10601 Machine Learning Project Milestone 3: Tuning Classifiers

Due: Nov 6th, 16:30 EST (Before the lecture), via AutoLab Late submission due: Nov. 6th, 23:59 EST with 50% discount of credits

TAs-in-charge: Avinava Dubey, Ying Sheng, Shangqing Zhang

Policy on Collaboration among Students

These policies are the same as were used in Dr. Rosenfeld's previous version of 2013. The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes are shared, or are taken at that time, and provided learning is facilitated, not circumvented. The actual solution must be done by each student alone, and the student should be ready to reproduce their solution upon request. The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved, on the rest page of their assignment. Specially,each assignment solution must start by answering the following questions in the report:

Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered `yes', give full details: (e.g. \Jane explained to me what is asked in Question 3.4")
Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered `yes', give full details: (e.g. \I pointed Joe to section 2.3 to help him with Question 2".
Collaboration without full disclosure will be handled severely, in compliance with CMU's Policy on Cheating and Plagiarism. As a related point, some of the homework assignments used in this class may have been used in prior versions of this class, or in classes at other institutions. Avoiding the use of heavily tested assignments will detract from the main purpose of these assignments, which is to reinforce the material and stimulate thinking.

Because some of these assignments may have been used before, solutions to them may be (or may have been) available online, or from other people. It is explicitly forbidden to use any such sources, or to consult people who have solved these problems before. You must solve the homework assignments completely on your own. I will mostly rely on your wisdom and honor to follow this rule, but if a violation is detected it will be dealt with harshly.

Collaboration with other students who are currently taking the class is allowed, but only under the conditions stated below.

Global Instruction for millstone 3 (Important):

In the mile3 round, the final goal is we want to let you find tuned learner that gives the best overall performance on the datasets. You will be asked to upload 12 models with One classier on autolab. But this time that 12 model should be consistent (only comes from one classifier). It might be that some of the options were just picked by hand, and held constant for all the datasets; or it could be that some options are set dynamically, eg by internal cross-validation. You should explore different values of some options, and try using internal cross-validation (Hint: some meta classifiers support this, one is called CrossValidatedClassifier, and one is called GridSearchClassifier).

Section 1. Tuning the XXX classifier learner.

This section should be the major part of your report. For each of your classifier, you should go through section 1.1 to 1.3.

Section 1.1.

Tell us the background of the XXX classifier you use. A sentence or two about what the classifier does and where it is described in the literature. You should include the "normal" name from the literature as well as the Weka class name.

Section 1.2. Parameters we varied.

A sentence or two about each parameter that you experimented with: what it's called, what it does with respect to the algorithm, and how it affects things we might have discussed in class (over-fitting? classifier size? learning time?)

Also, include a sentence or two about how you tried to vary the parameter, and how it affected error rates and robustness, and your final classifier-learner. (Eg, "JRip seems to be rather insensitive to this parameter over the test datasets, with the error rate never changing by more than 2-3% on any dataset. As a consequence max and average robustness was nearly unchanged when we varied it by using a CVParameterSelection wrapper around JRip." or "REP seems to be quite sensitive to the number or internal cross-validation folds used, with the most robust performance occurring for fairly large values. After experiments, we fixed this parameter at 50, which seemed to strike a good balance between compute time and robustness.
Also tell us the training time in weka with that tunning classifier.

Section 1.3. Final settings.

A sentence or two about how your final classifier learner was constructed and configured.

Section 2. Tuning the YYY classifier learner.

....

Section 3. Summary.

Here is an Example

"Among the classifiers we used, the best robustness was obtained by random forests, which was naturally quite robust without parameter tuning. However, by using nested Cross Validation to set the two most critical values of REP, and changing one default parameter setting, we were able to improve its average robustness from 1.04 to 0.93, and improve it's maximum robustness from 1.2 to 1.15."

2 Deliverable

Note (Important):
In millstone 3 your 12 model should be **consistent.** You should first do 3x12 experiments on 12 benchmark with 3 classifiers. Then tunning them to accomplish Session One. For code submission, you just need to pick One with best robustness you think, and provide 12 models with that classifier.
It follows the similar pattern as millstone 2. Each member of the team should submit a ms3.tgz to autolab. Here's the list of les need to be included in ms3.tgz:

1. 12 trained classier models after tuning, one for each benchmark, for anneal_train.arff and anneal_test.arff, you should have a corresponding model named anneal.model
2. 12 $error_c/error_{NB}$ values corresponding to 12 models, one for each benchmark, the average of $error_c/error_{NB}$ and the max of $error_c/error_{NB}$. You should submit a result.txt file with 14 lines. The first 12 lines cover the 12 models' $error_c/error_{NB}$ value. 13th line is the average and 14th line is the maximum.
3. Your java code to do this work.
4. A report.pdf file which should cover the contains on section 1.

3 Submission

You must submit your homework through Autolab via the “projectmilestone3" link.

1. Projectmilestone3: This is where you should submit your validated final submission.
2. You have a total of 5 possible submissions. Your performance will be evaluated, and feedback will be provided immediately.