Formalizing Statistical Beliefs in Hypothesis Testing Using Program Logic

Yusuke Kawamoto\textsuperscript{1,2}, Tetsuya Sato\textsuperscript{3}, Kohei Suenaga\textsuperscript{4}

\textsuperscript{1}AIST, Japan
\textsuperscript{2}PRESTO, JST, Japan
\textsuperscript{3}Tokyo Institute of Technology, Japan
\textsuperscript{4}Kyoto University, Japan

Abstract

We propose a new approach to formally describing the requirement for statistical inference and checking whether the statistical method is appropriately used in a program. Specifically, we define belief Hoare logic (BHL) for formalizing and reasoning about the statistical beliefs acquired via hypothesis testing. This logic is equipped with axiom schemas for hypothesis tests and rules for multiple tests that can be instantiated to a variety of concrete tests. To the best of our knowledge, this is the first attempt to introduce a program logic with epistemic modal operators that can specify the preconditions for hypothesis tests to be applied appropriately.

1 Introduction

Statistical inference has been widely used to derive and justify scientific knowledge in a variety of academic disciplines, from natural sciences to social sciences. This has significantly increased the importance of statistics, but also brought concerns about the inappropriate procedure and the incorrect interpretation of statistics in scientific research. In fact, previous studies have pointed out that many research articles in biomedical science contain severe errors in the application and interpretation of statistical inference (Lang and Altman 2014). Furthermore, large proportions of these errors have been reported for basic statistical methods, possibly performed by researchers who can use only elementary techniques. For example, the concept of statistical significance, evaluated using \(p\)-values, has been commonly misused and misinterpreted (Wasserstein and Lazar 2016).

One of the main issues behind these human errors is that the logical aspects of statistical inference are described informally or implicitly using natural languages, and handled manually by analysts who may not fully understand the statistical methods. In particular, this makes them overlook some assumptions necessary for statistical methods, hence choosing inappropriate methods. Nevertheless, to our knowledge, no prior work on formal methods has specified the preconditions for statistical inference or verified the choice of statistical techniques.

In this paper, we propose a method for formalizing and reasoning about statistical inference using symbolic logic. Specifically, we introduce belief Hoare logic (BHL) to formalize the statistical beliefs acquired via hypothesis tests, and to prevent errors in the choice of hypothesis tests by describing their preconditions explicitly. This is the first step to build a framework for formalizing and verifying the validity of empirical science on the basis of formal methods.

Contributions. Our main contributions are as follows:

- We propose a new approach to formalizing and reasoning about statistical inference in a program. In particular, this approach formalizes and checks the requirement for statistical methods to be used appropriately.
- We define an epistemic logic to express statistical beliefs obtained by hypothesis tests on datasets. Specifically, we formalize a statistical belief on a hypothesis \(\varphi\) as the knowledge that either \(\varphi\) holds or the sampled dataset is unluckily far from the population. Then we introduce a Kripke semantics to define the interpretation of the logic.
- Using this epistemic logic, we construct belief Hoare logic (BHL) for formalizing and reasoning about the statistical inference based on hypothesis testing. Specifically, we define axiom schemas and rules for hypothesis tests that can be instantiated to a variety of concrete tests. In particular, BHL does not rely on a specific philosophy of statistics but can deal with both the frequentist and the Bayesian statistics by introducing corresponding axioms.
- We show that BHL is useful to reason about practical issues concerning statistical inference, such as the multiple comparison problem and \(p\)-value hacking.
- We provide a whole picture of the justification of statistical belief acquired via hypothesis tests inside and outside BHL. For instance, we discuss the empirical conditions and the epistemic aspects of statistical inference.

To the best of our knowledge, this is the first attempt to introduce a program logic that can specify the preconditions for hypothesis tests to be applied appropriately.

Related Work. The Hoare logic (Winskel 1993) is one of the program logic for an imperative programming language. This program logic is then extended and adapted so that it can handle various types of programs and assertions, including heap-manipulating programs (Reynolds 2002), hybrid systems (Suena and Hasuo 2011), and probabilistic programs (den Hartog and de Vink 2002). Atkinson and Carbin propose an extension of Hoare logic with epistemic assertions (Atkinson and Carbin 2020). In their work, an
epistemic assertion is used to reason about the belief of a program about a partially observable environment, whereas their logic does not deal with a statistical belief arising from statistical tests conducted in a program. To the best of our knowledge, ours is the first program logic that formalizes the concept of statistical beliefs in hypothesis testing.

Epistemic logic (von Wright 1951) is a branch of logic for reasoning about knowledge and belief (Fagin et al. 1995a; Halpern 2003), and is used to specify and verify a variety of knowledge properties in systems, e.g., authentication (Burrows, Abadi, and Needham 1990) and anonymity (Syverson and Stubblebine 1999; Garcia et al. 2005). Many previous works incorporate certain notions of degrees of belief and confidence (Huber and Schmidt-Petri 2008), but not in the sense of the statistical significance in hypothesis testing.

