Statistica Sinica **7**(1997), 211-220

THE EVOLUTION OF A PROBLEM

Henry S. Baird and Colin L. Mallows

Bell Laboratories, AT &T Laboratories

Dedicated to Herbert Robbins on the occasion of his 80th birthday

Abstract: This paper describes several problems, all arising from one real-world problem. Some of these problems have been solved, others offer interesting challenges.

Key words and phrases: Decision trees, Good-Turing, OCR, optimal stopping.

Problem 1.

In a context that will be explained below, we came across the following stopping-rule problem. Consider a stochastic source S_0 that writes a sequence in $\{0,1\}^*$, e.g.

$$01110101111100111111110111\cdots,$$

where 1's occur independently with probability p. This is a sequence of Bernoulli trials so long as p is constant. However, in our case the probability of success is not constant. Whenever a failure ('0') occurs, the source is modified so that the success rate increases. Thus, after the first '0' is seen, the source S_0 is modified, giving a new source S_1 whose probability of success is $p_1 > p_0$. This occurs after every failure. We know that the probability of success rises asymptotically to 1. We have a target value t (say, .99). Ideally, we would like a stopping rule τ that would ensure that $p_\tau \geq t$. An attempt to formulate this requirement is:

$$\text{Minimize } E([\,p_\tau < t] + \lambda\tau)$$

(where [ ] is the indicator function). A standard computation shows that the optimum τ is given by

$$\text{Continue iff } \sum_{i=0}^{\infty} \frac{\lambda}{1-p_i}[\,p_i < t] < 1,$$

but this rule is useless, since we do not know the p's. For a more practical formulation, define a confidence coefficient α (say, .05) and require:

$$\text{Among rules with the property } \; P(p_\tau \geq t) \geq 1-\alpha, \text{ minimize } E(\tau).$$

A first suggestion is the following. Choose an integer r, and stop as soon as a run of 1's of length $\geq r$ occurs. The probability of this is p^r where p is the unknown but constant probability of success during the run. If we choose r so that $t^r < \alpha$ then for all $p < t$ the probability of getting a run as long as this is smaller than α, suggesting that this stopping rule may do the trick.

But this cannot be. The worst case is when there are a very large number of p's that are increasing very slowly, and all just less than the required target t. Then no matter how large we make r, if the number of such p's is large enough, with very high probability a long run will occur and the stopping rule will be invoked with p still less than t. The confidence coefficient of this procedure is zero!

Evidently to achieve a procedure with the desired confidence property, we must ensure that in the limit of the situation described above, i.e. when there are infinitely many p's all just less than t, we have probability $1-\alpha$ of never stopping at all. Robbins and co-workers (for references to fourteen of his papers see the review by Lai and Siegmund (1986)) have studied "tests of power one" that have this flavor, but as far as we are aware they have only considered parameters that stay fixed throughout the procedure.

A referee points out that the problem of estimating the size N of a finite population, which was studied by Darling and Robbins (1967), is similar to our problem, with p_k the probability of choosing a population element that has already been seen. In this case the only unknown is N, and after seeing k different population elements we know that $p_k = k/N$. In our problem we do not know a parametric form for p_k.

We can arrange that when there are infinitely many p's just less than t, with probability at least $1 - \alpha$ we will never stop. We can achieve this by letting the required lengths of the success-runs increase. Let R_j be the length of the jth run of successes. Suppose we stop as soon as one of the following events happens:

$R_1 \geq r_1$ or
$R_1 < r_1$ and $R_2 \geq r_2$ or
$R_1 < r_1$ and $R_2 < r_2$ and $R_3 \geq r_3$

etc. (We will stop with $R_j = r_j$ for some j). If we take

$$r_j = (2\log(j) + c)/\log(1/t) \tag{1}$$

we will have $P(\text{never stop}|p_1 = p_2 = \cdots = t) = \prod_{j=1}^{\infty} P(R_j < r_j) = \prod(1 - t^{r_j})$ and this product is convergent since $\sum t^{r_j} = e^{-c}\sum 1/j^2$ which is convergent. By changing c we can make the product anything we like between 0 and 1. For example taking $t = .99$ and $c = 3.478$ the r_j's are (rounding up):

