Year 2012-2013

M1 Informatique

TD 6

Outils pour l'exploration de données - Naive Bayesian Classifier

1 Data set

For this TD we will work with a variation of the data set `algae` (you can download it from CELENE). In this variation, called `algaeClassification.txt`, from the seven target variables (`a1` to `a7`) I kept only `a1`. Also, I transformed `a1` into a qualitative variable with three possible outcomes `feqLow`, `feqMedium` and `feqHigh`. So, now, we have a classification problem. The aim is, based on the other attributes, predict if an observation will present a low, medium or high frequency of alga `a1`.

Task 1: Supposing that the data set `algaeClassification.txt` is in your current working directory, load it into R using:

```
Temp_algae <- read.table("algaeClassification.txt", header=TRUE)
```

- Preprocess `Temp_algae` by removing any observation with more than 20% of missing values. Then, replace the remaining missing values by the mode (qualitative attributes) and the median (numeric) of the respective attribute. In order to accomplish the task, use the functions for replacing missing values you had to define in the previous TD.
- Store the "clean" data in `algae`.

Note: to save typing, one can use the function `attach(algae)`.

2 Naive Bayesian classifier

Consider each observation (instance/example) in the data set (sample) to be an n-dimensional vector of attributes (variables) values:

$$\mathbf{x}_=(x_1, x_2, x_3, ..., x_n)$$

The `algaeClassification.txt` data has 11 attributes (n=11), excluding the class attribute:

$$\mathbf{x}=(\texttt{season}, \texttt{size}, \texttt{speed}, \texttt{mxPH}, \texttt{mnO2}, \texttt{Cl}, \texttt{NO3}, \texttt{NH4}, \texttt{oPO4}, \texttt{PO4}, \texttt{Chla})$$

In a Bayesian classifier that assigns each data instance to one of m classes C_1,C_2, . . . ,C_m (m=3 for `algaeClassification.txt`), a data instance $\mathbf{x}$ is assigned to the class for which it has the highest posterior probability conditioned on $\mathbf{x}$, that is, the class which is most probable given the prior probabilities of the classes and the data $\mathbf{x}$.

In other words, $\mathbf{x}$ is assigned to class C_i if and only if $P(C_i|\mathbf{x}) > P(C_j|\mathbf{x})$ for all j such that $1 \leq j \leq m, j \neq i$. In fact, a naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem:

$$P(C_i|\mathbf{x}) = \frac{P(\mathbf{x}|C_i) * P(C_i)}{P(\mathbf{x})} \tag{1}$$

Since $P(\mathbf{x})$ is a normalizing factor which is equal for all classes, we need only to maximize the numerator $P(\mathbf{x}|C_i) * P(C_i)$ in order to do the classification. We can estimate both the values we need, $P(\mathbf{x}|C_i)$ and $P(C_i)$, from the data used to build the classifier.

Very important: Naive Bayesian classifier assumes that the effect of the value of an attribute x_k on a given class C_i is independent of the values of other attributes. This assumption is called *class conditional independence*. This concept will be fundamental for the most of the calculations we will make in the next sections.

2.1 Estimating the class prior probabilities

We can simply estimate the prior probabilities, $P(C_i)$, of the classes from their frequencies in the training data (sample).

Task 2: For the data set `algaeClassification.txt`, estimate the prior probabilities for `feqLow`, `feqMedium` and `feqHigh`.

2.2 Estimating the probability of the data given the class: qualitative attributes

In general, it can be very computationally expensive to compute $P(\mathbf{x}|C_i)$. For example, in the case of qualitative attributes, if each attribute x_k in $\mathbf{x}$ can have one of r values, there are r^n combinations to consider for each of the m classes.

However, as already mentioned, in order to simplify the calculation, the assumption of class conditional independence is made. That is, for each class, the attributes are assumed to be independent. The assumption allows us to write:

$$P(\mathbf{x}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{2}$$

that is, the product of the probabilities of each of the values of the attributes of $\mathbf{x}$ for the given class C_i

For the case of qualitative attributes we can simply estimate each $P(x_k|C_i)$ from the frequencies in the training data (sample).

Task 3: For the data set `algaeClassification.txt`, estimate the following probabilities:

- P(`season=winter`|C=`feqLow`)
- P(`size=medium`|C=`feqHigh`)
- P(`speed=slow`|C=`feqMedium`)

Hint: have a look at the function `table( )` and `prop.table( )`.

2.3 Estimating the probability of the data given the class: quantitative attributes

In the case of the "classic" naive Bayesian classifier, numerical variables need to be transformed to their categorical counterparts (binning) before constructing their frequency tables. However, a more rigorous method is to use probability density function (`pdf`) values, where we preserve the continuous values as such.

