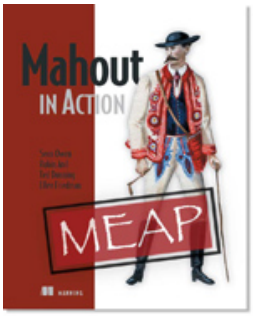
MANNING PUBLICATIONS



Mahout in Action
Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman

In this article, Mahout in Action author explain log-likelihood, a probability score measure that can be used with multiple target categories.

Computing Average Log-Likelihood

When we have a model that produces probability score outputs, the log of the score for the correct answer is a useful measure of how good the model is. When this value is averaged over a large number of held-out examples, you get the measure that is known as the average log-likelihood. This term is often abbreviated as log-likelihood, at the risk of some confusion.

Log-likelihood is a useful contrast with AUC (area under the receiver operating characteristic curve). Log-likelihood can be used with multiple target categories, while AUC is limited to binary outputs. AUC, on the other hand, can accept any kind of score, while log-likelihood requires scores that reflect probability estimates. AUC has an absolute scale from 0 to 1 that can be used to compare different models while the scale for log-likelihood is open ended.

The value of log-likelihood for a single training example is not quite useful because it varies so much from example to example, but the average over a number of examples is useful. `OnlineSummarizer` can average the log-likelihood estimates from a classifier and summarize the estimates using median, mean, and upper and lower quartiles like this:

```
OnlineSummarizer summarizeLogLikelihood = new OnlineSummarizer();
for (int i = 0; i < 1000; i++) {
  Example x = Example.readExample(); #1
  double ll = classifier.logLikelihood(x.target, x.features); #2
  summarizeLogLikelihood.add(ll);   #3
}

System.out.printf(
    "Average log-likelihood = %.2f (%.2f, %.2f) (25%%-ile,75%%-ile)\n",
    summarizeLogLikelihood.getMean(),
    summarizeLogLikelihood.getQuartile(1), #4
    summarizeLogLikelihood.getQuartile(2));
```

#1 Gets example
#2 Computes log-likelihood
#3 Adds to summary
#4 Computes and prints statistics

In this code, examples are read (#1) and a classifier computes the log likelihood (#2) which is then passed to an `OnlineSummarizer` (#3). After the input has been read, this summarize can be queried (#4) to get descriptive statistics for the distribution of log-likelihood.

Log-likelihood has a maximum value of zero and no bound on how negative it can go. For highly accurate classifiers, the value of average log-likelihood should be close to the average percent correct for the classifier multiplied by the number of target categories. For less accurate classifiers, especially those with many target categories, the average percent correct tends to go to zero, making it difficult to compare classifiers. The log-

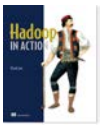
likelihood, on the other hand, can still distinguish better from worse classifiers even when the classifiers being compared are rarely or even never exactly correct, which makes the average percent correct zero.

For internal developer audiences, log-likelihood is probably a better measure for comparing models but, when presenting the results to a less technically sophisticated audience, percentage correct or even a confusion matrix is probably going to work better to convey your results. Metrics like AUC and log-likelihood give us a picture of which models perform better or worse overall.

Summary

We discussed log-likelihood, which is computed by taking the average of the log of the score for the target category.

Here are some other Manning titles you might be interested in:



Hadoop in Action
Chuck Lam

The Cloud at Your Service
Jothy Rosenberg and Arthur Mateos

Real-World Functional Programming
Tomas Petricek with Jon Skeet

Last updated: May 20, 2011