The first attempt to express statistical properties using modal logic is the work on statistical epistemic logic (StatEL) (Kawamoto 2019; Kawamoto 2020). They introduce its statistical model as two normal distributions \(N(\mu_{\text{ppl}}, \sigma^2)\) and \(N(\mu_{\text{ppl}}, \sigma^2)\) with a known variance \(\sigma^2\) and unknown true means \(\mu_{\text{ppl}}, \mu_{\text{ppl}}\). Let \(y_1\) and \(y_2\) be two given datasets where each data value was sampled independently from \(N(\mu_{\text{ppl}}, \sigma^2)\) and \(N(\mu_{\text{ppl}}, \sigma^2)\), respectively.

In the Z-test, we set the alternative hypothesis \(\varphi_1 \overset{\text{def}}{=} (\mu_{\text{ppl}} \neq \mu_{\text{ppl}})\) and the null hypothesis \(\varphi_0 \overset{\text{def}}{=} (\mu_{\text{ppl}} = \mu_{\text{ppl}})\). We calculate the Z-test statistic \(t(y_1, y_2) = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}\) where for \(b = 1, 2\), \(size(y_b)\) is the sample size of \(y_b\) and \(\text{mean}(y_b)\) is the mean of all data in \(y_b\). Then the \(p\)-value is defined by:

\[
Pr_{(d_1, d_2) \sim N(\mu_{\text{ppl}}, \sigma^2) \times N(\mu_{\text{ppl}}, \sigma^2)}(t(d_1, d_2) > t(y_1, y_2))\]

under the null hypothesis \(\varphi_0\). When the \(p\)-value is small enough, the datasets \(y_1, y_2\) are unlikely to be sampled from the same distribution, i.e., the null hypothesis \(\mu_{\text{ppl}} = \mu_{\text{ppl}}\) is unlikely to hold. Hence, in the Z-test, if the \(p\)-value is smaller than a certain threshold (e.g., 0.05), we reject the null hypothesis \(\varphi_0\) and accept the alternative hypothesis \(\varphi_1\).

When we have prior knowledge of \(\mu_{\text{ppl}} \geq \mu_{\text{ppl}}\), we apply the upper-tailed (resp. lower-tailed) Z-test with the alternative hypothesis \(\mu_{\text{ppl}} > \mu_{\text{ppl}}\) (resp. \(\mu_{\text{ppl}} < \mu_{\text{ppl}}\)), and with the \(p\)-value \(Pr(t(d_1, d_2) > t(y_1, y_2))\) (resp. \(Pr(t(d_1, d_2) < t(y_1, y_2))\)).

## 3 Illustrating Example

Throughout the paper, we use the following simple illustrating example to explain the basic ideas on our framework.

**Example 2** (Comparison tests on drugs). Let us consider three drugs 1, 2, 3 that may decrease blood pressure. To compare the efficacy of these drugs, we perform experiments and obtain a set \(y_i\) of the reduced values of blood pressure after taking drug \(i\). Then we apply hypothesis tests on the dataset \(y_i = (y_{i1}, y_{i2}, y_{i3})\). Let \(x_i\) be the true population from which the data values in \(y_i\) are sampled.

Suppose that drug 1 is composed of drugs 2 and 3, and we want to know whether drug 1 has better efficacy than both drugs 2 and 3. Then we take the following procedure:

- We first compare drugs 1 and 2 concerning the average decreases in blood pressure. We apply a two-tailed Z-test to see whether the means of the true populations \(x_1\) and \(x_2\) are different, i.e., \(\text{mean}(x_1) \neq \text{mean}(x_2)\). In this test, the alternative hypothesis is \(\varphi_{12} \overset{\text{def}}{=} (\text{mean}(x_1) \neq \text{mean}(x_2))\), and the null hypothesis is \(\neg \varphi_{12} \overset{\text{def}}{=} (\text{mean}(x_1) = \text{mean}(x_2))\).
- Let \(\alpha_{ij}\) be the \(p\)-value when only comparing drugs \(i\) and \(j\).
- If \(\alpha_{12} \geq 0.05\), the Z-test does not reject the null hypothesis \(\neg \varphi_{12}\) and concludes that the efficacy of drugs 1 and 2 may be the same. Then we are not interested in drug 1 any more, and skip the comparison with drug 3.
• If \( \alpha_{12} < 0.05 \), the Z-test rejects the null hypothesis \( \neg \varphi_{12} \) and concludes that the alternative hypothesis \( \varphi_{12} \) is true. Then we apply another Z-test to check whether the alternative hypothesis \( \varphi_{13} \) is true.

• Finally, we calculate the p-value of the combined tests, with the conjunctive alternative hypothesis \( \varphi_{12} \land \varphi_{13} \).

Note that these Z-tests assume that the distribution of each population \( x_i \) is a normal distribution with a variance \( \sigma_i^2 \).

