CSE 398/498   BIG DATA ANALYTICS

Fall 2021 • 2:05 pm – 3:20 pm MW • BC 115 • Prof. Daniel Lopresti

Weka KnowledgeFlow

https://liacs.leidenuniv.nl/~kokjn/DM/knowledge.htm

The KnowledgeFlow presents a "data-flow" inspired interface to Weka. The user can select Weka components from a tool bar, place them on a layout canvas and connect them together in order to form a "knowledge flow" for processing and analyzing data. At present, all of Weka's classifiers and filters are available in the KnowledgeFlow along with some extra tools.

The KnowledgeFlow can handle data either incrementally or in batches (the Explorer handles batch data only). Of course learning from data incrementally requires a classifier that can be updated on an instance by instance basis. Currently in Weka there are five classifiers that can handle data incrementally: NaiveBayesUpdateable, IB1, IBk, LWR (locally weighted regression).

Features of the KnowledgeFlow:

- intuitive data flow style layout
- process data in batches or incrementally
- process multiple batches or streams in parallel! (each separate flow executes in its own thread)
- chain filters together
- view models produced by classifiers for each fold in a cross validation
- visualize performance of incremental classifiers during processing (scrolling plots of classification accuracy, RMS error, predictions etc)

Components available in the KnowledgeFlow:
Evaluation:

- TrainingSetMaker - make a data set into a training set
- TestSetMaker - make a data set into a test set
- CrossValidationFoldMaker - split any data set, training set or test set into folds
- TrainTestSplitMaker - split any data set, training set or test set into a training set and a test set
- ClassAssigner - assign a column to be the class for any data set, training set or test set

- ClassValuePicker - choose a class value to be considered as the "positve" class. This is useful when generating data for ROC style curves (see below).
- ClassifierPerformanceEvaluator - evaluate the performance of batch trained/tested classifiers
- IncrementalClassifierEvaluator - evaluate the performance of incrementally trained classifiers
- PredictionAppender - append classifier predictions to a test set. For discrete class problems, can either append predicted class labels or probability distributions.

Visualization:

- DataVisualizer - component that can pop up a panel for visualizing data in a single large 2D scatter plot
- ScatterPlotMatrix - component that can pop up a panel containing a matrix of small scatter plots (clicking on a small plot pops up a large scatter plot)
- AttributeSummarizer - component that can pop up a panel containing a matrix of histogram plots - one for each of the attributes in the input data
- ModelPerformanceChart - component that can pop up a panel for visualizing threshold (i.e. ROC style) curves.
- TextViewer - component for showing textual data. Can show data sets, classification performance statistics etc.
- GraphViewer - component that can pop up a panel for visualizing tree based models
- StripChart - component that can pop up a panel that displays a scrolling plot of data (used for viewing the online performance of incremental classifiers)

Filters: All of Weka's filters are available
Classifiers: All of Weka's classifiers are available
DataSources: All of Weka's loaders are available

Exercise

Goal: Setting up a flow to load an arff file (batch mode) and perform a cross validation using J48 (Weka's C4.5 implementation).

The Weka GUI Chooser window is used to launch Weka's graphical environments.

1. Select the button labeled "KnowledgeFlow" to start the KnowledgeFlow. Alternatively, you can launch the KnowledgeFlow from a terminal window by typing "java weka.gui.beans.KnowledgeFlow".
2. First start the KnowlegeFlow.
3. Next click on the DataSources tab and choose "ArffLoader" from the toolbar (the mouse pointer will change to a "cross hairs").
4. Next place the ArffLoader component on the layout area by clicking somewhere on the layout (A copy of the ArffLoader icon will appear on the layout area).
5. Next specify an arff file to load by first right clicking the mouse over the ArffLoader icon on the layout. A pop-up menu will appear. Select "Configure" under "Edit" in the list from this menu and browse to the location of your arff file.
6. Next click the "Evaluation" tab at the left of the window and choose the "ClassAssigner" (allows you to choose which column to be the class) component from the toolbar. Place this on the layout.
7. Now connect the ArffLoader to the ClassAssigner: first right click over the ArffLoader and select the "dataSet" under "Connections" in the menu. A "rubber band" line will appear. Move the mouse over the ClassAssigner component and left click - a red line labeled "dataSet" will connect the two components.
8. Next right click over the ClassAssigner and choose "Configure" from the menu. This will pop up a window from which you can specify which column is the class in your data (last is the default).
9. Next grab a "CrossValidationFoldMaker" component from the Evaluation toolbar and place it on the layout. Connect the ClassAssigner to the CrossValidationFoldMaker by right clicking over "ClassAssigner" and selecting "dataSet" from under "Connections" in the menu.
10. Next click on the "Classifiers" tab at the left of the window and scroll along the toolbar until you reach the "J48" component in the "trees" section. Place a J48 component on the layout.
11. Connect the CrossValidationFoldMaker to J48 TWICE by first choosing "trainingSet" and then "testSet" from the pop-up menu for the CrossValidationFoldMaker.

12. Next go back to the "Evaluation" tab and place a "ClassifierPerformanceEvaluator" component on the layout. Connect J48 to this component by selecting the "batchClassifier" entry from the pop-up menu for J48.
13. Next go to the "Visualization" toolbar and place a "TextViewer" component on the layout. Connect the ClassifierPerformanceEvaluator to the TextViewer by selecting the "text" entry from the pop-up menu for ClassifierPerformanceEvaluator.
14. Now start the flow executing by selecting the “Start” button (which looks like a blue video player triangle). Depending on how big the data set is and how long cross validation takes you will see some animation from some of the icons in the layout (J48's tree will "grow" in the icon and the ticks will animate on the ClassifierPerformanceEvaluator). You will also see some progress information in the "Status" bar and "Log" at the bottom of the window.
15. When finished you can view the results by choosing show results from the pop-up menu for the TextViewer component.
16. Other cool things to add to this flow: connect a TextViewer and/or a GraphViewer to J48 in order to view the textual or graphical representations of the trees produced for each fold of the cross validation (this is something that is not possible in the Explorer). Try the visualizations of the incremental classifiers.

From:
https://liacs.leidenuniv.nl/~kokjn/DM/knowledge.htm