RescoreTenFcv

10-Fold Cross Validation

This is a log of what I did to run a 10-fold cross-validation test of the perceptron vs the GA when testing bug 2910, http://bugzilla.spamassassin.org /show_bug.cgi?id=2910 . Note that many things have changed since this log was created, and an attempt to re-run these commands verbatim will probably fail. (-- JustinMason 21/01/04).

First, I checked out the source:

```
svn co https://svn.apache.org/repos/asf/incubator/spamassassin/trunk
cd trunk
perl Makefile.PL
make
cd masses
```

(Update Jul 2007: nowadays, that's at https://svn.apache.org/repos/asf/spamassassin/trunk .)

get pgapack and install as "masses/pgapack". I just scp'd in an already-built tree I had here.

use the set-0 logs from the 2.60 GA run – taken from the rsync repository:

```
wc -l /home/corpus-rsync/corpus/Obsolete/submit-2.60-GA-run1/ham-set0.log \
      /home/corpus-rsync/corpus/Obsolete/submit-2.60-GA-run1/spam-set0.log
 210442 /home/corpus-rsync/corpus/Obsolete/submit-2.60-GA-run1/ham-set0.log
 354479 home/corpus-rsync/corpus/Obsolete/submit-2.60-GA-run1/spam-set0.log
```

we want about 2k in each bucket, otherwise it'll take weeks to complete. use split-logs-into-buckets (see SplitLogsIntoBuckets) to juggle the log files in blocks of 10% to get the ratio and size to around 2k:2k.

ham buckets first:

```
./tenpass/split-log-into-buckets 10 \
    < /home/corpus-rsync/corpus/Obsolete/submit-2.60-GA-run1/ham-set0.log
mv split-1.log new
./tenpass/split-log-into-buckets 10 < new
wc -l split-1.log
   2104 split-1.log
```

much better!

```
mv split-*.log ../../logs/nonspam-jm/

./tenpass/split-log-into-buckets 10 \
    < /home/corpus-rsync/corpus/Obsolete/submit-2.60-GA-run1/spam-set0.log
mv split-1.log new
wc -l new
  35437 new
```

given this, we want 6 of the 10 logfiles to make 21264 lines, which would result in a roughly even ham:spam ratio for testing. let's do that.

```
cat split-{1,2,3,4,5,6}.log > new
./tenpass/split-log-into-buckets 10 < new
wc -l split-1.log
   2126 split-1.log
```

perfect!

```
mv split-*.log ../../logs/spam-jm/
```

and doublecheck the log sizes:

```
wc -l ../../logs/*/*.log
   2104 ../../logs/nonspam-jm/split-1.log
   2103 ../../logs/nonspam-jm/split-10.log
   2106 ../../logs/nonspam-jm/split-2.log
   2103 ../../logs/nonspam-jm/split-3.log
   2102 ../../logs/nonspam-jm/split-4.log
   2105 ../../logs/nonspam-jm/split-5.log
   2102 ../../logs/nonspam-jm/split-6.log
   2103 ../../logs/nonspam-jm/split-7.log
   2103 ../../logs/nonspam-jm/split-8.log
   2104 ../../logs/nonspam-jm/split-9.log
   2126 ../../logs/spam-jm/split-1.log
   2127 ../../logs/spam-jm/split-10.log
   2126 ../../logs/spam-jm/split-2.log
   2126 ../../logs/spam-jm/split-3.log
   2128 ../../logs/spam-jm/split-4.log
   2126 ../../logs/spam-jm/split-5.log
   2126 ../../logs/spam-jm/split-6.log
   2126 ../../logs/spam-jm/split-7.log
   2126 ../../logs/spam-jm/split-8.log
   2125 ../../logs/spam-jm/split-9.log
  42297 total
```

looks fine.

Next step was to ensure the scores in ../rules are usable for the GA. I did this by grepping out all the rules that had scores of 0 in the tested score-set (score-set 0), and creating a new file called ../rules/99_ga_workaround_scores.cf containing lines like so:

```
   score NAME_OF_RULE_1   0.0001
   score NAME_OF_RULE_2   0.0001
   score NAME_OF_RULE_3   0.0001
...
```

This is necessary so that the mass-check log files, which were generated using the same ruleset but possibly with some rules enabled where they are now disabled, will still be useful; this way, all rules in the ruleset are again enabled, and will be considered by the GA as candidates for evolution.

now run the 10pass master script.

```
nohup sh -x ./tenpass/10pass-run &
```

Results will appear in "tenpass_results" – over the course of 4 days. 😉

These will be:

- scores.{1 .. 10}: scores and GA accuracy ratings output by GA
- {ham,spam}.log.{1 .. 10}: validation log files for that set of scores

To perform the validation step, run

```
./tenpass/10pass-compute-tcr
```

This will compute an accuracy rating, using those scores and those validation log files, for the 10 folds. Output looks like:

```
# TCR: 14.173333  SpamRecall: 96.002%  SpamPrec: 99.367%  FP: 0.31%  FN: 2.01%
# TCR: 13.986842  SpamRecall: 96.143%  SpamPrec: 99.320%  FP: 0.33%  FN: 1.94%
# TCR: 15.865672  SpamRecall: 95.579%  SpamPrec: 99.608%  FP: 0.19%  FN: 2.22%
# TCR: 14.173333  SpamRecall: 95.532%  SpamPrec: 99.461%  FP: 0.26%  FN: 2.25%
# TCR: 15.748148  SpamRecall: 95.532%  SpamPrec: 99.608%  FP: 0.19%  FN: 2.25%
# TCR: 12.807229  SpamRecall: 95.014%  SpamPrec: 99.409%  FP: 0.28%  FN: 2.51%
# TCR: 14.561644  SpamRecall: 94.779%  SpamPrec: 99.654%  FP: 0.17%  FN: 2.63%
# TCR: 12.432749  SpamRecall: 94.309%  SpamPrec: 99.504%  FP: 0.24%  FN: 2.86%
# TCR: 14.358108  SpamRecall: 95.859%  SpamPrec: 99.414%  FP: 0.28%  FN: 2.08%
# TCR: 18.318966  SpamRecall: 95.953%  SpamPrec: 99.707%  FP: 0.14%  FN: 2.03%
```

These figures can be compared with other 10FCV runs; they're a good measurement of training accuracy. In other words, they're what you came for. 😉

10-Fold Testing With The Perceptron Instead of GA

If all goes well, the Perceptron will take over from the GA as the main way we generate scores; in that case, this section will be obsolete.

copied ./tenpass/10pass-run to ./10pass-run-perceptron .

Changed these lines:

```
make clean >> make.output
make >> make.output 2>&1
./evolve
pwd; date
```

to

```
make clean >> make.output
make -C perceptron_c clean >> make.output
make tmp/tests.h >> make.output 2>&1
rm -rf perceptron_c/tmp; cp -r tmp perceptron_c/tmp
make -C perceptron_c >> make.output
( cd perceptron_c ; ./perceptron -p 0.75 -e 100 )
pwd; date
```

Change

```
cp craig-evolve.scores tenpass_results/scores.$id
```

to

```
perl -pe 's/^(score\s+\S+\s+)0\s+/$1/gs;' \
    < perceptron_c/perceptron.scores \
    > tenpass_results/scores.$id
```

(required to work around an extra digit output by the perceptron app) and run ./10pass-run-perceptron . This one completes a lot more quickly 😉