**Overview of the Framework.** In our framework, we describe a procedure of statistical tests as a program using a programming language (Section 6); in Example 2, we denote the Z-test program comparing drugs \( i \) with \( j \) by \( C_{ij} \), and the whole procedure by:

\[
C_{\text{drug}} \stackrel{\text{def}}{=} C_{12}; \text{if } \alpha_{12} < 0.05 \text{ then } C_{13} \text{ else skip}
\]

Then we use an assertion logic (Section 5) to describe the requirement for the hypothesis tests as a precondition formula. In Example 2, the precondition is given by:

\[
\psi_{\text{pre}} \stackrel{\text{def}}{=} \bigwedge_{i=1,2,3} \psi_i \land \bigwedge_{(i,j)=(1,2),(1,3),(2,1),(3,1)} P(\text{mean}(x_i) < \text{mean}(x_j))
\]

where \( \psi_i \) is defined as \( (x_i \approx N(\mu_i, \sigma) \land y_i \approx N(\mu_j, \sigma)) \).

In this formula, \( \psi_i \) represents that the true population \( x_i \) follows a normal distribution \( N(\mu_i, \sigma) \) (with an unknown true mean \( \mu_i \)), and that a set \( y_i \) of \( n_i \) data is sampled from \( x_i \). \( P(\text{mean}(x_i) < \text{mean}(x_j)) \) and \( P(\text{mean}(x_i) > \text{mean}(x_j)) \) represent that both the lower-tail and upper-tail are possible, hence the test \( C_{12} \) should be two-tailed. Remark that our assertion logic never deals with quantifiers over variables.

The statistical belief we want to acquire is specified as a postcondition formula. In Example 2, the postcondition is:

\[
\varphi_{\text{post}} \stackrel{\text{def}}{=} \big(K_y^{\leq 0.05} \land \varphi_{12} \land K_y^{\leq \min(\alpha_{12}, \alpha_{13})}(\varphi_{12} \land \varphi_{13}))\]

Intuitively, by testing on a dataset \( y \), when we believe \( \varphi_{12} \) with a p-value \( \alpha \leq 0.05 \), we believe the combined hypothesis \( \varphi_{12} \land \varphi_{13} \) with a p-value at most \( \min(\alpha_{12}, \alpha_{13}) \).

Finally, we combine all the above and describe the whole statistical inference as a judgment. In Example 2, we write:

\[
\Gamma \vdash \{ \psi_{\text{pre}} \} C_{\text{drug}} \{ \varphi_{\text{post}} \}
\]

By proving this judgment using rules in BHL (Section 7), we conclude that the statistical inference is appropriate.

We remark that the p-value can be larger for a different purpose of testing. Suppose that in Example 2, drug 1 was a new drug and we wanted to find out it had better efficacy than at least one of drugs 2 and 3. Then the procedure is:

\[
C_{\text{multi}} \stackrel{\text{def}}{=} C_{12} \parallel C_{13}
\]

and the alternative hypothesis is \( \varphi_{12} \lor \varphi_{13} \) with a p-value larger than \( \alpha_{12} \) and \( \alpha_{13} \) (at most \( \alpha_{12} + \alpha_{13} \)). This is the multiple comparisons problem (Bretz, Hothorn, and Westfall 2010), arising when the combined alternative hypothesis is in disjunctive form. We explain more details in Section 7.

## 4 Model

In this section, we introduce a Kripke model for describing statistical properties and formally define hypothesis tests.

### 4.1 Variables, Data, and Actions

We introduce a set \( \text{Var} \) of variables comprised of two disjoint sets of invisible variables and of observable variables: \( \text{Var} = \text{Var}_{\text{inv}} \cup \text{Var}_{\text{obs}} \). We can directly observe the values of the latter, but not those of the former. We use \( x \) as an invisible variable denoting a population, and \( y \) as an observable variable denoting a dataset sampled from the population.

We write \( \mathcal{O} \) for the set of all data values we deal with, including the Boolean values, integers, real numbers, and lists of data. A dataset is a list of lists of data. In particular, we deal with a list of real vectors as a dataset. Then the vectors range over \( \mathcal{X} = \mathbb{R}^l \) for an \( l \in \mathbb{N} \), a distribution over the population has type \( \mathbb{D}_\mathcal{X} \), and a dataset has type list \( \mathcal{X} \).

We write \( d \sim D^n \) for the sampling of a set \( d \) of \( n \) data from a population \( D \) where all these data are independent and identically distributed (i.i.d.). Let \( \text{Smpl} \) be a set of i.i.d. samplings of datasets from populations (e.g., \( d \sim D^n \)), and \( \text{Cmd} \) be a set of program commands\(^1\) (e.g., \( v := e \) and skip). Then we define an action as a sampling of a dataset or a program command; i.e., \( \text{Act} = \text{Smpl} \cup \text{Cmd} \).

### 4.2 States and Possible Worlds

A state is a pair \((m; a)\) consisting of the current assignment \( m : \mathcal{V} \rightarrow \mathcal{O} \) of data values to variables, and the action \( a \in \text{Act} \) that has been executed in the last transition.

