How to use `GramMatrixOptimizer.jar`

Kilho Shin

January 8, 2018

Overview. Given a `.kernel` file, this program finds an optimal setting of the following parameters through grid search and cross validation with the C-SVM classifier.

α**:** The value of node similarity when the labels of two maching nodes are identical.

β**:** The value of node similarity when the labels of two maching nodes are not identical.

C**:** The regulation parameter C of the C-SVM classifier.

The program takes advantage of a simple single-layer grid search. The entire parameter space specified by a configuration file is decomposed into equally spaced grids, and for each grid point, the program runs cross validation of a fold number also specified by the configuration file. Each execution of cross validation generates a confusion matrix, and the program chooses the combination of parameters that exhibits the best accuracy score, defined by $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$.

This program leverages the Spark framework for parallel computation: More than one combinations of parameters will be tested simultaneously.

Previous program. A program that computes gram matrices of kernels and outputs the computed matrices in `.kernel` files, for example, `TK.jar`.

Next program. `GramMatrixPredictor.jar`, which receives an output of this program and predicts classes given unknown instances.

Usage.

```
java -jar GramMatrixOptimizer.jar config.txt
```

Configuration files. A configuration looks as follows.

```
KERNEL: colon.kernel
LOG: log.csv
RESULT: problem.txt
```

```
alpha_min:  0.0
alpha_max:  1.0
alpha_div:  5
beta_min:  0.0
beta_max:  1.0
beta_div:  5
logc_min:  -3.0
logc_max:  3.0
logc_div:  10
cv:  5
norm
```

`KERNEL:` This specifies a `.kernel` to be optimized. The `.kernel` file should be an output of a program that computes gram matrices of kernels, for example, the `TK.jar` program. If left out, `./in.kernel` will be used.

`LOG:` If specified, all of the tested combinations of parameter will be written in the specified log file with TP, TN, FP and FN scores. The file format is CSV. If left out, `./log.csv` will be generated.

`RESULT:` This specifies a problem file to include the result of running this program. The problem file is an input into the `GramMatrixPredictor.jar` program, and specifies the kernel values at the chosen optimal combination of parameters. If left out, `./out.txt` will be generated.

`alpha_min, _max, _div:` These specifies the grids for the parameter α. The default values are:

```
alpha_min = 0.0, alpha_max = 1.0, alpha_div = 4.
```

This means that the values of 0.0, 0.25, 0.5, 0.75 and 1.0 will be tested for the parameter α.

`beta_min, _max, _div:` These specifies the grids for the parameter β. The combinations to test must meet $\beta \leq \alpha$.

`logc_min, _max, _div:` These specifies the grids for the logarithm of the parameter C. The default values are:

```
logc_min = -3.0, logc_max = 3.0, logc_div = 4.
```

Therefore, the values of $10^{-3.0}, 10^{-1.5}, 10^{-0.0}, 10^{1.5}$ and $10^{3.0}$ will be tested for the parameter C.