347 485 565 622 667 703 734 760 784 805 824 841 857 872 885 898 910 922 933 943 $\cdots$

and $\prod_1^k(1-t^{r_j}) = .95$. Thus this sequence of r's defines a stopping rule with the property that if when we stop we assert that $p \geq t$, we will make a mistake with probability at most 0.05. If the p's never get above t, with probability at least 0.95 we will keep on sampling indefinitely. If some p is $> t$, we are certain to stop eventually. Since in the original statement of the problem we were assured that the p's do indeed get to 1 eventually, this does the trick. We do not know any optimality properties of this procedure, or indeed of any procedure of this type. See Darling and Robbins (1967) equation (10) for another rule of this type.

If we know (or believe we know) something about the p's, the procedure can be tuned to make it more efficient. But to keep the procedure practical, we have to be careful not to build in too much information. For example, if we "knew" that at most k_0 p's were below t, we could simply sample until we have seen k_0 failures, but this procedure would be fragile (not valid if the assumption is wrong).

We could also implement a "backwards cusum" procedure, in which we choose a doubly-indexed sequence r_{ij} and stop the first time one of these events happens:

$R_i \geq r_{i1}$ or
$R_i + R_{i-1} \geq r_{i2}$ or
$R_i + R_{i-1} + R_{i-2} \geq r_{i3}$ etc.

Rules of this type were studied by Pollak and Siegmund (1975), see particularly (56) of their paper. However they were considering detecting a single increment in an unknown parameter, and their results do not seem to apply directly to our problem.

The problem we are considering is similar to a standard sequential testing situation. Suppose we specify two thresholds t_1, t_2 and a probability α. Standard Wald theory enables us to set up a stopping rule, determined by the lengths of success runs, with these properties:

$$\text{if the } p\text{'s stay below } t_1 \text{ then } P \text{ (never stop)} \geq 1-\alpha$$

$$\text{if the } p\text{'s ever get above } t_2 \text{ then } P \text{ (stop)} = 1.$$

Suppose that by trial N we have seen K failures. The rule is of the form:

$$\text{Stop when } N \log \frac{t_2}{t_1} > K \log \frac{t_2(1-t_1)}{t_1(1-t_2)} + a. \tag{2}$$

Now letting $t_2 \to t_1 \to t$ this becomes:

$$\text{Stop when } N > K/(1-t) + a.$$

Using this rule, we are sure to stop if p ever gets above t, while if p stays below t, $P(\text{stop}) < 1$ and can be controlled by choosing a appropriately. We have not

studied this rule further, since its performance will depend strongly on the initial p's.

We obtain a more attractive rule by turning the SPRT into a CUSUM procedure. Choose λ and a, and stop (at N) as soon as for some $n < N$,

$$K_n \geq K_n + a + \lambda(n - N).$$

Note that if we take $\lambda = 1/(1-t)$ and the p's stay just below t for a very long time, we have very high probability of stopping before p reaches the target t. While there is a large literature on CUSUM procedures, we have not seen any work relating to the present context.

Clearly very many stopping rules are possible. It is not clear how to choose among them. It is time to find out what the real problem is.

Problem 2.

The real problem is that of Optical Character Recognition (OCR)-teaching a machine to read printed text. There are many difficulties-many kinds of deformations occur, and there are imaging defects due to printing, optics, spatial quantization, etc. Each printed symbol (say an upper-case Times-Roman 'R') differs slightly from almost all other instances of that class of symbols, in size, orientation, and clarity. Also many different typefaces are in use. We need a classification method that is both fast and extremely reliable. For an introduction to the field, see the collection edited by Baird et al. (1992).