A possible world \( w \) is a sequence of the states \( (w[0], w[1], \ldots, w[k-1]) \) where \( w[i] \) represents the \( i \)-th state. The length \( k \) is denoted by \( \text{len}(w) \). \( w[0] \) and \( w[k-1] \) are called the initial state and the current state, respectively. Since a possible world records all updates of data values, it can be used to model the updates of knowledge and beliefs as with previous works on epistemic logic (Fagin et al. 1995a).

The observation of a state \( w[i] = (m,a) \) is defined by \( \text{obs}(w[i]) = (m,a) \) with an assignment \( m \cdot : \text{Var}_{\text{obs}} \rightarrow \mathcal{O} \) such that \( m(v) = m(v) \) for all \( v \in \text{Var}_{\text{obs}} \). The observation of a possible world \( w \) is defined by \( \text{obs}(w) = \{ \text{obs}(w[0]), \ldots, \text{obs}(w[k-1]) \} \). We abuse notations and denote by \( w : \mathcal{V} \rightarrow \mathcal{O} \) the assignment of data values to variables in the current state of a possible world \( w \).

### 4.3 Kripke Model

We introduce a Kripke model with labeled transitions where two kinds of relations \( \xRightarrow{a} \) and \( \xRightarrow{R} \) may relate possible worlds.

A transition relation \( w \xRightarrow{a} w' \) represents a transition from a world \( w \) to another world \( w' \) by performing an action \( a \). An observability relation \( w\xRightarrow{R}w' \) represents that two possible worlds \( w \) and \( w' \) have the same observation, i.e., \( \text{obs}(w) = \text{obs}(w') \). In Section 5, this relation is used to model the knowledge in the conventional Hintikka-style.

**Definition 1** (Kripke model). We define a Kripke model as a tuple \( \mathcal{M} = (\mathcal{V}, (\xRightarrow{a})_{a \in \text{Act}}, \xRightarrow{R}, (\text{Var})_{w \in \mathcal{W}}) \) consisting of:

\(^1\)In Section 6, we instantiate \( \text{Cmd} \) with a concrete example of commands used in a simple programming language.
• a non-empty set \( W \) of possible worlds;
• for each \( a \in \text{Act} \), a transition relation \( \delta \subseteq W \times W \);
• an observability relation \( R = \{ (w, w') \in W \times W \mid \text{obs}(w) = \text{obs}(w') \} \);
• for each \( w \in W \), a valuation \( V_w : \text{Pred} \rightarrow \mathcal{P}(\mathbb{O}^k) \) that maps a \( k \)-ary predicate to a set of \( k \)-tuples of data;
We assume \( w \cdot \forall y \cdot w' \) implies \( w'[\text{len}(w') - 1] = (m, a) \) for some \( m \). We also assume that each world in a model has the same sets \( \varphi_{\text{inf}} \) and \( \varphi_{\text{obs}} \) of variables.

4.4 Formulation of Hypothesis Testing

Next, we formalize hypothesis tests. We consider a basic test type \( s \in \{ L, U, T \} \) each representing a lower-tailed, upper-tailed, and two-tailed test. A hypothesis test is a tuple
\[
A(s) = (\varphi_0, t, D_{t,\varphi_0}, \varepsilon(s), P(\xi, \vartheta))\text{ consisting of:}
\]
- \( \varphi_0 \) is an assertion, called a null hypothesis;
- \( t \) is a function that maps a dataset \( d \in \mathcal{X} \) to its test statistic \( t(d) \), usually with range \( t = \mathbb{R}^k \) for \( k \geq 1 \);
- \( D_{t,\varphi_0} \in \mathbb{D}(\text{range}(t)) \) is a probability distribution of the test statistic when the null hypothesis \( \varphi_0 \) is true;
- \( \varepsilon(s) \subseteq \text{range}(t) \times \text{range}(t) \) is a likelihood relation where for a test type \( s \) and values \( d \) and \( d' \) of the test statistic, \( d \varepsilon(s) d' \) represents that \( d \) is at most as likely as \( d' \). For brevity, we often omit \( s \) to write \( \varepsilon \) and \( \varepsilon(s) \) and \( \varepsilon \).
- \( P(\xi, \vartheta) \) is a statistical model with unknown parameters \( \xi \) and known parameters \( \vartheta \) that characterizes the population.

Then we define a hypothesis testing over a set \( \Phi \) of possible hypotheses by \( A(s) = (A^s)_{\varphi \in \Phi} \). We denote by \( A \) a finite set of hypothesis testing we consider. For brevity, we sometimes omit \( s \) to write \( A^s \) and \( A \). We also often omit the statistical model \( P(\xi, \vartheta) \) from the description of \( A^s \).

