Performance Evaluation of Document Image Algorithms

Robert M. Haralick

Department of Electrical Engineering, University of Washington
Seattle, WA 98195
haralick@ee.washington.edu

1 Introduction

Performance evaluation for document image processing has a different emphasis than performance evaluation in other areas of image processing. Other areas of image processing can tolerate some error. Because it is so easily done nearly perfectly by humans, document image processing must also be done nearly perfectly. So the first aspect of performance evaluation for document image processing is to determine the domain in which the performance is nearly perfect. Outside this domain, the algorithm makes errors. Such instances of errors need to be examined and classified and categorized so that the weaknesses of the algorithm can be characterized.

A parametric aspect of performance evaluation for document image processing is the determination of the performance as a function of increasing level of image perturbation. This perturbation is associated with the various kinds of noise that make a real document image page differ from the ideal. Since this aspect of performance evalution is associated with an explicit noise model, it is possible to do this evaluation by the generation of synthetic images which are perturbed by simulated noise in accordance with the noise perturbation model.

The last category of performance evaluation for document image processing is the category of overall performance in a specific document image population. This requires an experiment to be done with a significant sized population of document images suitably randomly selected to be representative of some real application domain.

Performance can be measured as a function of predefined categories of documents, as a function of noise perturbation parameters, or as a function of internal algorithm tuning parameters. To obtain a single number, the various different dimensions of performance must be weighted in an overall manner consistent with the application of interest.

Because performance is an empirical measurement, and because the set of images used in making this empirical measurement is in effect only a small fraction from a large population, a sample used in making the empirical measurement one time to another time will differ and induce differences in performance. These differences are due to sampling variations. The sampling variations induce variations in the estimated system parameters during training and induce

Atul K. Chhabra and D. Dori (Eds.): GREC'99, LNCS 1941, pp. 315–323, 2000.

variations in the performance values during testing. Therefore, another aspect of performance evaluation must include the measuring of the variation of estimated system tuning parameters due to the sampling variations in the document image training sample as well as the variation in estimated system performance dues to sampling variations in the document image testing sample.

Finally, there is the issue of testing the algorithm to determine if it meets its performance specifications. Being sure that an algorithm meets its specifications can require assessing its performance on a much larger sample than intuition might lead one to believe. And depending on what is meant by "meeting specifications" being sure that an algorithm meets its specification can require that its performance in an assessment be much better than intuition might lead one to believe.

2 The Method of Performance Evaluation

Performance evaluation uses the method of controlled experiment to measure the extent to which the algorithm segments recognizes, and locates the entities it is designed to handle. It must measure the sensitivity of performance ot changes in tuning parameter values, training set sampling variations and testing set sampling variations. It must have a protocol that permits the hypothesis to be tested that the system meets its requirement specifications.

The controlled experiment requires an experimental protocol having a measurement plan, a sampling design, and a statistical data analysis plan. The measurement plan states what quantities will be measured, how they will be measured and the accuracy with which they will be measured. The sampling design defines the population of document images and describes how a suitable random independent set of document images will be sampled from the population. The statistical data analysis plan describes how the raw data gathered from the experiment will be analyzed. The analysis itself consists of one part that estimates the performance statistic and another part that estimates the statistical deviation of the true value of the performance statistic from the estimated value. The data obtained by the analysis must be complete enough so that the hypothesis that the system meets its stated requirements can be tested. And it must be supported by a theoretically developed statistical analysis which shows that an experiment carried out according to the measurement plan and sampling design and analyzed according to the statistical data analysis plan will produce a statistical test itself having the required accuracy. This analysis, if it is standard, can refer to statistical texts for the test used.

3 Meeting Performance Requirements

The requirement statement for a recognition task is a statement of the form: the probability that the algorithm fails to recognize an entity of interest (depending on the kind of algorithm, entities of interest might be zones, lines, words, tables, figures etc.) in the given population is less that f_o. The experiment that is done

to determine whether an algorithm meets its performance is called an acceptance test. In this section we give a short review of acceptance tests.