For example, it is common to assume that the probability distribution for an attribute follows a normal or Gaussian distribution. If it is known to follow some other distribution, such as Poisson's, the equivalent probability density function can be used.

In the case of the probability density function for the normal distribution, we can define it by two parameters: mean (μ) and standard deviation (σ).

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Mean

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5}$$

Standard deviation

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal distribution

We can easily estimated from the data these two parameters and, as consequence, we can estimate the `pdf` value at any given decision point (see function `dnorm( )`).

Task 4: For the data set `algaeClassification.txt`, estimate the following probabilities:

- p(`mxPH=8.25`|C=`feqLow`)
- p(`mnO2=10.4`|C=`feqHigh`)
- p(`PO4=57.833`|C=`feqMedium`)

2.4 Classifying a new instance

Suppose we receive the observation below, for which we do not the class, and we would like to assign a class to it using a Naive Bayesian classifier. The classifier should be built using the data in `algaeClassification.txt`.

x=(`season=autumn`, `size=large`, `speed=low`, `mxPH=8.53`, `mnO2=11.1`,
`Cl=63.292`, `NO3=1.726`, `NH4=227.60001`, `oPO4=84.3`, `PO4=146.452`, `Chla=21.22`)

Task 5: To which class would the classifier assign this observation? (See equation 2). **Important**: since it would be extremely laborious to calculate the probabilities for all the 11 attributes, let's "simplify" the task and consider only the first four attributes in our calculation, that is, x=(`season=autumn`, `size=large`, `speed=low`, `mxPH=8.53`, `mnO2=11.1`).

3 Assumptions

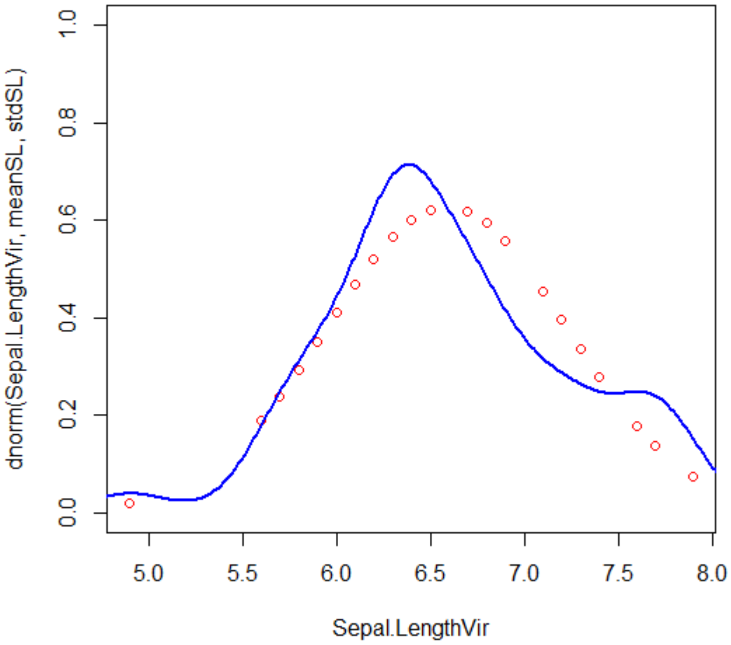
As mentioned in section 2, Naive Bayes classifier assumes that the effect of the value of an attribute x_k on a given class C_i is independent of the values of other attributes.

Task 6 If two random variable are independent, their covariance is zero (the correlation as well) — but two random variable can have zero covariance/correlation but still be dependent. Use this fact to analyze if the assumption of *class conditional independence* is true for the `algaeClassification.txt` data set.

- More specifically, check this for the values of attributes `mxPH` and `mnO2`, given that the class is `feqLow`.

Task 7 We have modeled the numeric attributes as if they were normally distributed. Does this assumption really hold for our data set? We can try to "visualize" this by drawing a plot of the `pdf` of the normal distribution modeling the attribute, overlapping with the result of applying the `density()` function. For example, for the case of the `iris` data set, the next figure shows the plot resulting of applying the values of the attribute `Sepal.Length` **given the class** `virginica` to the `dnorm()` modeling them (dotted red points), overlapping with the output of `density()` (`lines()`) — blue curve.

- Draw a similar set of graphics for `mxPH`, given that the class is `feqLow`.
- Draw a similar set of graphics for `mnO2`, given that the class is `feqLow`.
- To what degree do you think that the "normality" assumption really holds for the values of these two attributes?



4 Missing values: revisited

In the last TD you were required to implement different functions to deal with missing values. One of theses functions, `manyNAs(x,t)`, removes from the data frame `x` any observation that has more `t`% of its values as `NA`.

- Based on the previous code, create a R function (`manyAtrNAs(x,t)`) that removes from the data frame `x` any *attribute/variable* that has more `t`% of its values as `NA`.