Example 3 (The likelihood relation for Z-test). The two-tailed \( Z \)-test for two populations (Example 1) is denoted by \( A^{(s)}_{00} = (\varphi_0, t, N(0, 1), \varepsilon(T), N(p_{\text{pp}}, \sigma^2) \times N(p_{\text{pp}}, \sigma^2)) \). The likelihood relation \( d \varepsilon(T) d' \) expresses \( |d| \geq |d'| \). When the null hypothesis \( \varphi_0 \) is true, the test statistic \( t(y_1, y_2) \) follows the standard normal distribution \( N(0, 1) \), hence
\[
\Pr[t(y_1, y_2) \varepsilon(T) 1.96] = \Pr[|t(y_1, y_2)| \geq 1.96] = 0.05.
\]
In contrast, for the upper-tailed (lower-tailed) test, with alternative hypothesis \( \varphi_0 \subseteq (p_{\text{ppt}} > p_{\text{pp}}, \sigma^2 < \sigma^2) \), the likelihood relation \( d \varepsilon(U) d' \) (resp. \( d \varepsilon(L) d' \)) is defined by \( d \geq d' \) (resp. \( d \leq d' \)).

Next we define the disjoint combination of two hypothesis tests \( A_{\varphi_0} = (\varphi_0, t, D_{t,\varphi_0}, \varepsilon_1(t), P_1) \) for \( b = 1, 2 \) by \( A_{\varphi_1 \lor \varphi_2} = (\varphi_1 \lor \varphi_2, t, D_{t,\varphi_1 \lor \varphi_2}, \varepsilon(t), P_1 \cdot P_2) \) where \( t(y_1, y_2) = (t_1(y_1), t_2(y_2)) \) and \( D_{t,\varphi_1 \lor \varphi_2} = D_{t,\varphi_1} \times D_{t,\varphi_2} \).

Next we define the statistical belief modality \( K^\varphi_{\text{y,Ag}} \) such that for a formula \( \varphi \) representing a hypothesis,
\[
K^\varphi_{\text{y,Ag}} \varphi := K(\varphi \lor \varphi_{\text{obs}}(y))
\]
where \( A \) is a test with a null hypothesis \( \neg \varphi \). Then we define the statistical possibility \( P^\varphi_{\text{y,Ag}} \) by \( P^\varphi_{\text{y,Ag}} \varphi := K^\varphi_{\text{y,Ag}} \neg \varphi \). For brevity, we write \( K^\varphi_{\text{y,Ag}} \) instead of \( K^\varphi_{\text{y,Ag}} \).

In this section, we define semantics for the assertion logic.

5 Assertion Language

Next we define an assertion logic that can express epistemic properties including knowledge and statistical beliefs.

5.1 Syntax of the Assertion Logic

We introduce two kinds of epistemic modality \( K \) and \( K^\varphi_{\text{y,Ag}} \). Intuitively, a knowledge \( K^\varphi \) expresses that we know \( \varphi \), and this has been studied in a lot of previous work on epistemic logic. In contrast, a statistical belief \( K^\varphi_{\text{y,Ag}} \) expresses that we believe a hypothesis \( \varphi \) based on a statistical test \( A \) on an observed dataset \( y \) with a certain error level (p-value) at most \( \epsilon \). We formalize this as the knowledge that either the hypothesis \( \varphi \) holds or the observed dataset \( y \) is unlikely far from the population (from which \( y \) is sampled).

Formally, for a set \( \text{Var} \) of variables and a set \( \text{Pred} \) of predicates, the set \( \text{Fml} \) of formulas are defined by:
\[
\varphi ::= \eta(x_1, \ldots, x_n) \land \neg \varphi \lor \varphi | K\varphi
\]
where \( \eta \in \text{Pred} \) and \( x_1, \ldots, x_n \in \text{Var} \). The formulas have no quantifiers over variables. We denote the set of all observable (resp. invisible) variables occurring in a formula \( \varphi \) by \( \text{fv}^o(\varphi) \) (resp. \( \text{fv}^i(\varphi) \)). Let \( \text{fv}(\varphi) = \text{fv}^o(\varphi) \cup \text{fv}^i(\varphi) \).

As syntax sugar, we use conjunction \( \land \), implication \( \rightarrow \), and epistemic possibility \( P \), defined as usual by:
\[
\varphi \land \varphi_1 := \neg(\neg \varphi_0 \land \neg \varphi_1), \varphi_0 \rightarrow \varphi_1 := \neg \varphi_0 \land \varphi_1, \text{ and } \text{P} \varphi := \neg \text{K} \neg \varphi.
\]
We introduce three predicates for statistical inference:

\( x \approx P \) represents that a population \( x \) follows a probability distribution that we assume in a statistical model \( P \).

\( x = (x_1, \ldots, x_k) \) and \( n = (n_1, \ldots, n_k) \), \( y \in \mathcal{X} \) represents that for each \( i = 1, \ldots, k \), a dataset \( y_i \) is obtained by sampling \( n_i \) data from the population \( x_i \).

\( \varphi \in \{ =, \leq, \geq, <, > \} \) and \( \epsilon \in [0, 1] \), \( \text{r}_{\text{tA}}(\cdot) \) represents that the observation of a dataset \( y \) is unlikely to occur (with exception \( \approx \epsilon \)) according to a hypothesis test \( A \), and the dataset \( y \) is used in no other hypothesis tests.