Any kind of acceptance test inevitably has two kinds of errors: the errors of omission and the errors of commission. A system may actually meet its requirements, but due to testing set sampling variations, there will be some probability that the hypothesis that the system meets the requirements will be rejected. This is an error of omission. As the sampling set size increases, this error will decrease.

On the other hand, a system may not actually meet its requirements, but due to testing set sampling variations, there will be some probability that the hypothesis that the hypothesis that the system meets its requirements will not be rejected. This is an error of commission. Again as the sampling set size increases, this error will decrease.

3.1 The Derivation

Consider the case for omission errors. The case for commission errors is obviously similar. Let N, the sampling size, be the total number of entities observed and let K be the number of times that the algorithm fails to properly recognize the entity out of the N entities observed.

The simplest intuitive way of making the comparison between f_o and K is to use K in the natural manner to estimate the true probability rate f. The maximum likelihood estimate $\hat{f}$ of f based on K is $\hat{f} = K/N$. If the estimate $\hat{f}$ of f is less than f_o, we judge that the algorithm passes the test. If the estimate $\hat{f}$ is greater than f_o, we judge that the algorithm fails the acceptance test. The issue with such a procedure is how sure are we if we apply such a procedure that the judgment we make about the algorithm's performance is a correct judgment. To answer this issue we must estimate the performance of our judgment. We start from the beginning.

To carry out the estimation, we suppose that, conditioned on the true error rate f, the algorithm's recognition failures are independent and identically distributed. Let X_n be a random variable taking the value 1 for an incorrect recognition and taking a value 0 otherwise, when the algorithm is judging the n^{th} entity.[1] In the maximum likelihood technique of estimation, we compute the estimate $\hat{f}$ as the value of f which maximizes

$$Prob\ (\sum_{n=1}^{N} X_n = K \mid f) = \binom{N}{K} f^K (1-f)^{N-K}$$

Taking the partial derivative with respect to f and setting the derivative to zero results in $\hat{f} = K/N$, the natural estimate of f.

Suppose that we adopt the policy of accepting the algorithm if $\hat{f} \leq f_o$. To understand the consequences of this policy, consider the probability that the

[1] We assume the recognition incorrectness of the entities are statistically independent.

policy results in a correct acceptance decision. The probability that $f \leq f_o$ given that $\hat{f} \leq f_o$ needs to be computed.

$$Prob\ (f \leq f_o \mid \hat{f} \leq f_o) = \int_{f=0}^{f_o} Prob\ (f \mid \hat{f} \leq f_o)\ df$$

$$= \frac{\int_{f=0}^{f_o} Prob\ (\hat{f} \leq f_o \mid f)\ Prob\ (f)\ df}{Prob\ (\hat{f} \leq f_o)}$$

To make the mathematics simple, let f_o be constrained so that there is some integer K_o such that $f_o = K_o/N$. Then

$$Prob\ \ (f \leq f_o \mid \hat{f} \leq f_o) = \frac{\int_{f=0}^{f_o} Prob\left(\sum_{n=1}^{N} X_n \leq K_o \mid f\right) Prob\ (f)\ df}{\int_{f=0}^{1} Prob\left(\sum_{n=1}^{N} X_n \leq K_o | f\right) Prob(f)\ df}$$

The probability that the true value of f is less than or equal to f_o given that the observed value $\hat{f}$ is less than f_o will depend, in general, on the testor's prior probability function $Prob\ (f)$. So, depending on the acceptance testor's prior probability function $Prob\ (f)$, there will besome smallest number F, $0 \leq F \leq 1$, such that $Prob\ (f) = 0$ for all $f > F$. Here, the support for the prior probability function is the interval $[0, F]$.

For example, an acceptance testor who has had successful experience with previous algorithms from the same company might have a prior probability function whose support is the interval $[0, 2f_o]$. An acceptance testor who has had no previous experience with the company might have a prior distribution whose support is the interval $[0, 10f_o]$. An acceptance testor who has had an unsuccessful experience with a previous algorithm from the same company might have a prior distribution for f whose support is the interval $[0,\ .5]$.

