Lecture 6: Naive Bayes classifier

Introduction to Machine Learning

Sophie Robert

L3 MIASHS — Semestre 2

2022-2023

Probabilistic classifiers

Probabilistic classifiers

Probabilistic classifiers are classifiers that predict, given an observation of an input, a **probability distribution over a set of classes** (instead of simply the class like standard classifiers).

Probabilistic classifiers

Probabilistic classifiers

Probabilistic classifiers are classifiers that predict, given an observation of an input, a **probability distribution over a set of classes** (instead of simply the class like standard classifiers).

Given a record $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$ and a set of labels $y_i \in \mathcal{Y}$, provide an estimation of $\mathbb{P}(y_i|\mathbf{x})$ $y_i \in \mathcal{Y}$ (and assign most likely label to $\mathbf{x}$).

Probabilistic classifiers

Question

What is in your opinion one of the strength of working with a probabilistic model rather than a classification function ?

Probabilistic classifiers

Question

What is in your opinion one of the strength of working with a probabilistic model rather than a classification function ?

Possible native models are:

- Logistic regression
- Subtypes of neural networks
- Bayes classifiers

Probabilistic classifiers

Question

What is in your opinion one of the strength of working with a probabilistic model rather than a classification function ?

Possible native models are:

- Logistic regression
- Subtypes of neural networks
- Bayes classifiers

Non-probabilistic models can also be turned into a probabilistic classifier (SVM, trees . . .).

Main idea

The naïve Bayes classifier algorithm

The naïve Bayes classifier algorithm is a a probabilistic classifier based on applying Bayes' theorem with independence assumptions between the features.

Main idea

The naïve Bayes classifier algorithm

The naïve Bayes classifier algorithm is a a probabilistic classifier based on applying Bayes' theorem with independence assumptions between the features.

Each features **contribute to the class probability independently**: for example, the size and the weight of the dog contribute independently to its breed.

Mathematical framework

We want to estimate for each label $y_i \in \mathcal{Y}$ $\mathbb{P}(y_i|\mathbf{x})$ (*what is the probability of being label y_i given the data records ?*) .

Mathematical framework

We want to estimate for each label $y_i \in \mathcal{Y}$ $\mathbb{P}(y_i|\mathbf{x})$ (*what is the probability of being label y_i given the data records ?*) .
However, if n is large, the computation is infeasible.

Mathematical framework

We want to estimate for each label $y_i \in \mathcal{Y}$ $\mathbb{P}(y_i|\mathbf{x})$ (*what is the probability of being label y_i given the data records ?*) .
However, if n is large, the computation is infeasible.
Using the definition of conditional probabilities:

$$\mathbb{P}(y_i|\mathbf{x}) = \frac{\mathbb{P}(y_i, \mathbf{x})}{\mathbb{P}(\mathbf{x})}$$

$P(\mathbf{x})$ is a constant because $\mathbf{x}$ is given so we only need to find the value of $\mathbb{P}(y_i, \mathbf{x})$.

Mathematical framework

Using the definition of conditional probabilities iteratively,

Mathematical framework

Using the definition of conditional probabilities iteratively,

$$\begin{aligned}
\mathbb{P}(y_i, \mathbf{x}) =& \mathbb{P}(x_1, x_2, ..., x_n, y_i) \\
=& \mathbb{P}(x_1 | x_2, x_3, ..., y_i) \times \mathbb{P}(x_2, x_4, ..., y_i) \\
=& \mathbb{P}(x_1 | x_2, x_3, ..., y_i) \times \mathbb{P}(x_2 | x_3, x_4, ..., y_i) \times \mathbb{P}(x_3, x_4, ..., y_i) \\
=& \ ... \\
=& \mathbb{P}(x_1 | x_2, x_3, ..., y_i) \times \mathbb{P}(x_2 | x_3, x_4, ..., y_i) \times \mathbb{P}(x_n | y_i) \times \mathbb{P}(y_i)
\end{aligned}$$

