Feature Selection - User Guide

Johannes Schöllhorn, Julian Schwab, Dennis Schwartz

July 6, 2011

Contents

Input Format

At the moment the csv format is used for data input. The csv file has some format conventions listed here :

- The class of a dataset has to be in the second column
- The decimal mark has to be a point ('.')
- There should be no thousands separator

The delimiter of columns in the csv file is comma ',' per default, but the program will ask for the delimiter, if comma does not lead to the correct format.
The symbols or strings for the different classes do not matter, however it should be limited to two different classes in one file (because of binary classification).
An example, how a correct input should look like (here the classes are named 1 and 0):

name	class	feature 1	feature 2
col 1	1	1.00	23.345
col 2	0	-45.34	1.34

in a .csv-file this would look like this :

name, class, feature 1, feature 2
col1, 1, 1.00, 23,345
col2, 0, -45.34, 1.34

Programm Parameters

The feature selection has varoius parameters to choose.

Wrapper

There are two different wrapper types implemented in this software. A forward selection wrapper and EvA2.

Forward Selection

The forward selection wrapper starts with an empty set of features and will add one feature per iteration, choosing whichever feature improves the performance the most. When performance improvement stagnates over a certain time period, the wrapper stopps.

EvA2

EvA2 [1] is a workbench of evolutionary algorithms released by the University of Tuebingen. The algorithm forms populations of feature subsets (individuals). As in a biological system the individuals can mutate or couple. The fitness of an individual is defined by the performance of its feature subset.

Performance

At this state of development, the program has implemented two different ways to compute the performance which is defined by the success of binary classification. Both ways use a confusion matrix as input which is calculated by the cross validation. This matrix is computed by evaluating the predictions of the classifier as follows :

TP A true positive will be counted if both the actual class of a sample (as given in the input file) and its predicted class are true.

FN A false negative will be counted if the actual class of a sample is true but its predicted class is false.

FP A false positive will be counted if the actual class of a sample is false but its predicted class is true .

TN A true negative will be counted if both the actual class of a sample and its predicted class are false.

Matthews Correlation Coefficient

The Matthews correlation coefficient is computed using the following formula [2] :

[1] `http://www.ra.cs.uni-tuebingen.de/software/JavaEvA/introduction.html`
[2] `http://en.wikipedia.org/wiki/Matthews_correlation_coefficient`

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

The performance is measured between -1 and 1. -1 is the worst performance, with no class predicted correctly and 1 is the best, where all predictions of the classifier are correct.

Accuracy

Accuracy computes the percentage of correct predictions by the classifier.
Accuracy is calculated using the following formula[3] :

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

If the accuracy is 0, 0 % of the predictions are correct, if it is 1 every prediction is correct.

Classifier

At this state the only implemented classifier is the CSVC-Classifier by libSVM[4] using a linear kernel.

Cross validation

Cross validation is a statistical technique to evaluate the quality of the classifiers predictions. In k-fold cross validation the data will be split in k subsets(folds) of which one will be used to perform the prediction on, while the remaining subsets will be used to validate said prediction. The number of subsets used may have an impact on the results as well as on computational time. It can be set to anything from 2 (2-fold), up to the number of samples in the given data (leave-one-out).
For example: If you want to use a 2-fold cross validation, set $k = 2$. If you want to choose leave-one-out, set k to the number of samples in your input.

Console Usage

It is possible to run the program via console, but its usage will set limitations to output capabilities. At this stage there is no save function for the intermediate and final results, as

[3] `http://en.wikipedia.org/wiki/Accuracy_and_precision`
[4] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

well as the plot depicting the improving performance of a run.

To use the program in a shell or console run ConsoleClient.java.
There are multiple ways to start a run using the console client:

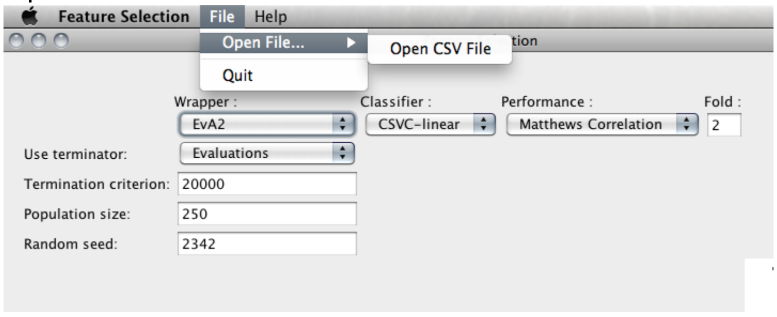
1. Entering consoleclient -i will start an interactive console client showing a dialogue, asking for parameters. You will have to enter parameters step by step.
2. Entering consoleclient [file] will run the program using default parameters.
   Parameters per default are set to:
   - wrapper : Forward selection
   - performance : MCC
   - cross validation : 2-fold cross validation
   - classifier : CSVC
3. If you enter consoleclient [file] and you want to choose different parameters, these are the options to choose from:
   - wrapper: enter FWD (Forward selection) or EVA (Eva2)
   - classifier: enter CSVC
   - performance: enter MCC for Matthews Correlation Coefficient or ACC for accuracy as method of performance computation