In each of the above cases, we assume that neither we nor the testor know anything more about the prior probability function than the interval of support $[0, F]$, where we assume that $F \geq f_o$, since if not, there would be no point to perform an acceptance test to establish something we already know. In this case, we take $Prob\ (f)$ to be that probability function defined on the interval $[0, F]$ having highest entropy. Such a $Prob\ (f)$ is the uniform density on the interval $[0, F]$. Hence, we take $Prob\ (f) = \frac{1}{F}$, $0 \leq f \leq F$. Therefore

$$Prob\ (f \leq f_o \mid \hat{f} \leq f_o) = \frac{\int_{f=0}^{f_o} \sum_{k=0}^{K_o} \binom{N}{k} f^k (1-f)^{N-k} df/F}{\int_{f=0}^{F} \sum_{k=0}^{K_o} \binom{N}{k} f^k (1-f)^{N-k} df/F}$$

$$= \frac{\sum_{k=0}^{K_\circ} \binom{N}{k} B(k+1, N+1-k) I_{f_\circ}(k+1, N+1-k)}{\sum_{k=0}^{K_\circ} \binom{N}{k} B(k+1, N+1-k) I_F(k+1, N+1-k)}$$

$$= \frac{\sum_{k=0}^{K_\circ} I_{f_\circ}(k+1, N+1-k)/(N+1)}{\sum_{k=0}^{K_\circ} I_F(k+1, N+1-k)/(N+1)}$$

$$= \frac{\sum_{k=0}^{K_\circ} I_{f_\circ}(k+1, N+1-k)}{\sum_{k=0}^{K_\circ} I_F(k+1, N+1-k)}$$

where $I_{f_\circ}(k+1, N+1-k)$ is the incomplete Beta ratio function.

In each of the above cases, we assume that neither we nor the testor know anything more about the prior probability function than the interval of support $[0, F]$, where we assume that $F \geq f_\circ$, since if not, there would be no point to perform an acceptance test to establish something we already know. In this case, we take $Prob\ (f)$ to be that probability function defined on the interval $[0, F]$ having highest entropy. Such a $Prob\ (f)$ is the uniform density on the interval $[0, F]$. Hence, we take $Prob\ (f) = \frac{1}{F}$, $0 \leq f \leq F$. Therefore

$$Prob\ (f \leq f_\circ \mid \hat{f} \leq f_\circ) = \frac{\int_{f=0}^{f_\circ} \sum_{k=0}^{K_\circ} \binom{N}{k} f^k (1-f)^{N-k} df/F}{\int_{f=0}^{F} \sum_{k=0}^{K_\circ} \binom{N}{k} f^k (1-f)^{N-k} df/F}$$

$$= \frac{\sum_{k=0}^{K_\circ} \binom{N}{k} B(k+1, N+1-k) I_{f_\circ}(k+1, N+1-k)}{\sum_{k=0}^{K_\circ} \binom{N}{k} B(k+1, N+1-k) I_F(k+1, N+1-k)}$$

$$= \frac{\sum_{k=0}^{K_\circ} I_{f_\circ}(k+1, N+1-k)/(N+1)}{\sum_{k=0}^{K_\circ} I_F(k+1, N+1-k)/(N+1)}$$

$$= \frac{\sum_{k=0}^{K_\circ} I_{f_\circ}(k+1, N+1-k)}{\sum_{k=0}^{K_\circ} I_F(k+1, N+1-k)}$$

where $I_{f_\circ}(k+1, N+1-k)$ is the incomplete Beta ratio function.

