IFC FOR WEKA: SOFTWARE IMPLEMENTATION

The IFC-NLR data mining methodology introduced by Kaufmann & Meier (2009) has been implemented by Graf (2010) as a supervised attribute filter in the WEKA machine learning workbench (Hall, Frank, Holmes, Pfahringer, Reutemann, & Witten, 2009). The aim of this implementation is to enable the use of IFC-NLR in a typical data mining process. This IFC-Filter allows the evaluation of the algorithm. In this section, WEKA is presented, and the software implementation is described. A description of the functionality of the algorithm will give the necessary background. A use case of the IFC-Filter in a data mining process will be presented, and a user’s guide will introduce the application of the software.

Software use case

WEKA can be used for data mining in order to create predictive models based on customer data. The IFC-Filter can facilitate visualization of the association between customer characteristics and the target class, and it can improve predictive performance of customer scoring by transforming attribute values into inductive fuzzy membership degrees, as explained above. Thus, it can perform membership function induction (MFI) and inductive attribute fuzzification (IAF), as shown in Figure 1.

The IFC-Filter transforms sharp input data into membership degrees which indicate the inductive support for the conclusion that the data record belongs to the target class. In order to do so, first, membership functions are induced from data and optionally displayed to the screen. Then, these functions are applied to fuzzify the original attributes. Visualization and Prediction based on the concepts of MFI and IAF are two main use cases for the IFC-Filter software in the data mining process.

In order to visualize associations between variables, inductive membership functions can be plotted with the method IFC-NLR described earlier. Thus, for every analytic variable, a function mapping from the variable’s domain into a degree of membership in the inductive fuzzy target class is displayed graphically. This plot gives intuitive insights about associations between attribute values and the target class.

For prediction, the transformation of crisp attribute values into inductive membership degrees can enhance the performance of existing classification algorithms. The IFC-Filter transforms the original attribute values into inductive membership indicating target class likelihood based on the original value. After that, a classical prediction algorithm such as logistic regression can be applied to the transformed data and to the original data in order to compare the performance. It is possible that IAF improves prediction. If that is the case, IAF data transformation can be applied to huge data volumes in relational databases using the SQL code generated by the IFC-Filter.
The software is available for download\textsuperscript{1} in source and binary format for researchers and practitioners for experimenting and practicing inductive fuzzy classification. To make your initiation to it as smooth as possible we provide here a short tutorial to use the IFC-filter software:

- To start the application you can simply double click the file weka.bat. It is configured to be initialized with 2 GB RAM. You can change this setting by modifying the parameter \texttt{-Xmx2024m}. Java Version 6 is required.

- The simplest way to use the filter is to use the KnowledgeFlow button on the appearing GUI Chooser. There you have the possibility to load knowledge flow schemes in the left upper corner. Five of them are already prepared. You find them in the directory Scheme and in the sub directory DB. The scheme in the DB-folder contains a Database Loader which is configured for an Oracle XE database which can be downloaded free of charge. The other four schemes in the Scheme-folder contain data loader for ARFF file formats. The data for them are in the directory called Data.

- By right clicking the data loader a menu appears in which you can choose start loading. This will activate the data mining algorithm. The results can be seen in the text viewer component of the scheme. Right click the text viewer after the algorithm has completed and choose show result.

- The graphical component of the filter can be activated by right clicking the IFC-Filter component in the flow. This can be done with a right click on the component, by choosing configure and setting the IFCWindow parameter to true.

\footnotesize{http://diuf.unifr.ch/is/ifc (accessed 11.2010)}
The Knowledge Flow allows the interconnection of elements such as filters classifiers and clustering in a graphical interface. The knowledge flow window contains the following main components: The tool bar provides a choice of different components which can be dragged and dropped into the knowledge flow layout. The knowledge flow layout gives the opportunity to assemble WEKA components into a data flow in order to test, evaluate and perform data mining methods. The status and log panel supports the monitoring of the data mining process. The forth component contains buttons for clearing, saving or loading knowledge flow layouts. Figure 2 shows an example of a knowledge flow using a linear regression with IFC-Filter executed in parallel to the linear regression without IFC-Filter. The following paragraphs explain the corresponding data flow in detail.

