$$\underbrace{(C^{(1)}, F^{(1)}), \dots, (C^{(n)}, F^{(n)})}_{\text{training dataset}}, \underbrace{(C^{(n+1)}, F^{(n+1)}), \dots, (C^{(n+m)}, F^{(n+m)})}_{\text{objects to be classified}}$$

- the likelihood function *lik* on Θ used by the naive classifiers is induced by the training dataset, **without** considering the information provided by the observations of $F^{(n+1)}, \dots, F^{(n+m)}$

using all the available information

- observed data:

$$\underbrace{(C^{(1)}, F^{(1)}), \dots, (C^{(n)}, F^{(n)})}_{\text{training dataset}}, \underbrace{(C^{(n+1)}, F^{(n+1)}), \dots, (C^{(n+m)}, F^{(n+m)})}_{\text{objects to be classified}}$$

- the likelihood function *lik* on Θ used by the naive classifiers is induced by the training dataset, **without** considering the information provided by the observations of $F^{(n+1)}, \dots, F^{(n+m)}$

- when $m = 1$, the whole information provided by the observation of $F^{(n+1)}$ is **automatically** used by the (precise or imprecise) Bayesian classifiers, but **not** by the likelihood-based ones

example of naive classification

$$\begin{tikzcd} & \mathit{class} \in \{a, b\} \arrow[dl] \arrow[dr] & \\ \mathit{color} \in \{\mathit{red}, \mathit{blue}\} & & \mathit{size} \in \{\mathit{BIG}, \mathit{small}\} \end{tikzcd}$$

complete training dataset:

class	#	#*red*	#*BIG*
a	50	1	1
b	50	1	50

$$P(\mathit{class} = a \mid \mathit{color} = \mathit{red},\ \mathit{size} = \mathit{BIG}):$$

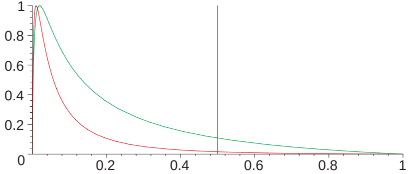
example of naive classification

$$\begin{tikzcd} & \mathit{class} \in \{a, b\} \arrow[dl] \arrow[dr] & \\ \mathit{color} \in \{\mathit{red}, \mathit{blue}\} & & \mathit{size} \in \{\mathit{BIG}, \mathit{small}\} \end{tikzcd}$$

complete training dataset:

class	#	#*red*	#*BIG*
a	50	1	1
b	50	1	50

$P(\mathit{class} = a \,|\, \mathit{color} = \mathit{red},\ \mathit{size} = \mathit{BIG})$:



example of naive classification

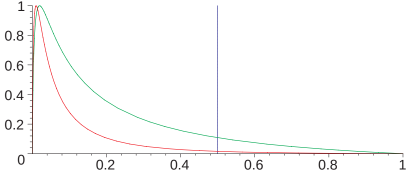
$$class \in \{a, b\}$$

$$color \in \{red, blue\} \qquad size \in \{BIG, small\}$$

complete training dataset:

class	#	#*red*	#*BIG*
a	50	1	1
b	50	1	50

$P(class = a \mid color = red,\ size = BIG)$:



ML estimate with all the available information:	0.010
ML estimate without considering $F^{(n+1)}$:	0.020
Bayesian estimate with uniform priors:	0.038
IDM estimate with $s = 2$:	$[0.0066, 0.15]$

example of naive classification with zero counts

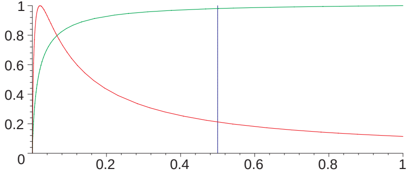
$class \in \{a, b\}$

$color \in \{red, blue\}$ $\quad$ $size \in \{BIG, small\}$

complete training dataset:

$class$	$\#$	$\#red$	$\#BIG$
a	50	1	1
b	50	0	50

$P(class = a \,|\, color = red, \; size = BIG)$:



ML estimate with all the available information:	0.021
ML estimate without considering $F^{(n+1)}$:	1
Bayesian estimate with uniform priors:	0.073
IDM estimate with $s = 2$:	$[0.0099, 1]$

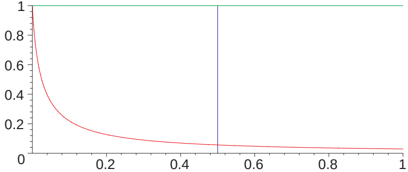
example of naive classification with zero counts

$$\begin{tikzcd} & class \in \{a, b\} \arrow[dl] \arrow[dr] & \\ color \in \{red, blue\} & & size \in \{BIG, small\} \end{tikzcd}$$

complete training dataset:

$class$	#	#red	#BIG
a	50	0	1
b	50	0	50

$P(class = a \mid color = red,\ size = BIG)$:



ML estimate with all the available information:	0
ML estimate without considering $F^{(n+1)}$:	$[0, 1]$
Bayesian estimate with uniform priors:	0.038
IDM estimate with $s = 2$:	$[0, 1]$