Mathematical framework

We now make the hypothesis that each feature x_i are **conditionnally independant** given y_i (and only depends on the label y_i):

Conditional independence

Conditional independence describes situations where an observation is irrelevant: the probability of the hypothesis given the uninformative observation is equal to the probability without. If A is the hypothesis, B and C the observations, $P(A \mid B, C) = P(A \mid C)$

Mathematical framework

We now make the hypothesis that each feature x_i are **conditionnally independant** given y_i (and only depends on the label y_i):

Conditional independence

Conditional independence describes situations where an observation is irrelevant: the probability of the hypothesis given the uninformative observation is equal to the probability without. If A is the hypothesis, B and C the observations, $P(A \mid B, C) = P(A \mid C)$

$$\mathbb{P}(x_1|x_2, x_3, ..., y_i) = \mathbb{P}(x_1|y_i)$$

Mathematical framework

We now have:

$$\mathbb{P}(y_i, \mathbf{x}) = \mathbb{P}(y_i) \prod_{j=1}^{n} \mathbb{P}(x_j | y_i)$$

and :

$$\mathbb{P}(y_i | \mathbf{x}) \propto \mathbb{P}(y_i) \prod_{j=1}^{n} \mathbb{P}(x_j | y_i)$$

Mathematical framework

We now have:

$$\mathbb{P}(y_i, \mathbf{x}) = \mathbb{P}(y_i) \prod_{j=1}^{n} \mathbb{P}(x_j | y_i)$$

and :

$$\mathbb{P}(y_i | \mathbf{x}) \propto \mathbb{P}(y_i) \prod_{j=1}^{n} \mathbb{P}(x_j | y_i)$$

We then select the most probable class

$$\hat{y} = argmax_{i=1,\dots,k}(\mathbb{P}(y_i) \prod_{j=1}^{n} \mathbb{P}(x_j | y_i))$$

Mathematical framework

Can you guess why this algorithm can be called naive ?

Mathematical framework

Can you guess why this algorithm can be called naive ?

We have two terms to estimate:

- $\mathbb{P}(y_i)$: either assume class equiprobability or estimate using the frequency in training dataset
- $\mathbb{P}(x_j|y_i)$: we need to decide on a conditional law

Mathematical framework

Possible assumptions include:

- **If X_j is a continuous variable ($\mathbf{x_j} \in \mathbb{R}$)**, the continuous values associated within class i are distributed according to a Gaussian distribution parametrized with mean μ_i and variance σ_i

$$f(v \mid y_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(v-\mu_i)^2}{2\sigma_i^2}}$$

Mathematical framework

Possible assumptions include:

- **If X_j is a continuous variable ($\mathbf{x_j} \in \mathbb{R}$)**, the continuous values associated within class i are distributed according to a Gaussian distribution parametrized with mean μ_i and variance σ_i

$$f(v \mid y_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(v-\mu_i)^2}{2\sigma_i^2}}$$

- **If X_j is a categorial variable ($\mathbf{x_j} \in \{\mathbf{0}, ..., \mathbf{K}\}$)**, the probability can be estimated as the proportion of values within class:

$$\mathbb{P}(x = j | y_i) = \frac{N_{ji}}{N_i}$$

Example

Training dataset:

Height	Weight	Tail	Label
45	30	0	Labradoodle
30	25	1	Labradoodle
40	35	1	Labradoodle
20	15	0	English cocker
22	18	1	English cocker
25	20	1	English cocker

Individual to classify

Height	Weight	Tail	Label
25	31	1	?

Example: training the model

Training the model

Training the model consists in computing **statistical estimators over the population**.

