Data Mining

Lab 2: Data pre-processing

Summary: Preparing the data for the data mining tasks:

1. Cleaning (e.g. deal with errors and missing values)
2. Transformation
    a. Conversion between types (e.g. discretization, binarization)
    b. Scaling
    c. Standardization
3. Reduction
    a. Feature selection
    b. Instance selection

1. Data cleaning

The data can contain erroneous or missing values caused by measuring devices malfunctioning, human errors, not provided info (e.g. privacy issues). In some cases these situations can be treated in an automatic way:

a) Values which are outside normal ranges (e.g. negative age, very large or very small body temperature etc). In such cases there are several approaches:
   - Remove the attributes which contain such values (e.g. by using Filters->Unsupervised->attribute->NumericCleaner)
   - Remove the instances which contain values which are not valid (e.g. by using Filters->Unsupervised->Instance->SubsetByExpression)
   - Consider the value as a missing one

b) Missing values can be treated in different ways:
   - Consider the missing value as a specific case which has a particular value assigned (e.g. like "?" in Weka files) – in the case of nominal attributes the value assigned to a missing field is interpreted as a usual value and it can be involved in the classification models (e.g. in the classification rules). This is the default approach in Weka.
   - Remove the instances which contain missing values (e.g. by using Filters->Unsupervised->Instance->SubsetByExpression; the expression can be not ismissing(ATT2) to specify that only the instances without missing values for attribute 2 are preserved in the dataset)
   - Replace the missing fields with values estimated based on the values corresponding to other instances in the dataset (e.g. the average value of the corresponding attribute over the entire dataset or over the most similar instances – similarity is measured with respect to the other attributes). Such an approach is called imputation.

Exercise 1:

a) Open in Weka the file "autos.arff" and remove all instances having values of the attribute 25 (price) larger than 10000 (Hint: use Filters->Unsupervised->Instance->SubsetByExpression with the expression ATT25<=10000)

b) Open in Weka the file "breast-w.arff". Identify which attributes contain missing values and remove the instances which are characterized by missing values. (Hint: use Filters->Unsupervised->Instance->SubsetByExpression with the expression not

ismissing(ATTx) where x should be the number of the attribute which contains missing values)

2. Transformations

a) Conversion between types

- Discretization: transformation of the attributes taking values in a continuous range (e.g. real) in attributes which take discrete values (e.g. integer, ordinal). The simplest approach is when the inteval of attribute values [min, max] is divided in r subintervals (e.g [min, min+h), [min+h, min+2h), …[min+(r-1)h,max], where h=(max-min)/r) and each interval will be associated to a discrete value (e.g. 1, 2, 3,….r).
- Binarization: used to transform a nominal attribute in a set of binary or logical attributes or to transform a set of transaction items into binary attributes. For instance if an attribute A has the values {v1,v2,v3} then it will be replaced with 3 binary attributes A1, A2 and A3.

b) Scaling

- useful when different attributes have values in highly different ranges (e.g. one attribute takes values in [-0.1, 0.2] and another one takes values in [10000, 20000])
- the simplest scaling method is the linear scaling which uses a linear function to transform all values v of an attribute A in [0,1]:

  $s(v) = \frac{v - \min_A}{\max_A - \min_A}$, where min and max correspond to the minimal and maximal values of the attribute

c) Standardization

- useful when we are interested in measuring how much are the data concentrated around their average
- by standardization, a value v is transformed as follows:
- $st(v) = \frac{v - avg(A)}{stdev(A)}$, where avg(A) denotes the average of the values of attribute A and stdev(A) is the standard deviation of the values of attribute A

Exercise 2: Open in Weka the file "breast-w.arff"

a) Transform all numerical values in discrete values by using: (i) 10 subintervals; (ii) 5 subintervals and save the results in files "breast-w-discrete10.arff" and "breast-w-discrete5.arff" respectively. Hint: use Filter->Unsupervised->Attribute->Discretize (set the number of subintervals as value of parameters bins (-B)

b) Use the Weka-Experimenter to analyze the accuracy of the following classifiers: ZeroR, OneR, J48, NaiveBayes on the three datasets "breast-w.arff", "breast-w-discrete10.arff" and "breast-w-discrete5.arff"

Exercise 3: Open in Weka the file "car.arff".

a) Transform all nominal attributes through binarization (Filter->Unsupervised->Attribute->NominalToBinary) and save the transformed data in file carBinary.arff

b) Use the Weka-Experimenter to analyze the accuracy of the following classifiers: ZeroR, OneR, J48, NaiveBayes on the two datasets car.arff and carBinary.arff

Reminder from Lab 1: how to use the Experimenter

1. Select Experimenter from the starting Weka panel
2. Click New button to create a new experiment
3. Add datasets (e.g. car.arff and carBinary.arff
4. Add algorithms: zeroR, oneR, J48, naiveBayes (oneR and zeroR are from the "Rules" group, naiveBayes is from the "Bayes" group and J48 is from the "Tree" group)
5. Run the experiment
6. Analyze the results:
    a. Click on Experiment
    b. Perform the statistical test
    c. Interpret the results of the statistical test (using zeroR as reference method)
        i. v/ /* should be interpreted as: v= number of cases (datasets) on which the current method is better than the reference method; / /= number of cases (datasets) on which the current method is not significantly different than the reference method; *= number of cases (datasets) on which the current method is wors than the reference method;