One of us has implemented a pre-classification procedure, in which a very fast binary decision tree is constructed (from training data) and used to winnow out the less-likely readings of each symbol, preparatory to a slower and more careful computation that decides among the remaining possibilities. In this preprocessing decision tree, each leaf is labeled with a set of possible class-names. Ideally, each leaf will have just one label. The tree will be effective if it simultaneously reduces the number of possibilities to be considered, and makes it very unlikely that a character is assigned to a leaf that does not carry the correct class-name. For simplicity of analysis, and somewhat pessimistically, we assume here that an error at this preliminary stage cannot be corrected in later stages of classification or contextual analysis.

It is often possible to construct shallow preclassification trees with an acceptably low probability of error. However, trees that are deeper and more strongly pruning-and thus offering greater speed-up at the later stages-often exhibit unacceptable error-rates. This suggests the possibility that the essential problem is not that the greedy tree-building heuristic is sub-optimal; it is that the training data is too sparse. The tree becomes unreliable as it deepens simply because the leaves are each only sparsely occupied.

This conjecture motivated an experiment in which we built a tree using 1,066,639 training samples; this gave a tree with 933 leaves, which was perfect on the training set. The pruning factor was 9; that is, the expected number of classes at a leaf was 1/9 of the maximum number possible, averaged over the training set. In our trials, a total of 4034 classes–typeface/symbol combinations–were possible, so a tree with a pruning factor of 9 reduces these to 447 on average, resulting in significantly faster classification by the OCR system as a whole. However testing on a distinct set of 1,030,000 samples revealed a 15% error rate (and a pruning factor of 9.3). We therefore tried the effect of deleting all the labels on the leaves, and reassigning them using a new training set. In this computation the decision tree itself is not changed; only the labels on the leaves. From here on, we consider only a single class of symbols (for example, Times-Roman 'R'. For brevity, we say simply 'R'). The problem is to decide which leaves should be labeled with this class.

We can regard the successive test cases as a series of trials, where a "success" occurs if a test case (of the class being studied) is assigned to a leaf that is already labeled with this class; if this leaf has not been visited previously, the class-label is now added to the class-labels already assigned to this leaf. Thus the probability that a properly-labeled leaf is hit at the next trial is increased. Problem 1 which we studied above arises as an idealization, where we regard the number of leaves as unbounded. But since we actually know the total number of leaves, the asymptotic difficulties we encountered are seen to be irrelevant. Also, we now see that a loss function of the form

$$(P_\tau - t)^2 + \lambda\tau \tag{3}$$

might be more appropriate, since we would be happy with a rule that stopped with P_τ close to t. We do not want to make P_τ much larger than t, since this cuts down the utility of the decision tree (reduces its pruning factor). Also, we do not want P_τ too small, since (we are assuming) errors at this pre-processing stage cannot be recovered from. This formulation suggests a large new class of procedures for Problem 1, which we have not studied. Instead, we turn to a classical result due to Turing (see Good (1953)).

Suppose we have n cells with unknown probabilities p_i, $i = 1, \ldots, n$ (These are not the same p's as in Problem 1). We throw N balls into these cells, getting Z_i in the ith cell, $i = 1, \ldots, n$. Consider the set of cells that are each hit exactly k times; call this the kth block of cells. The number of cells in the kth block is $n_k^{(N)} = \sum_{i=1}^n [Z_i = k]$ and the empirical relative frequency of these cells is $Q_k^{(N)} = (k/N)n_k^{(N)}$. Let $P_k^{(N)}$ be the total true probability of these cells, $P_k^{(N)} = \sum_{i=1}^n p_i[Z_i = k]$. Note that $P_k^{(N)}$ is a random quantity. Thus $P_0^{(N)}$ is the total

probability of the cells that have not been visited. Turing's result is that $P_{k-1}^{(N)}$ and $Q_k^{(N)}$ are approximately equal; in fact we have exactly, no matter what the p's are,

$$E(P_{k-1}^{(N-1)}) = \sum_{i=1}^{n} p_i \binom{N-1}{k-1} p_i^{k-1}(1-p_i)^{N-k} = \frac{k}{N}\sum_{i=1}^{n}\binom{N}{k} p_i^k(1-p_i)^{N-k} = E(Q_k^{(N)}). \tag{4}$$

Thus if we can keep track of the size of the "1" block of leaves, those that have been hit exactly once, we will have an estimate of the total probability of the leaves that have not been visited.

