Data Mining Classification Algorithms: an analysis

Daniel Hanspeter 6129, Daniel Graziotin 4801, Thomas Schievenin 5701

October / November 2010

Abstract

Blah blah blah a couple of words regarding the choice of the two datasets. Blood transfusion missing class, for testing prediction. Census Income has class, good for benchmarking classification techniques. Expand this! Talk about sections, summary of one page of the four algorithms as last page

1 Data Examination

1.1 Blood Transfusion

1.2 Census Income

The dataset contains 15 attributes and a total of 32561 instance. The following is a representation of the attributes and their types, extracted from the ARFF file:

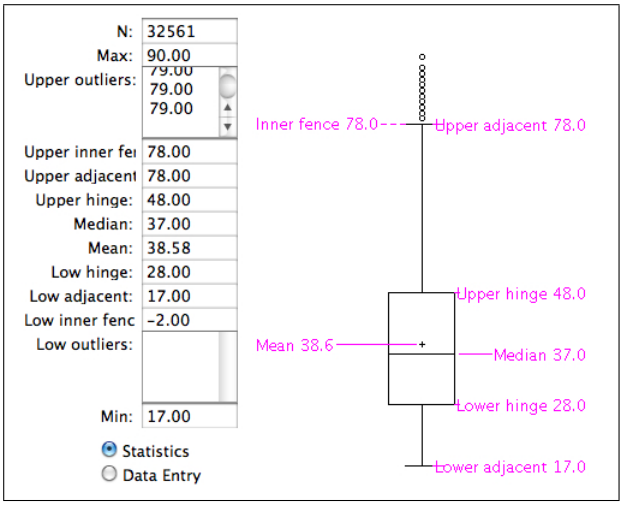
```
age NUMERIC
workclass {Private, Self-emp-not-inc, Federal-gov, [..], Never-worked}
fnlwgt NUMERIC
education {Bachelors, Some-college, [..], Preschool}
education-num NUMERIC
marital-status {Married-civ-spouse, Divorced, [..], Married-AF-spouse}
occupation {Tech-support, Craft-repair, [..], Armed-Forces}
relationship {Wife, Own-child, Husband, [..], Unmarried}
race {White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black}
sex {Female, Male}
capital-gain NUMERIC
capital-loss NUMERIC
hours-per-week NUMERIC
native-country {United-States, Cambodia, England, [..]}
class { >50K, <=50K}
```

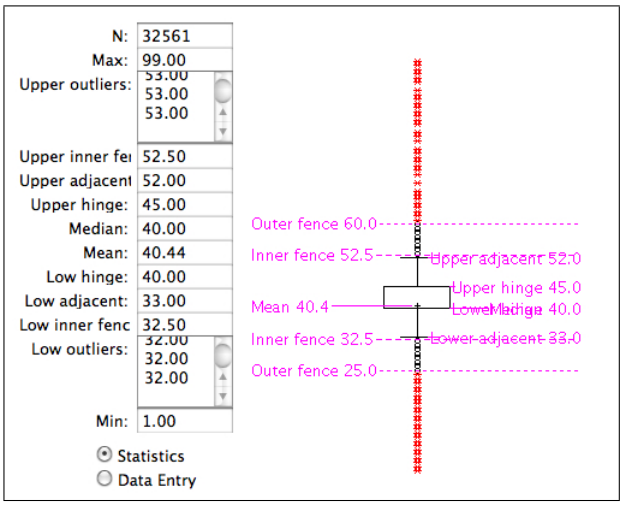
Education and Education-Num seems are referring to the same value, where education-num is a numerical representation of education.

Attribute fnlwgt does not appear to make any sense and will be ignored.

There are missing values for attributes workclass (6%), occupation (6%), native-country (2%). Two possible strategies to deal with the missing values are complete removal of the instances having missing values and substitution of the missing values with the mode value of the attribute. Both of them will be analyzed.

The two attributes age and hours-per-week have been selected. The following are their boxplot representation:



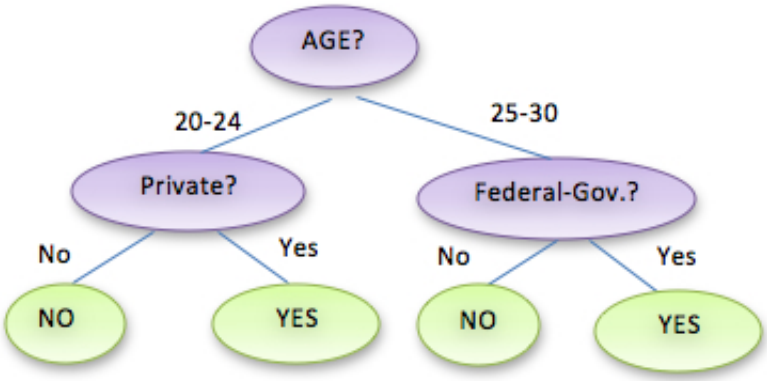


There are outliers for both the selected attributes. They will be treated with discretization techniques.