If $f_o << 1$ and $k \leq Nf_o$, then $\frac{2f_o(N+1-k)}{1-f_o} = 2f_oN = 2K_o$. Therefore

$$Prob\left(\mathcal{F}_{2(k+1),\ 2(N+1-k)} \leq \frac{f_o(N+1-k)}{(1-f_o)(k+1)}\right) = Prob\left(\mathcal{X}^2_{2(k+1)} \leq 2K_o\right).$$

Since

$$Prob\left(\mathcal{X}^2_{2(k+1)} \leq 2K_o\right) = \sum_{i=k+1}^{\infty} e^{-K_o}(K_o)^i/i!$$

(Johnson and Kotz 1969, p. 114) we may use tables of the cumulative Poisson distribution (Pearson and Hartley 1958) and there results

$$Prob\ (f \leq f_o \mid \hat{f} \leq f_o) = \sum_{k=0}^{K_o}\left(\sum_{i=k+1}^{\infty} e^{-K_o}K_o^i/i!\right) \Big/ \sum_{k=0}^{K_o}\left(\sum_{i=k+1}^{\infty} e^{-K_1}K_1^i/i!\right)$$

where $K_1 = FN$. When $F >> \frac{k+1}{N+2}$, the value of $I_F(k+1,\ N+1-k) = 1$ since the variance of a Beta $(k+1, N+1-k)$ random variable will be smaller than $\frac{k+1}{(N+1-k)^2} << F$ for large N. In this case, the denominator is only a few percent smaller than K_o+1. ¿From the form of the Poisson approximation, it is apparent that $I_f(k,\ N+1-k)$ depends only on the product fN when $N >> 1$, $k \leq f_oN$ and $f << 1$. This can also be seen directly from the formula.

Under the particular conditions we are interested in, $N >> 100$, $f << 0.1$, and $k << N$. Hence $I_f(k+1,\ N+1-k) \approx I_f(k+1,\ N)$. This can be observed from the recurrence relation

$$I_x(a,b) = xI_x(a-1,b) + (1-x)I_x(a,b-1).$$

Now when $a+b > 6$ and $x << 1$, $I_x(a,b) \approx \phi(y)$ where

$$y = \frac{3\left\{(bx)^{\frac{1}{3}}(1-1/9b) - [a(1-x)]^{\frac{1}{3}}(1-1/9a)\right\}}{\left[\frac{(bx)^{\frac{2}{3}}}{b} + \frac{[a(1-x)]^{\frac{2}{3}}}{a}\right]^{\frac{1}{2}}}$$

and ϕ is the cummulative normal (0,1) distribution function (Abramovitz and Steger, 1972). From this approximation it follows that when $f_oN > 1$, $x << 1$, $f_omk << 1$, $\frac{N}{m} >> 1$, and $m > 1$; then

$$\begin{aligned} I_{f_o}(k+1\ N+1-k) &\approx I_{f_o}(k+1,\ N) \\ &\approx I_{mf_o}(k+1,\ \frac{N}{m}) \\ &\approx I_{mf_o}(k+1,\ \frac{N}{m}+1-k). \end{aligned}$$

This means that instead of having to parametrize by f_o and N independently,we can create tables parametrized by the product f_oN.

For example, if $f_o = 0.0001, F \geq 10f_o$ and $N = 10^4$, then $K_o = 1$ and $Prob\ (f \leq f_o \mid \hat{f} \leq f_o) = \frac{1}{2}[0.6321 + 0.2642] = 0.4481$. If $f_o = .0001, F \geq 10f_o$

and $N = 2 \times 10^4$, then $K_\circ = 2$ and $Prob\ (f \leq f_\circ \mid \hat{f} \leq f_\circ) = \frac{1}{3}[0.8647 + 0.5940 + 0.3233] = 0.5940$. This means that with 2 or fewer observed incorrect recognitions out of 20,000 observations, the probability is only 0.5940 that the true false alarm rate is less than 0.0001.

It seems that such a policy does not provide very certain answers. Perhaps more observations would be helpful. If $N = 10^5$, then $K_\circ = 10$. In this case

$$\begin{aligned} Prob\ (f \leq f_\circ \mid \hat{f} \leq f_\circ) = \frac{1}{11}[&10.0000 + 0.9995 + 0.9972 + 0.9897 \\ &+ 0.9707 + 0.9329 + 0.8699 + 0.7798 + 0.6672 \\ &+ 0.5461 + 0.4170] \\ &= 0.8332 \end{aligned}$$

Thus, with 10 or fewer observed incorrect recognitions out of 100,000 observations, the probability is 0.8336 that the incorrect recognition rate is less than 0.0001. This is certainly better, but depending on our own requirement for certainty in our judgment it may not be sure enough.