Robbins (1968) has shown that it is not simply in expectation that $P_0^{(N-1)}$ and $Q_0^{(N)}$ are close; he shows that

$$E(P_0^{(N-1)} - n_1^{(N)}/N)^2 < 1/N. \tag{5}$$

Thus the Chebyshev inequality gives us a crude confidence interval for P_0. Robbins remarks that the inequality (5) can certainly be improved, and similar inequalities for higher moments may yield shorter intervals.

Can we turn this result into a stopping-rule? A simple rule would be: stop at N^* where

$$N^* = \text{smallest } N \text{ such that } n_1^{(N)}/N \leq 1 - t.$$

But this will not work; if (as is actually the case in practice) there are a few cells with large probabilities; then when N is small n_1 is very volatile. For example, it will be quite likely that when $N = 2$ both trials land in the same cell, so $n_1 = 0$ and the stopping rule will be invoked immediately. We have to build in an initial transient stage to allow n_1 to get safely above $(1-t)N$ before starting to apply the rule. In our application we have about 1000 cells, and a transient of 1000 is adequate; typically this makes n_1 about 40, so that n_1/N is about .04, safely above the target .01. From this point on, for a very long time n_1 stays roughly constant as N increases, so n_1/N decreases slowly (but not completely monotonely).

We can also consider stopping at N^{**}, the last time $n_1^{(N)}/N \geq 1 - t$, or at the N half-way between N^* and N^{**}. Note that strictly these are not proper stopping rules, since they do not depend only on the past. In practice there is no difficulty in implementing them. When $n_1^{(N)}/N$ has got as small as $(1-t)/2$ there is very little chance that it will rise above $1 - t$ again.

We have studied these procedures by simulation. We considered n=1000 cells, and assigned probabilities generated from a symmetric Dirichlet distribution $D(n, \gamma)$, i.e. with density proportional to

$$p_1^{\gamma-1}\ p_2^{\gamma-1} \cdots p_n^{\gamma-1} \tag{6}$$

on the unit simplex $\sum p_i = 1$. Guided by our real data, we chose $\gamma = .04$ and .05. For each value of γ, running 300 trials of the procedure, (10 for each of 30 realizations of the Dirichlet p's) we found

$\gamma = .04$	ave(N)	s.d.(N)	ave(P_0)	s.d.(P_0)
first below	3252	587	.01055	.00283
last above	3366	544	.00991	.00242
mid-point	3309	555	.01019	.00255
$\gamma = .05$	ave(N)	s.d.(N)	ave(P_0)	s.d.(P_0)
first below	3978	613	.01064	.00232
last above	4123	621	.01002	.00216
mid-point	4051	603	.01031	.00218

QQ-plots showed that the P_0 values are approximately Gaussian. Variability among the Dirichlet realizations was negligible. These results are a little disappointing. While all three rules hit the target of .01 pretty closely on the average, the values of P_0 are more variable than we would like.

Problem 3.

Reconsideration of the OCR problem suggests a different formulation, and a more attractive class of procedures. We have a (fixed) decision tree, and a supply of test cases. For the present, assume these are all of one class, say 'R'. For test-cases of this class, the ith leaf has probability p_i. We have a target $t\,(= .99)$. We want to label each leaf either "R" or "not R", in such a way that as few leaves as possible are labeled 'R', while the total probability $P(\text{'}R\text{'})$ is as close as possible to the target t.

