1995年度日本オペレーションズ・リサーチ学会
春季研究発表会

Error-Free and Best-Fit Extensions of Partially Defined Boolean Functions

ラトガース大学 (Rutgers University) BOROS Endre
01001374 京都大学 茨木俊秀 (IBARAKI Toshihide)
02601514 京都大学 *牧野和久 (MAKINO Kazuhisa)

1 Introduction

In this paper, we address a fundamental problem related to the induction of Boolean logic (e.g., [2, 3]), that is, given a set of data, represented as a set of binary "true n-vectors" (or "positive examples") and a set of "false n-vectors" (or "negative examples"), we have to establish a Boolean function (extension) f with some specified properties, so that f is true (resp. false) in every given true (resp. false) vector.

For instance, data x represent the symptoms to diagnose a disease, e.g., x_1 denotes whether temperature is high ($x_1 = 1$) or not ($x_1 = 0$), and x_2 denotes whether blood presure is high ($x_2 = 1$) or not ($x_2 = 0$), etc. Establishing an extension f, which is consistent with the given data, amounts to finding a logical diagnostic explanation of the given data. Therefore, this may be considered as a form of knowledge acquisition from given examples.

In this process, some knowledge or hypothesis about the extension f may usually be available beforehand. Such knowledge may be obtained from experience or from the analysis of mechanisms that may or may not cause the phenomena under consideration. In the above example of diagnosing diseases, it would be natural to assume that we somehow know the direction of each variable that tends to cause the disease to appear. By changing the polarities of variables if necessary, therefore, the extension $f(x)$ can be assumed to be positive in all variables. In other words, we are asked to establish an extension f, which is a positive Boolean function. In this application, not only the obtained function f itself but also the fact that the given set of data actually has a positive extension are important information to know, since the latter verifies that the assumption on the directions of variables is in fact correct.

Restriction on the functional form of an extension may also arise in a different context. For instance, applications in artificial intelligence often require the extension f to be a Horn function, because such function can be characterized by DNF (disjunctive normal form) of Horn terms, and hence can be realized by Horn rules. These suggests an important problem in this area, that is, determining the existence or nonexistence of an extension f of given data, which is in a given class of Boolean functions. In addition to classes of positive functions and Horn functions, some other classes of functions, such as k-DNF functions, h-term DNF functions, dual-comparable functions, threshold functions, read-once functions, are discussed in this paper.

Unfortunately, the real-world data might contain errors. As for the above examples, measurement error might come in when getting data, or there may be some other factors not represented as variables in the vectors (e.g., some bacteria which cause the disease, in the above example of diagnosis). To cope with such situations, we may have to give up the goal of establishing an extension that is perfectly consistent with the given data. If there is no such extension, the best we can expect is to establish an extension f, which has the minimum error of misclassifications. This problem will also be extensively studied in this paper.

The problem of finding extensions of given data arises in various fields including not only artificial intelligence [2, 3] but also learning theory [1], game theory, and so on.

2 Definitions and Problems

A *Boolean function*, or a *function* in short, is a mapping $f : \{0,1\}^n \mapsto \{0,1\}$, where $x \in \{0,1\}^n$ is called a *Boolean vector* (a *vector* in short). If $f(x) = 1$ (resp. 0), then x is called a *true* (resp. *false*) vector of f. The set of all true vectors (false vectors) is denoted by $T(f)$ ($F(f)$).

A *partially defined Boolean function* (*pdBf*) is defined by a pair of sets (T, F) of Boolean vectors of n variables, where T denotes a set of true vectors (or positive examples) and F denotes a set of false vectors (or negative examples). A function f is called an *extension* (or theory) of the pdBf (T, F) if $T \subseteq T(f)$ and $F \subseteq F(f)$.

Evidently, the disjointness of the sets T and F is a necessary and sufficient condition for the existence of an extension, if any Boolean function f may be used.

It may not be evident, however, to find out whether a given pdBf has a extension in $\mathcal{C}$, where $\mathcal{C}$ is a subclass of Boolean functions, such as the class of positive functions, the class of k-DNF's, etc. Therefore, we first consider the following problem:

Problem EXTENSION($\mathcal{C}$)
Input: a pdBf(T, F), where $T, F \subseteq \{0,1\}^n$.
Question: Is there an extension $f \in \mathcal{C}$ of (T, F)?