Exercise 3. Open in Weka the file "iris.arff" and "breast-w.arff"

a) Apply scaling to all attributes (Hint: use Filters->Unsupervised->Attribute->Normalize) and save the files as "iris_scaled.arff" and "breast-w_scaled.arff". The effect of Weka normalization is the fact that the attribute values are scaled
b) Apply standardization to all attributes (Hint: use Filters->Unsupervised->Attribute->Standardize) and save the file as "iris_standardized.arff" and "breast-w_standardized.arff"
c) Analyze the impact of scaling and standardization on the performance of the following classifiers: ZeroR, OneR, J48, Bayes, IB1, MultilayerPerceptron (remark: change to False the NormalizeAttributes option for MultilayerPerceptron)

Exercise 4. Multiply all values of the first attribute in "breast-w.arff" by 100 (Hint: use Filters->unsupervised->attribute->MathExpression) and then apply all operations specified in Exercise 3.

3. Data reduction

Motivation: some attributes or instances could be irrelevant or redundant. For instance an attribute having the same value over the entire dataset is not relevant and should be ignored. Two highly correlated attributes (e.g. A1=2*A2) are redundant, as it is enough to use only one of them

- Reducing the number of attributes can reduce the computational cost but also improve the performance of a data mining model (e.g. classifier)
- Reducing the number of instances in the training set can reduce the computational cost of the training process or of the classification process (in the case of lazy classifiers) but also improve the quality of a classifier especially in the case of unbalanced data sets (the number of examples in a class is significantly larger than the number of data in another class)

Attribute reduction can be done through:

- Attribute selection:
    - Filter-based selection (the selection is done by analyzing only the properties of data and it is independent of a data mining task)

- Wrapper-based selection (the selection is done such that the selected attributes are the most relevant ones in the context of a given data mining task)

- Data projection on a space of a smaller size (e.g. Principal Component Analysis)

Instance reduction can be done through:
- Instance removal (based on some specific rules)
- Instance sampling (using a random sampler with or without replacement)

3.1. Filter-based attribute selection

Attribute selection can be done in at least two ways:
- By searching the space of attribute subsets (in the case of n attributes the search space is of cardinality 2^n-2 – we ignore the empty set and the full set)
- By ranking the attributes according to their relevance and take the most relevant ones (in the case of n ranked attributes A1,A2,….An, there are n-1 variants to analyze: {A1}, $\{A_1,A_2\}$, $\{A_1,A_2,A_3\}$….{ $A_1,A_2,\ldots,A_{n-1}$}

In each case it should be specified:
- A measure of relevance of an attribute or a subset of attributes
  - Unsupervised case: the class information is not used to evaluate the attribute relevance (the analysis is mainly based on the correlation or the similarities between attributes)
  - Supervised case: the class information is not used to evaluate the attribute relevance (the analysis is based on the correlation between attribute values and classes)
- A search technique:
  - Exhaustive search (all subsets of attributes are analyzed)
  - Greedy search (forward/backward): add/remove at each step the best/worst attribute
  - Random search
  - Rank-based search (the attributes are selected in the order specified by a ranking)

Exercise 5. Open the iris.arff file

a) Apply a attribute subset selection (e.g. Select attributes -> CfsSubsetEval, GreedyStepwise) and save the reduced dataset (right click on the result and Save reduced date)
b) Apply a attribute ranking method (e.g. Select attributes -> InfoGainAttributeEval, Ranker). Compare the result with that obtained at a).
c) Analyze the impact of the data reduction on the performance of classifiers (using the Experimenter as in exercises 2-3.

3.2. Data projection – Principal Component Analysis

Aim: transform the data (by changing the system of coordinates) such that the correlation between attributes is removed and then keep only the attributes corresponding to the coordinate axes which capture the most of data variability. This transformation can be done by using Principal Component Analysis which consists of the following steps:

- Compute the covariance matrix C of the data set (if there are n attributes then the matrix will have the size nxn); in practice before computing the covariance matrix the data are centered (the average of the dataset is substracted from each data instance)
- Compute the eigenvalues and the eigenvectors of C
- Sort the eigenvalues decreasingly
- Select m<n eigenvectors corresponding to the largest eigenvalues; m is usually chosen such that the transformed data preserve most of the original data variability (e.g. 95%)
- Project the data on the space defined by the m eigenvectors

Exercise 6. Open the iris.arff file

a) Apply a Principal Component Analysis on the data (e.g. Select attributes -> Principal Components, Ranker and save the reduced dataset (right click on the result and Save transformed data)
b) Visualize the transformed data and compare them with the initial ones.
c) Analyze the impact of PCA transformation on the performance of classifiers using the Experimenter tool (see Exercises 2-3).

Exercise 7. Apply the PCA transformation to the data "arrhythmia.arff" and analyze the results.