   as well as further options like the input file format (at the moment only .csv is supported), the random seed and fold for cross validation and some additional wrapper options for EvA2. Not specified options are set to default.

   -f x to set the fold, with x being the fold value
   –CrossValSeed x to set the random seed.
   -EvA2P x to set the desired the population size for EvA.
   -EvA2T specifies the termination criteria for EvA.
   -EvA2Seed specifies the random seed for EvA.

GUI Usage

1. First load the file to process by opening the file-open dialog in File -> Open File -> Open CSV File.



Wrapper :	EvA2
Classifier :	CSVC-linear
Performance :	Matthews Correlation
Fold :	2
Use terminator:	Evaluations
Termination criterion:	20000
Population size:	250
Random seed:	2342

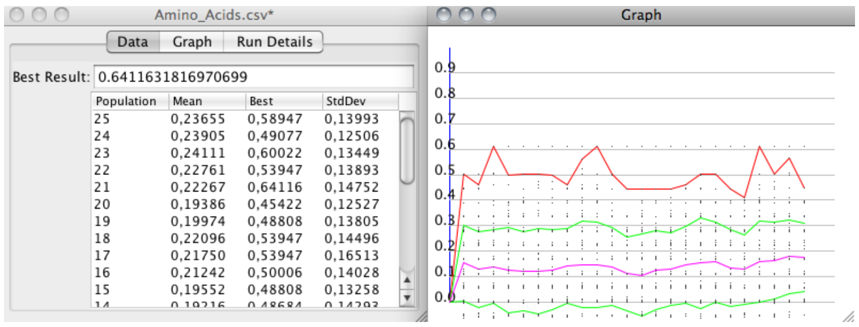
2. Specify the wrapper, classifier, performance and fold to be used in this run.

3. If necessary, specify the parameters of EvA.

4. Start the selection by hitting the 'Start' button.

While the selection process is running you will not be able to start a new one, however the running process can be forced to quit by clicking the stop button.

Upon the start of the process two new windows appear :



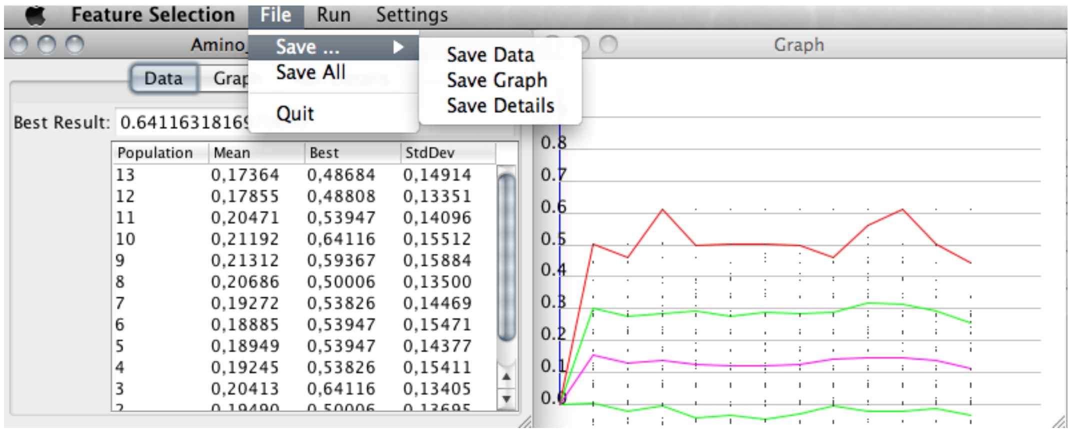
Best Result: 0.6411631816970699

Population	Mean	Best	StdDev
25	0,23655	0,58947	0,13993
24	0,23905	0,49077	0,12506
23	0,24111	0,60022	0,13449
22	0,22761	0,53947	0,13893
21	0,22267	0,64116	0,14752
20	0,19386	0,45422	0,12527
19	0,19974	0,48808	0,13805
18	0,22096	0,53947	0,14496
17	0,21750	0,53947	0,16513
16	0,21242	0,50006	0,14028
15	0,19552	0,48808	0,13258
14	0,19216	0,48684	0,14293

The left one, showing three tabs: results in a table, the graph, and a description which holds the file name, the chosen settings an the final results.

