Optimal decision and naive Bayes

Fabrice Rossi

CEREMADE
Université Paris Dauphine

2021
Outline

Standard supervised learning hypothesis

Optimal model

The Naive Bayes Classifier
Data model

Data spaces

- $\mathcal{X}$: “input” space, always observed
- $\mathcal{Y}$: “output” space, values to predict, observed during learning

Minimal structural assumption
$\mathcal{X}$ should be equipped with a dissimilarity $d$:

- $d$ is a function from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}^+$
- $d$ is symmetric
- $\forall \mathbf{X}, \mathbf{X}' \in \mathcal{X}, \quad d(\mathbf{X}, \mathbf{X}) \leq d(\mathbf{X}, \mathbf{X}')$

Multivariate assumption
In general $\mathcal{X}$ is structured into “variables”:

- $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_P$
- $\mathbf{X} = (X_1, X_2, \ldots, X_P)^T$
Definition and use
- A predictive model is a function \( g \) from \( \mathcal{X} \) to \( \mathcal{Y} \)
- for an observation \( (X, Y) \) one expects \( g(X) \) to be “close to” \( Y \)

Loss function
A loss function \( l \) is
- a function from \( \mathcal{Y} \times \mathcal{Y} \) to \( \mathbb{R}^+ \)
- such that \( \forall Y \in \mathcal{Y}, \quad l(Y, Y) = 0 \)

Interpretation
\( l(g(X), Y) \) measures the loss incurred by the user of a model \( g \) when the true value \( Y \) is replaced by the prediction \( g(X) \).
Examples

$\mathcal{Y} = \mathbb{R}$

- $l_2(p, t) = (p - t)^2$
- $l_1(p, t) = |p - t|$
- $l_{APE}(p, t) = \frac{|p - t|}{|t|}$

$|\mathcal{Y}| < \infty$

- $l_b(p, t) = 1_{p \neq t}$
- general case when $\mathcal{Y} = \{y_1, y_2\}$

<table>
<thead>
<tr>
<th>( l(p, t) )</th>
<th>( t = y_1 )</th>
<th>( t = y_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p = y_1 )</td>
<td>0</td>
<td>( l(y_1, y_2) )</td>
</tr>
<tr>
<td>( p = y_2 )</td>
<td>( l(y_2, y_1) )</td>
<td>0</td>
</tr>
</tbody>
</table>

asymmetric costs are important in practice (think SPAM versus non SPAM)
Stochastic framework

Main hypotheses

- observations are random variables with values in $\mathcal{X} \times \mathcal{Y}$
- they are distributed according to a fixed and unknown distribution $D$
- observations are independent

A data set

- $\mathcal{D} = ((X_i, Y_i))_{1 \leq i \leq N}$
- $(X_i, Y_i) \sim D$ and $\mathcal{D} \sim D^N$
- notation: $X_i = (X_{i1}, X_{i2}, \ldots, X_{iP})^T$

Risk of a model

The risk of $g$ for the loss function $l$ is

$$R_l(g) = \mathbb{E}_{(X, Y) \sim D}(l(g(X), Y))$$
Technical aspects

Additional hypotheses

- there is an underlying probability space
- $\mathcal{X} \times \mathcal{Y}$ must be a measurable space
- in general this is done via the standard Borel sigma field on $\mathbb{R}^d$

Measurability

- loss functions must be measurable functions
- ditto for models
- technically, the loss could be $+\infty$

Independence and stationarity

- independence can be relaxed for e.g. time series
- stationarity also for e.g. drift analysis
Standard supervised learning hypothesis

Optimal model

The Naive Bayes Classifier
Optimal model

Optimal risk

- $R_i^* = \inf_g R_i(g)$
- called the Bayes risk when $\mathcal{Y}$ is finite

Is $R_i^*$ reachable?

- in general $\arg\min_g R_i(g)$ is a set: could it be empty?
- $g_i^*$: a model such that $R_i(g_i^*) = R_i^*$
Optimal model

Optimal risk

\[ R_i^* = \inf_g R_i(g) \]

- called the Bayes risk when \( \mathcal{Y} \) is finite

Is \( R_i^* \) reachable?