We introduce a set \( \text{Pred}_c \) of global predicate, whose interpretations are identical in every possible worlds. We assume \( c \in \text{Pred}_d \) and \( \text{Pred} = \text{Pred}_c \cup \{ c \} \).

As syntax sugar, we introduce the statistical belief modality \( K^\varphi_{\text{y,Ag}} \) such that for a formula \( \varphi \) representing a hypothesis,
\[
K^\varphi_{\text{y,Ag}} \varphi := K(\varphi \lor \varphi_{\text{obs}}(y))
\]
where \( A \) is a test with a null hypothesis \( \neg \varphi \). Then we define the statistical possibility \( P^\varphi_{\text{y,Ag}} \) by \( P^\varphi_{\text{y,Ag}} \varphi := \neg K^\varphi_{\text{y,Ag}} \neg \varphi \). For brevity, we write \( K^\varphi_{\text{y,Ag}} \) instead of \( K^\varphi_{\text{y,Ag}} \).

For a finite set \( \text{A} \) of hypothesis tests, we write \( K^\varphi_{\text{y,Ag}} \) by \( K^\varphi_{\text{y,Ag}} \).

In this section, we define semantics for the assertion logic.
We define the interpretation of formulas in a world $w$ in a Kripke model $\mathcal{M}$ (Definition 1) by:

- $\mathcal{M}, w \models \eta(x_1, \ldots, x_k)$ if and only if $(w(x_1), \ldots, w(x_k)) \in V_w(\eta)$
- $\mathcal{M}, w \models \neg \varphi$ if and only if $\mathcal{M}, w \not\models \varphi$
- $\mathcal{M}, w \models \varphi \lor \varphi'$ if and only if either $\mathcal{M}, w \models \varphi$ or $\mathcal{M}, w \models \varphi'$
- $\mathcal{M}, w \models K \varphi$ if and only if for all $w' \in \mathcal{W}$, $(w, w') \in R$ implies $\mathcal{M}, w' \models \varphi$.

$\mathcal{M}$ is sometimes omitted when it is clear from the context.

Next we define the interpretation of predicates. Each global predicate has the same interpretation in all worlds; i.e., for any $\eta \in \text{Pred}_G$ and $w, w' \in \mathcal{W}$, $V_w(\eta) = V_{w'}(\eta)$.

Let $A_\varphi = (\varphi, t, D_{t, \varphi}, \varepsilon^{(s)})$ be a hypothesis test. Recall that the distribution over the population has type $\mathcal{D}\mathcal{X}$, and that $\varepsilon^{(s)}$ is the likelihood relation (Section 4.4). In a world $w$, we interpret the predicates by:

- $V_w(\sim) = \{(X, D) \in \mathcal{D}\mathcal{X} \times \mathcal{D}\mathcal{X} \mid X = D\}$
- $V_w(\sim) = \{(d, D_n) \in \text{list}(\mathcal{X}) \times \mathcal{D}\mathcal{X} \times \mathcal{X} \mid \text{there is an } i \in \mathbb{N} \text{ such that } |w_0^i(w)| = |w_1^i(x)|+1\}$
- $V_w(\tau_{\mathcal{A}_\varphi}^{(s)}) = \{o \in \mathcal{O} \mid \text{Pr}_{d \sim D_{t, \varphi}}[d \varepsilon^{(s)}(t)(o)] < \epsilon \text{ and } w \text{ has no transition where } o \text{ is used} \}$,

Intuitively, $V_w(\tau_{\mathcal{A}_\varphi}^{(s)})$ is the set of all transitions that reject a null hypothesis $\varphi$. More specifically, the $p$-value $\text{Pr}_{d \sim D_{t, \varphi}}[d \varepsilon^{(s)}(t)(o)]$ is the probability that a data $d$ is at most as likely as the test statistic $t(o)$ when it is sampled from the distribution $D_{t, \varphi}$ in the world $w$; e.g., $\text{Pr}_{d \sim \mathcal{N}(0,1)}[d \varepsilon^{(T)}] = \text{Pr}_{d \sim \mathcal{N}(0,1)}[|d| \geq 1.96] = 0.05$. By definition, we have:

- $\mathcal{M}, w \models \tau_{\mathcal{A}_\varphi}^{(s)}(y)$ if and only if $\text{Pr}_{d \sim D_{t, \varphi}}[d \varepsilon^{(s)}(t(w(y))] < \epsilon \text{ and } w(y) \text{ is not used in other tests than } \mathcal{A}_\varphi$.

This represents that in the possible world $w$, the observation of a dataset $y$ is unlikely to occur (except with probability $\epsilon$) according to the hypothesis test $\mathcal{A}_\varphi$ where the test statistic follows the distribution $D_{t, \varphi}$ in the world $w$.