Thus the acceptance test itself has a requirement: the probability with which we wish the acceptance test itself to yield correct judgement.

If we adopt a different policy, we can be more sure about our judgment of the true false alarm rate. Suppose we desire to perform an acceptance test which guarantees that the probability is α that the machine meets specifications. In this case, we adopt the policy that we accept the machine if $\hat{f} \leq f^*$ where f^* is chosen so that for the fixed probability $\alpha(f^*)$, $Prob\ (f \leq f_\circ \mid \hat{f} \leq f^*) = \alpha(f^*)$. This means to accept if $K \leq K^*$, where $K^* = Nf^*$. Proceeding as before to find K^*, we have

$$\begin{aligned} \alpha(K^*) = Prob\ (f \leq f_\circ \mid \hat{f} \leq f^*) &= \frac{\int_{f=0}^{f_\circ} \sum_{k=0}^{K^*} \binom{N}{k} f^k (1-f)^{N-k} df}{\int_{f=0}^{F} \sum_{k=0}^{K^*} \binom{N}{k} f^k (1-f)^{N-k} df} \\ &= \frac{\sum_{k=0}^{K^*} I_{f_\circ}(k+1, N+1-k)}{\sum_{k=0}^{K^*} I_F(k+1, N+1-k)} \\ &= \frac{\sum_{k=0}^{K^*} \sum_{i=k+1}^{\infty} e^{-K_\circ} K_\circ^i / i!}{\sum_{k=0}^{K^*} \sum_{i=k+1}^{\infty} e^{-K_\circ} K_\circ^i / i!.} \end{aligned}$$

Then if $f_\circ = .0001, F \geq 10 f_\circ, N = 10^5$, and $K^* = 8$, there results

$$\begin{aligned} \alpha\ (K^*) = \frac{1}{9}[&1.000 + .9995 + .9972 + .9897 + .9707 + .9329 + .8699 \\ &= +.7798 + .6672] \\ &= .9119 \end{aligned}$$

So if $\hat{f} \leq 8/10^5$, the probability will be .9119 that the true incorrect recognition is less .0001.

In summary, we have obtained that $Prob\ (f \leq f_\circ \text{ and } K \leq K^*) = \frac{1}{N+1} \sum_{k=0}^{K^*} I_{f_\circ}(k+1, N+1-k)$ and $Prob\ (K \leq K^*) = \frac{K^*+1}{N+1}$. Since $Prob\ (f)$ is uniform, $Prob\ (f \leq f_\circ) = f_\circ$. These three probabilities determine the missed acceptance rate $Prob\ (f \leq f_\circ \mid K > K^*)$, the false acceptance rate $Prob\ (K \leq K^* \mid f > f_\circ)$, the error rate $Prob\ (f \leq f_\circ \text{ and } K > K^*) + Prob\ (f > f_\circ \text{ and } K \leq K^*)$, the identification accuracy $Prob\ (f \leq f_\circ \text{ and } K \leq K^*) + Prob\ (f > f_\circ \text{ and } K > K^*)$, the acceptance capture rate $Prob\ (K \leq K^* \mid f \leq f_\circ)$, and the capture certainty rate $P(f \leq f_\circ \mid K \leq K^*)$. Thus, the complete operating characteristics can be determined of the acceptance policy we have discussed.