Figure 2: Knowledge Flow with IFC-Filters

First, a data source is attached to the Knowledge Flow. For data contained in WEKA ARFF files, this is done with the ArffLoader which can be found in the DataSource section of the tool bar. DatabaseLoader is also a possible option if the data resides in a relational database. After that, training and test sets are generated. To achieve this, the source data is separated into two random data sets. For this purpose, the Randomize filter and the RemovePercentage filter can be applied, which can be found in the Filter toolbar. The Randomize filter blends the instances of a data set randomly. The RemovePercentage filter removes a percentage of the data set. By clicking with the right mouse button on the RemovePercentage icon, a configuration menu appears. Two parameters can be modified: the parameter percentage determines the percentage of the data set which will be removed; the parameter InvertSelection defines whether the data removal begins at the top or the bottom of the data. It is proposed to set invertSelection to true and percentage to 33% for training data, and to set invertSelection to false and percentage to 66% for test data. The resulting two randomly split data sets can be used for evaluating the performance of predictive models. The class label of the data set is defined with a ClassAssigner. In order to apply the IFC_Filter node, it is put between ClassAssigner and TestSetMaker or TrainingSetMaker. The nodes TestSetMaker and TrainingSetMaker define which part of the data split is used for training and which one for testing.

The configuration of the IFC_Filter can be performed by clicking with the right mouse button on the symbol of the IFC_Filter in the Knowledge Flow layout. In the appearing menu, configure can be chosen. This will open the configuration frame for the filter. The activation of the graphical illustration of the
results of the IFC_Filter can be activated by setting IFCWindow to true. The field classValue allows choosing the target variable if the class of the data set is categorical. Accepted values are “first”, “last”, or a numerical value which represents the position of the target variable. If the class is binary, this selection is not supported. The parameter percentageOfDataSet defines the percentage of the data dataset contained in a quantile at which the iteration of the algorithm is stopped. This function gives the possibility to compensate for over fitting. The parameter targetType gives a choice between categorical or binary target variables as output of the algorithm. In order to apply algorithms that require categorical class labels, the target type is set to categorical. For algorithms with numerical or binary target variables such as linear regression, the target type is set to binary.

Finally, the training and test datasets serve as input into a classifier node, which calculates a predictive model and computes predictions for the test set. In one case, the data is inductively fuzzified using an IFC filter, and in the second case, regression is applied directly to the original data. ClassifierPerformanceEvaluator compares the predictions and the actual values in the test set in order to assess the predictive performance for both variants. Finally, the results are output in a TextViewer.

**Visualization of membership functions**

As proposed in the section on application of inductive fuzzy classification to analytics, the IFC-Filter can visualize membership functions that indicate target membership likelihoods. Figure 8 shows two screenshots of the membership function plots for the variables “Duration” and “Checking status” from the German Credit dataset. The IFC-Filter has the possibility to activate a frame containing a graphical illustration of the resulting membership functions. Each analytical variable is represented by a tab in this frame. The presentation of numerical analytical variables differs slightly from that of categorical analytical variables, because it presents a continuous membership function. An additional tab, the SQL Panel contains the membership functions of all analytic variables in SQL syntax.

As shown in Figure 3, the illustration of numerical analytical variables consists of four fields. The first field is a table containing the normalized likelihood ratios (NLR) with the corresponding quantiles and average quantile values (AQV). The second field shows a histogram containing the NLR and their corresponding AQV. The third field shows the membership function of the analytical variable. The fourth field shows the membership function in SQL syntax for this particular analytical variable which can be used directly in a relational database for fuzzy classification of variables.

The illustration of categorical analytical variables can be reduced to three fields. The first field is a table containing the NLR with the corresponding quantile and average value of the quantile. The second field shows a histogram containing the NLRs corresponding to the categorical values. The third field shows the membership function in SQL syntax.

The SQL-Panel displays a concatenation of the membership functions for all analytical variables that have been input for membership function induction by the IFC-Filter. This database script can be applied in a database in order to transform large database tables into inductive degrees of membership in a target class. This can improve the predictive quality of multivariate models.
Figure 3: Visualization of membership functions with the IFC-Filter