If we knew, or were able to estimate accurately, all the p's, we would simply sort them and assign the label 'R' in decreasing order of size, stopping when the target t is attained. Since we don't know the p's, we resort to experimentation, using a (pseudo-random) sequence of test cases. The problem is to find a stopping rule τ and an associated rule for labeling the leaves of the tree. Let $R_0^{(\tau)}$ be the total probability of the unlabeled leaves, when this pair of rules is applied. We define the loss function

$$L = (R_0^{(\tau)} - \bar{t}\,)^2 + \lambda\tau \tag{7}$$

($\bar{t} = 1 - t$) and we want to minimize the expected loss. A more realistic loss function would involve the number of unlabeled leaves, (which affects the pruning factor of the decision tree), or even the outcomes of trials involving all the classes of test-cases, but we have not tried to deal with this. We have seen that using Turing's result (4) in a straightforward way (using a stopping rule based on $n_1^{(N)}$) makes $R_0\,(= P_0)$ rather variable. The following procedure reduces the variability

in simulations. Define

$$cum\, Q_k^{(N)} = \sum_{i=1}^{k} Q_i^{(N)}$$

$$cum\, P_k^{(N)} = \sum_{i=0}^{k} P_i^{(N)}.$$

We determine K so that $cum\, Q_{K+1}^{(N)} = 1 - t$, and to choose to label (as"not R") the blocks $0, 1, \ldots, K$. This should make $cum\, P_K^{(N)}$ approximately equal to $1 - t$. In practice, we must allow K to take non-integer values, by interpolating as necessary; if we find $K = k + f$ where $0 \le f < 1$ we label blocks $0, 1, \ldots, k$ and a random fraction f of the leaves in the $k+1$st block.

We can study the properties of this procedure if we make the very convenient assumption that the p's have the Dirichlet prior distribution (6). Then the posterior distribution of the p_i's is Dirichlet with density proportional to $\prod_{i=1}^{n} p_i^{\gamma + Z_i - 1}$. Thus the posterior distribution of the quantity $cum\, P_k^{(N)}$ is simply a beta distribution with exponents $W_k^{(N)} - 1$, $n\gamma + N - W_k^{(N)} - 1$, where $W_k^{(N)} = \sum_{j=0}^{k} n_j^{(N)}(\gamma + j)$. The posterior expectation of the loss (7) is

$$\Big[\frac{W_k^{(N)}}{n\gamma + N} - \bar{t}\Big]^2 + \frac{W_k^{(N)}(n\gamma + N - W_k^{(N)})}{(n\gamma + N)^2(n\gamma + N + 1)} + \lambda N.$$

This suggests that (if γ is known), the optimal rule is very nearly this deterministic rule:

$$\text{choose } N \text{ to minimize } t(1-t)/(n\gamma + N) + \lambda N$$

$$\text{label (as "not } R\text{") } K \text{ blocks, where } W_k^{(N)} \approx (n\gamma + N)(1 - t).$$

Choosing λ is equivalent to choosing the variance of the posterior distribution. We expect that if we run N trial cases, the mean square error of $cum\, P_K^{(N)}$ will be about $t(1-t)/(N + n\gamma)$.

In practice, we do not know γ, or even that the p's have a Dirichlet distribution. In fact we have evidence that the Dirichlet is not an appropriate model; estimating γ by maximum likelihood from one sequence of trials gave these estimates:

N	1019	2045	5152	10311	20619	51552
$\hat{\gamma}$	.0315	.0372	.0390	.0419	.0462	.0501

which show a clear trend. With $n = 1000$ and $\gamma = .05, n\gamma$ is much smaller than our $N = 10000$ so we might hope that the value of γ is not critical. But we do not want to have to rely on the Dirichlet assumption. One possibility would be

to use a more flexible (less informative) prior than the Dirichlet. Since we can generate the Dirichlet(γ) distribution by taking $p_i = X_i / \sum X$ where $X_1, \ldots, X_n$ are independent Gamma(γ), we might replace Gamma by a Beta distribution of the second kind, i.e. an F distribution (see Kempton (1975)). However this is not very tractable.