4 Sampling Variations

Most recognition algorithms have their internal tuning parameters set to a value based on a training set. A training set consists of the images or subimages of the entities of interest along with their correct identifications. If the entire population of entities could be sampled and used for training, the value of the tuning parameter would be Θ. However, due to the finite sampling of the training set, the value obtained by training is $\hat{\Theta}$. By sampling the exact same size sample another time, the value obtained by training would differ from $\hat{\Theta}$. This issue of sampling variations is exactly put by asking how much variation will there be in the value $\hat{\Theta}$ due to the finite sample size, if we were to repeat the training process many independent times. For example, if we were to repeat the training process N times, and observe tuning parameter values $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$, we could obtain a reasonable estimate of the covariance matrix Σ for $\hat{\Theta}$ by

$$\Sigma = \frac{1}{N} \sum_{n=1}^{N} (\hat{\Theta}_n - \Theta^*)'(\hat{\Theta}_n - \Theta^*)$$

where

$$\Theta^* = \frac{1}{N} \sum_{n=1}^{N} \hat{\Theta}_n$$

Then we could estimate the probability $P(\delta)$ that $\hat{\Theta}$ will lie within an hyperellipsoid of parameter δ

$$(\hat{\Theta}_n - \Theta^*)' \Sigma^{-1} (\hat{\Theta}_n - \Theta^*) \leq \delta$$

by simply counting:

$$P(\delta) = \frac{1}{N} \#\{n \mid (\hat{\Theta}_n - \Theta^*)' \Sigma^{-1} (\hat{\Theta}_n - \Theta^*) \leq \delta\}$$

For any interval Δ, we could then compute the fraction f of time that the training sample would yield a tuning parameter value in the hyperellipsoid annulus of

width Δ around the point δ.

$$f(\delta;\Delta) = P(\delta + \Delta/2) - P(\delta - \Delta/2)$$

Associated with the subset

$$S = \{n \mid \delta - \Delta/2 \leq (\hat{\Theta}_n - \Theta^*)'\Sigma^{-1}(\hat{\Theta}_n - \Theta^*) \leq \delta + \Delta/2\}$$

is the corresponding set of tuning parameter values.

$$T = \{\Theta_n \mid n \in S\}$$

Now consider taking group of Q testing samples for which we test the algorithm when the tuning parameter values are set to a value in T. For each of the $Q\#T$ combinations, there will be an observed incorrect recognition rate. Thus for a given δ and Δ, there will be a distribution of observed incorrect recognition rates and an associated mean μ incorrect recognition rate and variance σ^2 of incorrrect recognition rate. As we change δ we can observe the dependency of μ on δ. From this dependency we can determine the variation in performance due to testing set sampling variation.

If in doing this kind of analysis we find that the variation in the system tuning parameters caused by training set sampling variations induces large changes in recognition accuracy, then it suggests that the algorithm is not robust and probably undertrained. Such a finding would lead us to more carefully inspect the reasons for the failures and we would take pains to redesign the algorithm as well as use larger sample size for training.

References

1. Abramovitz, M. and Stegum, I. (Eds.), *Handbook of Mathematical Functions,* Dover Publications Inc., New York, 1972.
2. Devijver, P. A. and Kittler, J., *Pattern Recognition: A Statistical Approach,* Prentice Hall International, London, 1982.
3. Fununaga, K., *Introduction to Statistical Pattern Recognition,* Academic Press, New York, 1972.
4. Jockel, K. H., "Finite Sample Properties and Asymptotic Efficency of Monte Carlo Tests," *The Annals of Statistics, Vol. 14,* 1986, pp. 336-347.
5. Johnson, N. and Kotz, S., *Discrete Distributions,* Hougton Mifflin Co., Boston, 1969.
6. Johnson, N. and Kotz, S., *Continuous Univariate Distribution–2,* Houghton Mifflin Co., Boston, 1970.
7. Marriott, F. H. C., "Barmarks Monte Carlo Tests: How many simulations?", *Applied Statistics, Vol. 28,* 1979, pp. 75-77.
8. Pearson, E. S. and Hartley, E. O., *Biometrical Tables for Statisticians, Vol. 1,* (2nd Edition) Cambridge University Press, London, 1958.
9. Pearson, K., (Ed.), *Tables of Incomplete Beta Function, 2nd Edition,* Cambridge University Press, London, 1968.