- in general \( \arg \min_g R_i(g) \) is a set: could it be empty?
- \( g_i^* \): a model such that \( R_i(g_i^*) = R_i^* \)

Quadratic case

- \( \mathcal{Y} = \mathbb{R} \), \( l_2(p, \nu) = (p - \nu)^2 \)
- \( g^*(x) = \mathbb{E}_{(x, y) \sim D}(Y|X = x) \)
Discrete case

If $\mathcal{Y}$ is finite

- a loss function is a table with $|\mathcal{Y}|(|\mathcal{Y}| - 1)$ non zero entries
- $g^*$ can be obtained using conditional probabilities $P(Y = y|X = x)$

Simple case

- $l_b(p, t) = 1_{p \neq t}$
- $g^*_b(x) = \arg \max_{y \in \mathcal{Y}} P(X, Y) \sim D(Y = y|X = x)$

General case

\[ g^*_i(x) = \arg \min_{y \in \mathcal{Y}} \sum_{y' \neq y} l(y, y') P(X, Y) \sim D(Y = y'|X = x) \]
Idea of the proof

- conditional reasoning

\[ R_t(g) = \mathbb{E}_{(x, y) \sim D} \left\{ \mathbb{E}_{(x, y) \sim D}(l(g(x), y) | X = x) \right\} \]

- standard properties of the expectation

\[ \mathbb{E}_{(x, y) \sim D}(l(g(x), y) | X = x) = \sum_{y' \in Y} l(g(x), y') \mathbb{P}_{(x, y) \sim D}(Y = y' | X = x) \]

- pointwise minimization and \( l(y, y) = 0 \) gives the result
Remarks

**Discriminant versus Generative**

- **Bayes rule:**\[ P(Y = y | X = x) = \frac{P(X = x | Y = y)P(Y = y)}{P(X = x)} \]

- for a fixed \( x \), one can compare the \( P(X = x | Y = y)P(Y = y) \) rather than the \( P(Y = y | X = x) \)

- \( Y \) given \( X \): discriminant, \( X \) given \( Y \): generative

\( \mathcal{Y} = \{y_1, y_2\} \)

\[
g_i^*(x) = \begin{cases} 
  y_1 & \text{if } \frac{l(y_1, y_2)P(Y = y_2 | X = x)}{l(y_2, y_1)P(Y = y_1 | X = x)} \leq 1 \\
  y_2 & \text{in the other case}
\end{cases}
\]

Ratios of probabilities are sufficient to compute the optimal model
Outline

Standard supervised learning hypothesis

Optimal model

The Naive Bayes Classifier
Generative models

Generative model
▶ a model that explains both $X$ and $Y$
▶ as opposed to $Y$ given $X$
▶ one road to optimal models

Hypotheses
▶ $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_p$
▶ $\mathcal{Y}$ and all the $\mathcal{X}_j$ are finite sets

Probability distributions on $\mathcal{X} \times \mathcal{Y}$
▶ $|\mathcal{X}_1| \times |\mathcal{X}_2| \times \ldots \times |\mathcal{X}_p| \times |\mathcal{Y}| - 1$ parameters
▶ generally intractable
A simple model

Conditional independence

▶ explanatory variables are assumed independent given the target variable

▶ \( \perp \ 1 \leq j \leq P \ X_j | Y \) and thus

\[
P(X = x | Y = y) = \prod_{j=1}^{P} P(X_j = x_j | Y = y)
\]

Consequences

▶ only \( \left(1 + \sum_{j=1}^{P} (|X_j| - 1)\right) |Y| - 1 \) parameters

▶ easy estimation

▶ but very strong assumption
Naive Bayes

Categorical distribution

- arbitrary distribution on \( x_j = \{ u_1^{(j)}, \ldots, u_{|x_j|}^{(j)} \} \)
- parameter vector \( \Gamma = (\gamma_1, \ldots, \gamma_{|x_j|}) \)
- \( \mathbb{P}_{X \sim C(\Gamma)}(X = u_l^{(j)}) = \gamma_l \) and by extension \( \mathbb{P}_{X \sim C(\Gamma)}(X = u) = \gamma_u \)

Naive Bayes distribution

- \( \Gamma = (\Gamma_Y, (\Gamma_{j,y})_{1 \leq j \leq P, y \in Y}) \)
- distribution on \( \mathcal{X} \times \mathcal{Y} \)

\[
\mathbb{P}_{(X,Y) \sim NB(\Gamma)}(X = x, Y = y) = 
\mathbb{P}_{Y \sim C(\Gamma_Y)}(Y = y) \prod_{j=1}^{P} \mathbb{P}_{X_j | Y = y \sim C(\Gamma_{j,y})}(X_j = x_j | Y = y)
\]
Maximum Likelihood Estimation

**MLE**

- $\hat{\Gamma}_{MLE} = \arg \max_{\Gamma} P(D|\Gamma)$
- equivalently maximizing $\log P(D|\Gamma)$

**Naive Bayes log Likelihood**

- standard i.i.d. assumptions
- separability

$$
\log P(D|\Gamma) = \sum_{i=1}^{N} \log P(X,Y)_{NB}(X = X_i, Y = Y_i)
= \sum_{i=1}^{N} \sum_{j=1}^{P} \log P_{X_j|Y=Y_i}\sim C(\Gamma_{j,Y})(X_j = X_{ij}|Y = Y_i)
+ \sum_{i=1}^{N} \log P_{Y\sim C(\Gamma_Y)}(Y = Y_i)
$$
Separate estimations