Reverting to a frequentist approach, if we assume the number of trials is Poisson with mean N, it is straightforward to derive expressions for the variances and covariances of P_k, $k = 0, 1, \ldots, K-1$ and Q_k, $k = 1, \ldots, K$, and hence of $cum\, P_{K-1}$ and $cum\, Q_K$. Hence, assuming these are approximately bivariate normal, we obtain an approximation for the variance of $cum\, P_{K-1}$ conditional on $cum\, Q_K = 1 - t$, namely

$$\frac{1}{N}\Big[kS_{k+1} + \frac{[\sum_{j=1}^{k} S_j][\sum_{j=2}^{k}(j-1)S_j - T_k]}{\sum_{j=1}^{k} jS_j - T_k}\Big], \tag{8}$$

where

$$S_m = E(Q_m) = \frac{m}{N}\sum (Np_j)^m e^{-Np_j}/m!, \quad T_m = \sum_{i=1}^{k}\sum_{j=1}^{k} ij \binom{i+j}{i} U_{i+j}$$

and

$$U_m = \sum (Np_j)^m e^{-2Np_j}/m!.$$

Numerical evaluations show that the relative size of the two terms in (8) depends strongly on the configuration of the p's.

At this point we do not see how to make further progress theoretically. We are unable to reconcile the Good-Turing approach with the Bayesian approach of Hill (1979).

We do have an "engineering" solution to the problem. We choose N by a rule we state below. We run N trials, and estimate P_{k-1} by a weighted straight-line smooth of Q_k, with weights depending on the variances of the individual Q's. We sum to get an estimate $cum\, P_k$, and determine K so that $cum\, P_K = 1 - t$. We label (as "not R") blocks $0, 1, \ldots, K$, interpreting fractional blocks as explained above. By simulation using the Dirichlet model, and by experience with both pseudo-random and real trials on the real decision tree, we have verified that this gives true coverage approximately equal to the target, with a standard error about $1.5\sqrt{t(1-t)/N}$. Thus we simply choose N large enough to give the precision we desire (about 10000 seems right). Applying our method to 3720 <typeface, symbol> pairs, we used 37,200,000 samples to populate the tree. When tested on a distinct set of 18,600,000 samples, the tree's error rate was measured to be 1.05%, a little higher than the target. The effective pruning factor was 3.43, down of course from 9 but still high enough to speed up the OCR system markedly.

Clearly many open questions remain. We do not know how to deal with a loss function that involves the number of labeled leaves. We cannot handle any prior other than Dirichlet. We have made no progress on rules that consider more than one type of symbol, and that attack the pruning factor problem directly. We would like to know how to divide our effort between building the decision tree and labeling its leaves.

Acknowledgement

Thanks to Larry Shepp for commenting on an earlier draft. Thanks also to a referee for his helpful report, which led to several improvements.

References

Baird, H. S., Bunke, H. and Yamamoto, K. (eds.) (1992). *Structured Document Image Analysis*. Springer-Verlag, New York.

Darling, D. A. and Robbins, H. E. (1967). Finding the size of a finite population. *Ann. Math. Statist.* **38**, 1392-1398.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika* **40**, 237-264.

Hill, B. M. (1979). Posterior moments of the number of species in a finite population and the posterior probability of finding a new species. *J. Amer. Statist. Assoc.* **74**, 668-673.

Kempton, R. A. (1975). A generalized form of Fisher's logarithmic series. *Biometrika* **62**, 29-38.

Lai, T. L. and Siegmund, D. (1986). The contributions of Herbert Robbins to mathematical statistics. *Statist. Sci.* **1**, 276-284.

Pollak, M. and Siegmund, D. (1975). Approximations to the expected sample size of certain sequential tests. *Ann. Statist.* **3**, 1267-1282.

Robbins, H. E. (1968). Estimating the total probability of the unobserved outcomes of an experiment. *Ann. Math. Statist.* **39**, 256-257.

Statistical Models & Methods Research Department, AT&T Bell Laboratories, Rm. 2C-265, 600 Mountain Ave., Murray Hill, NJ 07974-0636, U.S.A.

(Received March 1995; accepted September 1996)