The right window shows a graph depicting the results in a separate window.

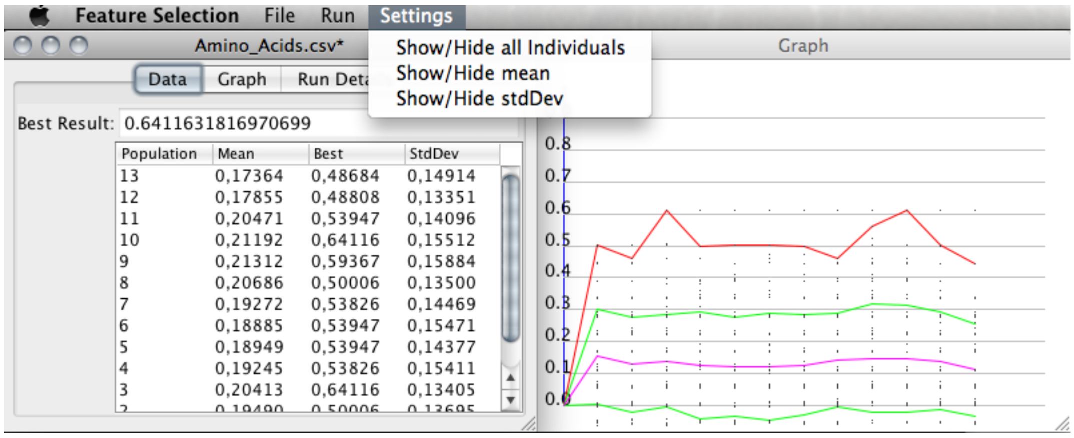
The right result window holds a menu to save the plot as .svg file, the table as .csv file and the description as txt file.



Best Result: 0.64116318169

Population	Mean	Best	StdDev
13	0,17364	0,48684	0,14914
12	0,17855	0,48808	0,13351
11	0,20471	0,53947	0,14096
10	0,21192	0,64116	0,15512
9	0,21312	0,59367	0,15884
8	0,20686	0,50006	0,13500
7	0,19272	0,53826	0,14469
6	0,18885	0,53947	0,15471
5	0,18949	0,53947	0,14377
4	0,19245	0,53826	0,15411
3	0,20413	0,64116	0,13405
2	0,19490	0,50006	0,13695

Additionally you can set the visibility of the single curves in the plot.



Best Result: 0.6411631816970699

Population	Mean	Best	StdDev
13	0,17364	0,48684	0,14914
12	0,17855	0,48808	0,13351
11	0,20471	0,53947	0,14096
10	0,21192	0,64116	0,15512
9	0,21312	0,59367	0,15884
8	0,20686	0,50006	0,13500
7	0,19272	0,53826	0,14469
6	0,18885	0,53947	0,15471
5	0,18949	0,53947	0,14377
4	0,19245	0,53826	0,15411
3	0,20413	0,64116	0,13405
2	0,19490	0,50006	0,13695

How to enlarge the feature selection API

Create a new wrapper

To create a new wrapper you will have to create a class that extends the abstract Wrapper-Base class. The parameters your new wrapper may need can be stored in a config class that implements WrapperConfigBase.
After the implementation of a new wrapper class you will have to add the new wrapper to the enum WrapperTypes. This will make it possible to choose the new wrapper in the GUI, however if your wrapper requires parameters, you will have to update the GUI.

Create a new performance

If you would like to implement a new method to calculate the performance of the binary classification you will have to create a new class that implements the PerformanceI interface and implement its calculatePerformance method. If you need to store parameters for your performance class you can create a new config class that extends PerformanceConfigBase. Add your new performance to the enum PerformanceTypes to make it accesible via GUI.

Create a new cross validation

If you would like to implement another cross validation, create a new class that extends the CrossValidationBase class and implement all necessary methods. If you need to store parameters for your cross validation class you can create a new config class that extends CrossValidationConfigBase.

Create a new classifier

If you would like to implement another classifier just create a new class that extends the ClassifierBase class and implement all the methods you need to. If you need to store parameters for your classifier class you can create a new config class that extends ClassifierConfigBase. Add your new classifier to the enum ClassifierTypes to make it accesible via GUI.