Then the statistical belief modality $K_{y, A}^{\varphi}$ is interpreted as:

- $\mathcal{M}, w \models K_{y, A}^{\varphi}$ if and only if $\mathcal{M}, w \models (\varphi \lor \tau_{\mathcal{A}_\varphi}^{(s)}(y))$
- $\mathcal{M}, w \not\models \neg \varphi \rightarrow \tau_{\mathcal{A}_\varphi}^{(s)}(y)$.

Intuitively, $K_{y, A}^{\varphi}$ represents a belief that an alternative hypothesis $\varphi$ on the population is true. More specifically, $w' \vdash \neg \varphi \rightarrow \tau_{\mathcal{A}_\varphi}^{(s)}(y)$ means that if we consider a possible world $w'$ where the null hypothesis $\varphi$ is false, then a hypothesis test $\mathcal{A}_\varphi$ would conclude that the observation of a dataset $y$ is unlikely to occur (with exceptions at most $\epsilon$), i.e., $\tau_{\mathcal{A}_\varphi}^{(s)}(y)$ holds in $w'$. We discuss the implication of our formalization of $K_{y, A}^{\varphi}$ in Section 5.3.

Although the modality $K$ expresses the knowledge in terms of S5, the syntax sugar $K_{y, A}^{\varphi}$ represents belief instead of knowledge. This is because $\varphi$ can be false when

$\tau_{\mathcal{A}_\varphi}^{(s)}(y)$ holds (i.e., we may have a false belief on $\varphi$ when the sampled dataset $y$ is unlucky far from the population).

**Example 4 (Statistical belief in Z-tests).** Recall again the two-tailed $Z$-test for two population means in Example 1. The alternative hypothesis is $\varphi_1 \equiv (\mu_{\text{pop1}} \neq \mu_{\text{pop2}})$, and the null hypothesis is $\varphi_0 \equiv (\mu_{\text{pop1}} = \mu_{\text{pop2}})$. We denote this $Z$-test by $A_{\varphi_0, \varphi_1} = (\varphi_0, \varphi_1, N(0, 1), \varepsilon^{(Z)})$.

Suppose that in a world $w$, we sample two datasets $w(y_1)$ and $w(y_2)$ respectively from two populations $w(x_1)$ and $w(x_2)$, and calculate the $Z$-test statistic $t(w(y_1), w(y_2))$ defined in Example 1. When the null hypothesis $\varphi_0$ is true, $t(w(y_1), w(y_2))$ follows the distribution $N(0, 1)$.

If $t(w(y_1), w(y_2)) = 3$, then $\text{Pr}_{d \sim \mathcal{N}(0,1)}[|d| \varepsilon^{(T)}] < 0.05$. Then the null hypothesis $\varphi_0$ is rejected, and we obtain the statistical belief that the alternative hypothesis $\varphi_1$ is true with the significance level 0.05, i.e., $w \models K_{y, A}^{\varphi_1}$. In contrast, if $t(w(y_1), w(y_2)) = 1.8$, then $w \not\models K_{y, A}^{\varphi_1}$ because $\text{Pr}_{d \sim \mathcal{N}(0,1)}[|d| \varepsilon^{(T)}] t(w(y_1), w(y_2))] > 0.05$.

### 5.3 Remarks on the Formalization

**Implication** The universe $\mathcal{W}$ of the model $\mathcal{M}$ is assumed to include all possible worlds we can imagine. If there exists no possible world satisfying the null hypothesis $\neg \varphi$ in the model $\mathcal{M}$, then $\varphi$ is satisfied in all worlds in $\mathcal{M}$, hence so are $K\varphi$ and $K_{y, A}^{\varphi}$. This reflects the fact that if we cannot imagine a possible world where $\neg \varphi$ is true, then we already know that $\varphi$ is true without executing hypothesis tests.

**Type II error** The type II error rate is the probability that the hypothesis test $A$ does not reject the null hypothesis $\neg \varphi$ when $\varphi$ is false. Assume that the true population satisfies a hypothesis $\xi$ in the world $w$. Let $y'$ be a dataset such that the $p$-value (type I error rate) $\alpha$ of the test $A$ is 0.05; i.e., $w \not\models K_{y, A}^{\varphi_1}$. Then the type II error rate $\beta$ when $\alpha = 0.05$ is given by $w \models K_{y, A}^{\beta} \neg \xi$.

### 5.4 Properties of Statistical Beliefs

The statistical possibility $P_{y, A}^{\varphi}$ means that we think a null hypothesis $\varphi$ may be true after a hypothesis test $A$ did not reject $\varphi$ with a significance level $\epsilon$. Formally, we have:

- $\mathcal{M}, w \models P_{y, A}^{\varphi}$ if and only if there is a $w'$ s.t. $(w, w') \in R$ and $\mathcal{M}, w' \not\models \neg \varphi \lor \tau_{\mathcal{A}_\varphi}^{(s)}(y)$.
- $\mathcal{M}, w \models P_{y, A}^{\varphi}$ if and only if $\mathcal{M}, w \models P(\varphi \land \neg \tau_{\mathcal{A}_\varphi}^{(s)}(y))$.

