Discrete Applied Mathematics 226 (2017) 17–31

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Optimal ordering of statistically dependent tests

Daniel Berend [a], Ronen Brafman [a], Solomon E. Shimony [a], Shira Zucker [b,*], Shimon Cohen [c]

[a] *Departments of Computer Science and Mathematics, Ben-Gurion University, Israel*

[b] *Department of Computer Science, Sapir College, Israel*

[c] *Orbotech, Israel*

A R T I C L E I N F O

Article history:
Received 25 September 2016
Received in revised form 23 February 2017
Accepted 19 April 2017
Available online 13 May 2017

Keywords:
Test ordering problem
Object detection

A B S T R A C T

We consider scenarios where a sequence of tests is to be applied to an object, such that one outcome of a test may be a decision to terminate the sequence (e.g. deciding that the object is faulty) without running additional tests. One seeks an ordering of the tests that is minimal in expected resource consumption. In prior work, we examined conditions under which statistically independent test sequences can be optimized under precedence constraints. This paper examines conditions under which one can efficiently find an optimal ordering of tests with statistical dependencies. We show that with dependencies the optimization problem is NP-hard in the general case, and provide low-order polynomial time algorithms for special cases with non-trivial dependency structures.

1. Introduction

In numerous classes of applications, including object detection and acceptance test applications, an important issue is to classify objects in real-time. For instance, security systems need to detect targets in real-time and act on them. Robots need to quickly decide whether an observed artifact is an obstacle, in order to avoid collision. Factory inspection lines must decide as quickly as possible whether or not a manufactured object is faulty. An important method that addresses this issue for image classification is the feature cascade architecture for rapid object detection [15].

The basic problem is defined as follows. We are given a set of tests (“detectors”) $X = \{x_1, x_2, \ldots, x_n\}$ for a certain property, such as a defect. Each test can return “reject” or “do not know”. Each object is tested by a sequence (or “cascade”) of tests (usually a permutation of *all* the above tests), unless rejected by an earlier test. The tests are assumed to be one-sided perfect, i.e., a “reject” means that the tested object is not a defect, with no errors possible. Also given is an execution time for each test (or any other resource required for the test, but henceforth we assume without loss of generality that the resource to be minimized is time), and a probability of “reject” for each. The problem is to find a sequence of tests in the cascade which has optimal expected runtime. (An abridged version of this paper has been presented at the Modelling, Computation and Optimization Conference in Information Systems and Management Sciences (Berend et al., 2015) [3].)

In the simplest setting, the tests are statistically independent and there is no other structure to the tests, making the problem mathematically equivalent to job sequencing [11]. In related work, introduction of structure of various types is examined, such as precedence constraints [1,5,8,12,2,4], system structure [7], and statistical dependencies [6]. For a survey on the different types of structures and solution techniques see [14].

* Corresponding author.
E-mail addresses: berend@cs.bgu.ac.il (D. Berend), brafman@cs.bgu.ac.il (R. Brafman), shimony@cs.bgu.ac.il (S.E. Shimony), shiraz@mail.sapir.ac.il (S. Zucker).

http://dx.doi.org/10.1016/j.dam.2017.04.009
0166-218X/

In this paper we examine optimization of test ordering with **dependent** tests, with **no** precedence constraints, as a natural continuation of [2] which examined **independent** tests **with** precedence constraints. That is because in most applications tests are correlated, due to common mechanisms, correlated noise, and numerous other reasons. Hence, it is important to be able to optimize sequences of statistically dependent tests, which we undertake in this paper.

Initially, assume as in [2] that the tests are statistically independent. Denote by r_i the marginal probability that test x_i rejects (necessitating no further processing), by t_i the time for test x_i, and by $q_i = \frac{r_i}{t_i}$ the "quality" of the test. We assume that $1 > r_i > 0$ for all i, as with $r_i = 0$ the test is useless, and with $r_i = 1$ we always get rejection, and there is no need for testing in the first place. Using this notation, the rejection rate of a sequence S of statistically independent tests, composed of all tests in any order, is

$$R(S) = 1 - \prod_{i=1}^{n}(1 - r_i). \tag{1.1}$$

The expected runtime of the sequence, assuming that it is ordered according to the initial indexing of the tests, is

$$T(S) = \sum_{i=1}^{n} t_i \prod_{j=1}^{i-1}(1 - r_j) \tag{1.2}$$

(using the convention that a product over an empty set is 1). The "quality" measure for the sequence as a whole is defined as

$$Q(S) = \frac{R(S)}{T(S)}.$$

We similarly define the quality of any sequence of tests, consisting of an arbitrary subset of X.

The dependent test ordering problem is defined as follows. Given the set $X = \{x_1, x_2, \ldots, x_n\}$, for each x_i its runtime t_i, and a joint distribution $\mathbf{r}$ over rejection events, find a subsequence S of the tests from X such that:

1. The expected runtime of S is minimal,
2. If $x_i \in X$ is not in S, the probability that x_i rejects given that none of the tests in S reject is 0.

The disjoint rejection distributions can be specified in various ways, such as complete joint probability table (undesirable), or using a graphical probability model such as a Bayes network or Markov network [13].

Note that in this formulation of the problem, we are allowed to drop some of the tests from the set due to case 2 above, as they provide no useful information. Such tests are called redundant (given the context S). Also observe that:

1. If the tests are independent, then no test x_i can be redundant.
2. If the probability that x_i rejects given that none of the tests in S reject is 1, x_i is also redundant. However, we assume that this cannot occur, as it implies that the overall probability of rejection is 1, and there is no need for testing at all.

In this paper we examine complexity and algorithms for solving the dependent test ordering problems. We begin (Section 2) by showing that for general dependency structures, optimizing the test sequence is NP-hard. This is followed by an observation that the problem is surprisingly complex even for very simple settings, due to a non-locality anomaly. In Section 2.4 we examine a special case where optimal ordering can be done efficiently, specifically negatively correlated dependent pairs and similar structures. Cases of extreme dependencies ("disjoint" tests, and "dominating" tests), and combinations thereof that can be optimized efficiently, are examined in Section 3.

2. Statistically dependent tests—basic properties

Since tests are dependent, we need notation for describing joint and conditional reject probabilities. To denote joint distributions, we use r subscripted by the outcomes of some of the tests. For example, r_{ij} is the probability that tests x_i and x_j both reject. We use bar to denote negation in the subscript, e.g. $r_{\bar{i}j}$ is the probability that test x_j rejects but test x_i does not reject (under the obvious assumption that both tests are performed). We also use conditioning notation in the subscript: $r_{i|S}$ denotes the probability that test x_i rejects given previous occurrences S, where typically S would be the reject/non-reject of previous tests. For example, $r_{i|\bar{j}}$ denotes the probability that test x_i rejects given that x_j has not rejected. Quality values use the same conditioning notation.

2.1. Complexity of optimal ordering

In the general case, the problem of finding the optimal ordering with dependent tests is intractable:

Theorem 1. *The problem of finding the optimal test ordering is NP-hard.*