For the labradoodle population:

	Height	Weight
Mean	38.33	30
Var	38.89	16.67

For the cocker population:

	Height	Weight
Mean	22.33	17.66
Var	4.22	4.22

Example: solution

Estimate:

$$\begin{aligned}&\mathbb{P}(\text{labradoodle} \mid \text{height} = 25, \text{weight} = 31, \text{tail} = 1)\\ &\propto \mathbb{P}(\text{labradoodle}) \times \mathbb{P}(\text{height} = 25|\text{labradoodle})\\ &\times \mathbb{P}(\text{weight} = 31|\text{labradoodle})\\ &\times \mathbb{P}(\text{tail} = 1|\text{labradoodle})\end{aligned}$$

$$\mathbb{P}(\text{labradoodle}) = \frac{1}{2}$$

$$\mathbb{P}(\text{height} = 25|\text{labradoodle}) = \frac{1}{\sqrt{2\pi \times 38.89}}\, e^{-\frac{(25-38.33)^2}{2\times 38.89}} = 0.006$$

$$\mathbb{P}(\text{weight} = 31|\text{labradoodle}) = \frac{1}{\sqrt{2\pi \times 16.67}}\, e^{-\frac{(31-30)^2}{2\times 16.67}} = 0.09$$

$$\mathbb{P}(\text{tail} = 1|\text{labradoodle}) = \frac{2}{3}$$

$$\mathbb{P}(\text{labradoodle} \mid \text{height} = 25, \text{weight} = 31, \text{tail} = 1) = 0.00017$$

Example: solution

Estimate:

$$\begin{aligned}&\mathbb{P}(\text{cocker} \mid \text{height} = 25, \text{weight} = 31, \text{tail} = 1)\\&\propto \mathbb{P}(\text{cocker}) \times \mathbb{P}(\text{height} = 25|\text{cocker})\\&\times \mathbb{P}(\text{weight} = 31|\text{cocker})\\&\times \mathbb{P}(\text{tail} = 1|\text{cocker})\end{aligned}$$

$\mathbb{P}(\text{cocker}) = \frac{1}{2}$

$\mathbb{P}(\text{height} = 25|\text{cocker}) = \frac{1}{\sqrt{2\pi \times 4.22}} e^{-\frac{(25-22.33)^2}{2\times 4.22}} = 0.08$

$\mathbb{P}(\text{weight} = 31|\text{cocker}) = \frac{1}{\sqrt{2\pi \times 4.22}} e^{-\frac{(31-17.66)^2}{2\times 4.22}} = 1.31e - 11$

$\mathbb{P}(\text{tail} = 1|\text{cocker}) = \frac{2}{3}$

$$\mathbb{P}(\text{cocker} \mid \text{height} = 25, \text{weight} = 31, \text{tail} = 1) = 0.00$$

Hyperparameters

Hyperparameters

What **hyperparameters*** do the naive Bayes classifier require ?

Advantages and limits

Limits:

Advantages and limits

Limits:

- Strong independance hypothesis (but in practice, naive bayes behave rather well)
- Unable to classify unknown classes that do not show in training set (always sets it to 0, except in the case of artificial equiprobability)

Advantages and limits

Limits:

- Strong independance hypothesis (but in practice, naive bayes behave rather well)
- Unable to classify unknown classes that do not show in training set (always sets it to 0, except in the case of artificial equiprobability)

Advantages:

Advantages and limits

Limits:

- Strong independance hypothesis (but in practice, naive bayes behave rather well)
- Unable to classify unknown classes that do not show in training set (always sets it to 0, except in the case of artificial equiprobability)

Advantages:

- Extends naturally to multi-class
- Naturally deals with categorical variables

Other probabilistic classifiers

One of the most famous probabilistic classifier is **logistic regression**: probability distribution is expressed as the logit of the linear combination of features.

Other probabilistic classifiers

One of the most famous probabilistic classifier is **logistic regression**: probability distribution is expressed as the logit of the linear combination of features.

Using **softmax function** as activation layer in **neural networks** transforms output into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers.

Questions

Questions ?