HitFrequencies

Hit Frequencies

"hit-frequencies" is a script in the 'masses' directory of the SpamAssassin source distribution, used to measure rule accuracy and hit-rates, based on the output log files from MassCheck.

Once you've run MassCheck, you have a "ham.log" and a "spam.log" file. To turn those into a useful summary, you run "hit-frequencies" to generate a "freqs report". Here's how – run:

```
    ./hit-frequencies -x -p -a > freqs
```

(Add the -s switch to use alternate scoresets; for example `-s 3` will measure rule scores with scoreset 3.)

That will take "ham.log" and "spam.log" and generate a "freqs" file from the data. This gives you the frequencies that each rule hits and details of its accuracy in hitting spam vs. ham.

This script counts the occurences of rules in *.log files and calculates some details of their accuracy in hitting spam vs. ham. The rules are read from the *.cf files (the ones that begin with a digit) in the "../rules" folder.

The Format

HitFrequencies output looks like this:

```
OVERALL%   SPAM%     HAM%     S/O    RANK   SCORE  NAME
   6317     2614     3703    0.414   0.00    0.00  (all messages)
100.000  41.3804  58.6196    0.414   0.00    0.00  (all messages as %)
  2.153   5.2028   0.0000    1.000   1.00    4.30  RCVD_IN_OPM_HTTP
  1.219   0.0000   2.0794    0.000   1.00   -0.10  RCVD_IN_BSP_OTHER
  0.364   0.8799   0.0000    1.000   0.99    4.30  RCVD_IN_OPM_SOCKS
  0.332   0.0000   0.5671    0.000   0.99   -4.30  RCVD_IN_BSP_TRUSTED
  0.063   0.1530   0.0000    1.000   0.99    4.30  RCVD_IN_OPM_WINGATE
  1.061   2.5249   0.0270    0.989   0.96    0.64  RCVD_IN_NJABL_SPAM
  0.697   1.6067   0.0540    0.967   0.90    1.10  RCVD_IN_SORBS_SMTP
  1.520   3.4430   0.1620    0.955   0.87    1.10  RCVD_IN_SORBS_HTTP
```

The columns are:

OVERALL%	the percentage of mail overall that the rule hits
SPAM%	the percentage of spam mails hit by the rule
HAM%	the percentage of ham mails hit by the rule
S/O	"spam over overall ratio" – the probability that, when the rule fires, it hits on a spam message
RANK	An artificial ranking that indicates how "good" the rule is.
IG	Information gain of the rule, normalized to a value between 1 and 0. Intuitively this shows how much knowing the rule helps to guess the correct classification of a e-mail. In general, RANK works better.
SCORE	the score listed in the "../rules/50_scores.cf" file for that rule
NAME	the rule's name

The first two lines list the number of messages in the corpora, and the percentage makeup of the corpus as ham vs. spam (so in this example, the corpus is 41.38% spam vs 58.61% ham).

"freqs" is the best way to determine a rule's usefulness, since it immediately shows up any false-positive issues. The development team run a nightly mass-check and freqs report from the rules in CVS to test them, with several people scanning their corpora (see NightlyMassCheck); in addition, there are multiple mass-check/hit-frequencies run after every SVN check-in (see PreflightBuildBot).

The S/O Ratio

S/O needs more explanation, as it's a key figure. A rule with S/O 1.0 is very very accurate at hitting spam without hitting ham; a rule with S/O 0.0 hits only ham, no spam; but a rule with 0.5 hits both evenly (and is therefore pretty useless).

A *good* rule has a very extreme S/O (near as possible to 1.0 or 0.0) and a high percentage of hits in the correct category. In other words, RCVD_IN_OPM_HTTP is a very good rule in the example above, because it hits 5.2028% of all spam mails without hitting any ham at all (no false positives).

S/O stands for "spam / overall" for which the formula is "spam% / (ham% + spam%)", in other words, the proportion of the total hits that were spam messages. As such, it is equivalent to Bayesian probability, or Positive Predictive Value in bioinformatics or medicine.

Measuring Rule Overlap

There's one more tool to determine how much 2 rules overlap with each other – "overlap". This is occasionally useful if you suspect that two rules are redundant, checking the same data or hitting exactly the same messages as each other. Take a look at the comments at the top of the "masses/overlap" script for details on how to run this against one or more "mass-check" output log files.

Alternatively, "hit-frequencies" has the `-o` switch to measure overlap; warning, however, this can be quite a bit slower and RAM-hungry than running without it, as it then needs to track a lot more data internally.

Usage

usage:

hit-frequencies [-c rules dir] [-f] [-m RE] [-M RE] [-X RE] [-l LC] [-s SC] [-a] [-p] [-x] [-i] [spam log] [ham log]

-c p	use p as the rules directory, default: "../rules"
-f	falses. count only false-negative or false-positive matches
-m RE	print rules matching regular expression
-t RE	print rules with tflags matching regular expression
-M RE	only consider log entries matching regular expression
-X RE	don't consider log entries matching regular expression
-l LC	also print language specific rules for lang code LC (or 'all')
-L LC	only print language specific rules for lang code LC (or 'all')
-a	display all tests
-p	percentages. implies -x
-x	extended output, with S/O ratio and scores
-s SC	which scoreset to use
-i	use IG (information gain) for ranking

options -l and -L are mutually exclusive.

options -M and -X are *not* mutually exclusive.

if either the spam or and ham logs are unspecified, the defaults are "spam.log" and "ham.log" in the current working directory.