Data similarity:

Data mining tasks: an interesting task could focus around the income class. For example we could construct a tree which tells us if a 20-24 or a 25-30 years old with different work class can be mapped in a >50K income class.

E.G Age + Work => Class income: >50K /<=50K



2 Analysis of Techniques

For analyzing the classification algorithms, we decided to prepare an evaluation framework containing variables commonly adjustable with each algorithm.

For preparing it, we first decided how to pre-process the data that can be evaluated by all four algorithms. All other settings not commonly shared by the algorithms will be kept to the values that Weka proposes. Further experimenting with the single algorithms will be performed in the next section.

Missing Values and Outliers

We decided to simply substitute the missing values of instances with the mode value of each attribute, as the method of removing instances is discouraged. Outliers will be treated by applying discretization.

Chosen Datasets

We will use the four algorithms on both the supplied data sets at full.

Approach The framework will be run as described in this paragraph. The four classification models will be run in this ways:

1. Without any other preprocessing techniques

2. By substituting missing values
3. By treating outliers
4. By treating outliers and substituting missing values
5. By removing fnlwgt and education-num
6. Removing fnlwgt, education-num, substituting missing values
7. Removing fnlwgt, education-num, treating outliers
8. Removing fnlwgt, education-num, substituting missing values, treating outliers

The best options for each classification algorithm will be chosen and used for further experiments.

2.1 Results

The following is the list of the results of the tests run using the configurations presented in the previous paragraph. We highlighted with different colors the casistics in which each algorithm performed better. We also enclosed in square brackets the best performing algorithm for each run.

1. J48 87.8566%; NaiveBayes 83.4465%; Logistics 84.896%; k-nearest neighbour: 99.9969%
2. J48 88.0102%; NaiveBayes 83.3144%; Logistics 84.896%; k-nearest neighbour: 99.9969%
3. J48 87.8566%; NaiveBayes 83.4557%; Logistics 84.896%; k-nearest neighbour: 99.9969%
4. J48 88.0102%; NaiveBayes 83.2837%; Logistics 84.896%; k-nearest neighbour: 99.9969%
5. J48 87.0858%; NaiveBayes 82.4729%; Logistics 85.1663%; k-nearest neighbour: 97.515%
6. J48 87.1380%; NaiveBayes 82.3193%; Logistics 85.1663%; k-nearest neighbour: 97.5615%
7. J48 85.1172%; NaiveBayes 81.1032%; Logistics 85.326%; k-nearest neighbour: 92.9179%
8. J48 85.1172%; NaiveBayes 80.9588%; Logistics 85.326%; k-nearest neighbour 92.9517%

K-nn (k=1) has been the most well-performing algorithm in general.

Conclusions

blah blah

	Decision tree	Naïve Bayesian Classifier	Logistic Regression	KNN classifier
Input Parameter	The attributes of a tuple	The posterior probabilities of Hypothesis H based on additional information	The measure of the total contribution of all independent variables used in the model.	A new unknown tuple for which a class has to be assigned.
Principle	The attributes of a tuple are tested against the decision tree and a path is traced from the root to a leaf node which holds the prediction for that tuple	Given a tuple X, the classifier will predict that X belongs to the class having the highest posterior probability conditioned on X.	Given x representing the exposure to some set of risk factors, LR predicts the probability of occurrence of an event by fitting data to a logistic curve, f(x), which represents the probability of a particular outcome	Nearest-neighbor classifiers compare a given test tuple with training tuples that are similar and described by n attributes and are stored in n-dimensional space ➔ Find the k-nearest tuples from the training set to the unknown tuple
ALG. FORMULA	At start, all the training tuples are at the root. Then, tuples are partitioned recursively based on selected attributes. The test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain). We stop when all samples for a given node belong to the same class or there are no remaining attributes for further partitioning ➔ majority voting.	At start, compute P(C) The prior probability ofC_i. Each class can be computed based on the training tuples. Then compute each independent probability for attribute xi in reference with Class C and multiply P (X_i|C). Example: The probability of class C to be yes is P(C) = 9/14. The amount of attribute Xi being test with respect to C is $P(x_i|C_i) = \frac{xi=test}{Ci=yes} = \frac{2}{9}$ Finally, compute $P(X|C_i)P(C_i)$ for each class ➔ The naïve Bayesian Classifier predicts C=yes for tuple X	Since x is defined as $x = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$ the LR is $\frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_k X_k}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_k X_k}}$ Estimate the parameters using the Maximum Likelihood Function and then by computing the partial derivatives of the log likelihood, equate each partial derivative to zero, and solve the resulting nonlinear equations	The ALG assigns for the unknown tuple the most common class among its k-nearest neighbor. When k=1 the unknown tuple is assigned the class of the training tuple that is closest to it. To measure the distance we can use the Euclidean distance $\sqrt{\sum_{i=1}^{n}(x_{1i}-x_{2i})^2}$ **Choose K:** If k=1 the classification will be 1:1 sensitive to other data. If k=n we'll suffer high noise. Go through various K's and choose one giving lowest misclassification error!