Furthermore, for a pdBf(T, F), define the *error size* of a function f by

$$\varepsilon(f) = |\{a \in T \mid f(a) = 0\}| + |\{b \in F \mid f(b) = 1\}|.$$

Based on this, we introduce the following problem:

Problem BEST-FIT($\mathcal{C}$)
Input: a pdBf(T, F), where $T, F \subseteq \{0,1\}^n$.
Output: $f \in \mathcal{C}$ that realizes $\min_{f \in \mathcal{C}} \varepsilon(f)$.

Clearly, problem EXTENSION is a special case of problem BEST-FIT, since EXTENSION has a solution f if and only if BEST-FIT has a solution f with $\varepsilon(f) = 0$. This means that if BEST-FIT($\mathcal{C}$) is solvable in polynomial time (i.e., polynomial in $n, |T|$ and $|F|$), for some class $\mathcal{C}$, then EXTENSION($\mathcal{C}$) is also polynomially solvable; conversely if EXTENSION($\mathcal{C}$) is NP-hard, then so is BEST-FIT($\mathcal{C}$).

A function f is *positive* if $x \leq y$ (i.e., $x_i \leq y_i$ for all $i \in \{1, 2, \ldots, n\}$) always implies $f(x) \leq f(y)$. A positive function is also called *monotone*. The variables $x_1, x_2, \ldots, x_n$ and their complements $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ are called *literals*. A *term* is a conjunction of literals such that at most one of x_i and $\bar{x}_i$ appears for each variable. The constant 1 (viewed as the conjunction of an empty set of literals) is also considered to be a term. A *disjunctive normal form (DNF)* is a disjunction of terms. Clearly, a DNF defines a function, and it is well-known that every function can be represented by a DNF (however, such a representation may not be unique), and f is positive if and only if f can be represented by a DNF, in which all the literals of each term are uncomplemented. A function is called a *k-DNF* if it has a DNF with at most k literals in each term, *h-term DNF* if it has a DNF with at most h terms, and *Horn* if it has a DNF with at most one negative literal in each term.

The *dual* of a function f, denoted f^d, is defined by

$$f^d(x) = \bar{f}(\bar{x}),$$

where $\bar{f}$ and $\bar{x}$ denote the complement of f and x, respectively. As is well-known, a Boolean expression defining f^d is obtained from that of f by exchanging $\vee$ (or) and $\cdot$ (and), as well as the constants 0 and 1. It is easy to see that $(f \vee g)^d = f^d g^d$, and so on. A function f is called *dual-minor* if $f \leq f^d$, *dual-major* if $f \geq f^d$, *dual-comparable* if $f \leq f^d$ or $f \geq f^d$, and *self-dual* if $f^d = f$.

3 Results

In the table below, we summarize the complexity of EXTENSION($\mathcal{C}$) and BEST-FIT($\mathcal{C}$) for various classes $\mathcal{C}$ of functions.

Function classes	EX	B-F
Transitive:		
General	P	P
Positive	P	P
Regular	P	P
Hereditary:		
(Positive) k-DNF	NPC	NPH
(Positive) k-DNF with fixed k	P	NPH
(Positive) h-term-DNF	NPC	NPH
(Positive) h-term-DNF with fixed $h \geq 2$	NPC	NPH
(Positive) 1-term-DNF	P	NPH
(Positive) h-term-k-DNF	NPC	NPH
(Positive) h-term-k-DNF with fixed $h \geq 1$	NPC	NPH
(Positive) h-term-k-DNF with fixed $k \geq 1$	NPC	NPH
(Positive) h-term-k-DNF with fixed h, k	P	P
Horn	P	NPH
Dual-comparable:		
Self-dual	P	P
Dual-minor	P	P
Dual-major	P	P
Positive self-dual	P	NPH
Positive dual-minor	P	NPH
Positive dual-major	P	NPH
Threshold	P	NPH
2-monotonic positive	NPC	NPH

EX: EXTENSION, B-F: BEST-FIT, P: Polynomial, NPC: NP-complete, NPH: NP-hard

Table 1: Summary of results.

References

[1] M. Anthony and N. Biggs, *Computational Learning Theory*, Cambridge University Press, 1992.

[2] Y. Crama, P. L. Hammer and T. Ibaraki, Cause-effect relationships and partially defined boolean functions, *Annals of Operations Research*, 16 (1988) 299-326.

[3] D. Kavvadias, C. H. Papadimitriou and M. Sideri, On horn envelopes and hypergraph transversals, *ISAAC'93* Algorithms and Computation, edited by K. W. Ng et al., Springer Lecture Notes in Computer Science, 762 (1993) 399-405.