Universidad de La Sabana
Facultad de Ingenería

Felix Mohr
2022.1

Business Intelligence

- Sheet 3 -

> "El camino que les espera por recorrer es largo y dificil, pero todo lo bueno se obtiene con esfuerzo y ese es el caso de esta materia, ademas del conocimiento que obtendran a partir de los contenidos impartidos, esta materia los retara constantemente y los hara crecer como ingenieros! "
>
> — Estudiante graduado del curso 2021.1

Exercise 1 (3 points)

1. Instead of using a multivariate Bernoulli variable, we could replace an m-valued categorical variable by a single numerical variable and use numbers 1,..,m for the possible values of the variable. Create an example that shows that two different such numeric replacements of the same nominal variable can lead to *different distances* between two points. What do you conclude from this observation?

   Hint: Consider a database with only a single categorical and no numerical attribute.

2. To see that Bernoulli encoding overcomes this problem, consider a dataset with d categorical and no numerical attributes. You can *not* assume a particular value for d like 1 or 2. Now show that the distance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in this database is

   a) independent of how the Bernoulli encoding is realized, i.e., to which vector $\mathbf{e}_i$ an attribute value is mapped.

   b) at most $\sqrt{d}$ for any Bernoulli encoding

Exercise 2 (5 points) Use the `credits.csv` for all the tasks in this exercise (iris datasets only for some unit tests). You might stumble over some difficulties and may want to check the `replace` and `fillna` function of Pandas data frames to get rid of missing values. **Do not use** the `get_dummies`, `digitize`, `cut`, `crosstab` (or any similar) function in this exercise.

1. a) Write a function `binarizeCategoricalAttributeVector(column)` that takes a categorical attribute vector (do *not* assume a categorical Pandas type) and *returns* an $n \times m$ dimensional numpy array, where m is the number of categorical values occurring in the column and n is the length of the original column.

   b) Then write a function `getCategoricalAttributes(df)` that returns a list of column names of a Pandas DataFrame that contain non-numeric values.

   c) Finally, write a function `readFrameAsMatrix(df)` to convert a given DataFrame into a purely numeric nd array such that each categorical attribute with m values is converted into m binary attributes (columns).

2. This exercise is about discretizing attributes.

a) Write a function `discretizeBasedOnThresholds(column, thresholds, names=None)` that takes a vector of observations, the desired thresholds and optionally names for the bins; if no names are given, name the bins `c0`,`c1`,... The first bin contains all instances with values at most the lowest threshold, and the last bin contains all instances with values greater than the biggest threshold. Hence, the number of bins is the number of thresholds + 1. The function should return a vector with the discretized values.

b) Write functions `discretizeEqualLength(column, k, names = None)` and `discretizeEqualFrequency(column, k, names = None)` that convert a numeric attribute vector into a discrete attribute vector, applying the respective technique for a specified number of bins or frequency. The parameter k is for the number of bins. Use the above function to realize these two.

Hint: Only compute thresholds and then use the first function.

3. In this exercise we want to check the independence of categorical attributes.

a) Write a function `getContingencyTable(M)` that receives a 2D numpy array with *two* columns and computes a table containing the *absolute* observed frequenties of the pairs of occuring values.

b) Write a function `computeExpectedOccurrences(ct)` that receives a contingency table and computes a table containing, for each pair of values, the number of occurences one would expect given independency of the attributes.

c) Write a function `computeChiSquare(M)` that receives a 2D numpy array with two discrete columns and computes the χ^2 score of the two attributes.

d) Write a function `checkIndependence(df, c1, c2, alpha)` that receives a Pandas DataFrame and the *names* of two columns and that returns `true` iff the independence hypothesis is sustained in a χ^2 test (considering the appropriate degree of freedom) for a given confidence threshold α for the p-value.

e) Then check independence hypothesis for all pairs of categorical variables of the credit dataset. Plot the χ^2 curve for every pair of categorical variables with the respective (given) critical point (we assume $\alpha = 0.01$).

Are there pairs of independent variables?