- parameters of the different categorical distributions are independent
- \( \hat{\Gamma}_{MLE} \) can be computed block by block

\[
\hat{\Gamma}_{Y,MLE} = \arg \max_{\Gamma_Y} \sum_{i=1}^{N} \log P_{Y \sim C(\Gamma_Y)}(Y = Y_i)
\]

\[
\hat{\Gamma}_{j,y,MLE} = \arg \max_{\Gamma_{j,y}} \sum_{i=1, Y_i = y}^{N} \log P_{X_j \mid Y = y \sim C(\Gamma_{j,y})}(X_j = X_{ij} \mid Y = y)
\]

Consequences

- simple unidimensional estimation
- conditional frequencies
Frequencies

Target variable

\[ \forall y \in \mathcal{Y}, \gamma_{y} \text{MLE} = \frac{|\{i|Y_i = y\}|}{N} \]

Explanatory variables

\[ \forall y \in \mathcal{Y}, \forall j, \forall x \in \mathcal{X}_j, \gamma_{y,x} \text{MLE} = \frac{|\{i|Y_i = y \text{ and } X_{i,j} = x\}|}{|\{i|Y_i = y\}|} \]

Computational cost

Very efficient method: \( O(NP) \) (with a one pass algorithm)
Extensions

Infinite discrete set

- $X_j = \mathbb{N}$
- replace the categorical distribution by e.g. a Poisson distribution (or any distribution on $\mathbb{N}$)
- frequency based estimation, e.g. if $X_j \mid Y = y$ is a Poisson distribution with parameter $\lambda_{j,y}$ then

$$\hat{\lambda}_{j,y}^{MLE} = \frac{\sum_{i, Y_i = y} X_{i,j}}{|\{i \mid Y_i = y\}|}$$

Continuous variables

- $X_j = \mathbb{R}$
- same principle: choose a parametric distribution on $\mathbb{R}$, e.g. the Gaussian distribution
- block based estimation
Naive Bayes Classifier

Strategy

- estimate the parameters of a NB model $NB(\Gamma)$ for a data set
- approximate the data distribution by the $NB$ distribution i.e.
  \[ P(x, y) \sim D(x, y) \approx P(x, y) \sim NB(\hat{\Gamma}_{MLE})(x, y) \]
- use the approximation to compute the “optimal” classifier, i.e.
  \[ g_i^*(x) \approx \arg \min_{y' \in Y} \sum_{y' \neq y} l(y, y') P(x, y) \sim NB(\hat{\Gamma}_{MLE})(Y = y' | X = x) \]
- the classifier is optimal only for data distributed exactly according to $NB(\hat{\Gamma}_{MLE})$: this is seldom the case in practice!
Naive Bayes Classifier

Example

- as \( \mathbf{x} \) is fixed when one computes \( g_i^*(\mathbf{x}) \), only
  \[
P_{(\mathbf{x}, Y) \sim \text{NB}(\hat{\Gamma}_{MLE})}(Y = y', \mathbf{X} = \mathbf{x})
  \]
  is needed

- we have
  \[
P_{(\mathbf{x}, Y) \sim \text{NB}(\hat{\Gamma}_{MLE})}(Y = y, \mathbf{X} = \mathbf{x}) = 
  \]
  \[
P_Y \sim \text{C}(\hat{\Gamma}_Y_{MLE})(Y = y) \prod_{j=1}^{P} P_{X_j | Y = y \sim \text{C}(\hat{\Gamma}_{j,Y_{MLE}})}(X_j = x_j | Y = y)
  \]
  and thus
  \[
P_{(\mathbf{x}, Y) \sim \text{NB}(\hat{\Gamma}_{MLE})}(Y = y, \mathbf{X} = \mathbf{x}) = \hat{\gamma}_{Y,Y_{MLE}} \prod_{j=1}^{P} \hat{\gamma}_{j,Y,X_j_{MLE}}
  \]

- very simple frequency comparison!
Pros and Cons

+ very fast
+ handles mixed data easily
- limited predictive performances compared to state-of-the-art methods
- needs a very good set of explanatory variables

Best practices

▶ avoid using the NBC when frequencies are very close to 0 or 1
▶ use a variable selection method
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-sa/4.0/
Last git commit: 2021-01-19
By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)
Git hash: 97cfd0a9975cf193f5790845c00e476c1572a327
Changelog

- January 2018: clarified and reorganized
- December 2017: initial version