Now we show basic properties of statistical beliefs.

**Proposition 1 (Properties of $K_{y, A}^{\varphi}$).** Let $\varphi$ be a formula. $A$ be a hypothesis test. $y \in \text{Var}_{\text{obs}}$, and $\epsilon \in \mathbb{R}_{\geq 0}$.

1. Knowledge is also regarded as belief: $\models K \varphi \rightarrow K_{y, A}^{\varphi}$.
2. If we believe $\varphi$ based on a test $A$, then we know this statistical belief: i.e., $\models K_{y, A}^{\varphi} \rightarrow K_{y, A}^{\varphi}$.
3. If we failed to reject $\varphi$ and think it is possible, then we know this possibility: i.e., $\models P_{y, A}^{\varphi} \rightarrow K_{y, A}^{\varphi}$.
4. If $\epsilon \leq \epsilon'$, $\models K_{y, A}^{\varphi} \rightarrow K_{y, A}^{\varphi}$ and $\models P_{y, A}^{\varphi} \rightarrow P_{y, A}^{\varphi}$.

---

2We also define the interpretation of $\tau_{\mathcal{A}_\varphi}^{(s)}(y)$ analogously.
5. $K_\varphi^< e_{A,A}$ may represent a false belief. The alternative hypothesis $\varphi$ we believe may be false, i.e., the rejected null hypothesis $\neg \varphi$ may be true: $e > 0 \iff K_\gamma^< y_{A} \varphi \rightarrow \varphi$.

6. $\models K_\gamma^< y_{A} \varphi \rightarrow K_\gamma^< y_{A} \varphi$.

The proofs are straightforward from definitions. See the full version of this paper for the proofs.

## 6 A Simple Programming Language

We introduce an imperative programming language Prog.

### 6.1 Syntax of Prog

We define the syntax of Prog by the following BNF:

$$
T := \text{bool} | \text{int} | \text{real} | T \times T | \text{list}(T) \quad \text{(Types)}
$$

$$
e := v \mid f(e_1, \ldots, e_k) \mid \text{skip} \mid v := e \quad \text{(Terms)}
$$

$$
c := \text{if } e \text{ then } C \text{ else } C \mid \text{loop } e \text{ do } C \quad \text{(Commands)}
$$

$$
C := e \mid C; C \mid C | C \mid \text{loop } e \text{ do } C \quad \text{(Programs)}
$$

where $v$ is an element of Var$_\text{obs}$. $f$ is a (built-in) function symbol, and constants are dealt as functions with arity 0. Notice that a program can handle only observable variables.

$T$ represents types. A type is either bool for Boolean values, int for integers, real for real numbers, $T_1 \times T_2$ for pairs consisting of a value of type $T_1$ and a value of type $T_2$, or list$(T)$ for lists of values of type $T$. $e$ represents expressions that evaluate to values. An expression is either a variable $v$ or a function call $f(e_1, \ldots, e_k)$: the latter is typically a call to a function that computes a test statistic. $c$ and $C$ represent commands and programs respectively. We give their intuitive explanation as follows.

- skip does nothing.
- $v := e$ updates $v$ with the result of an evaluation of $e$.
- $C_1; C_2$ executes $C_1$ and then $C_2$.
- $C_1 | C_2$ executes $C_1$ and $C_2$ in parallel that may share some data.
- if $e$ then $C_1$ else $C_2$ executes $C_1$ if $e$ evaluates to true; executes $C_2$ if $e$ evaluates to false.
- loop $e$ do $C$ iteratively executes $C$ as long as $e$ evaluates to true.

For instance, the example program in Section 7.5 conforms to the programming language Prog.

Hereafter we assume that all programs are well-typed although we do not explicitly mention the types. Checking this condition for our language can be done by adapting a standard type-checking algorithm to our setting.

We write upd$(C)$ for the set of variables that may be updated by executing $C$: upd$(\text{skip}) = \emptyset$, upd$(v := e) = \{v\}$, upd$(C_1; C_2) = \text{upd}(C_1) \cup \text{upd}(C_2)$, and $\text{upd}(\text{loop } e \text{ do } C) = \text{upd}(C)$.

Then we impose the following restriction to every occurrence of $C_1 | C_2$: $\text{upd}(C_1) \cap \text{Var}(C_2) = \emptyset$. This restriction is to ensure that an execution of $C_1$ does not interfere with that of $C_2$.

### 6.2 Semantics of Prog

We define the semantics of Prog over a Kripke model $\mathcal{W}$ with labeled transitions given in Section 4.3. The semantics is based on the standard structural operational semantics (e.g. (Nielsen and Nielson 2007)).

For a possible world $w \in \mathcal{W}$ and $n = \text{len}(w)$, we write $w = w[0], w[1], \ldots, w[n - 2], (m, a)$ where $(m, a)$ is the current state $w[n - 1]$ with an assignment $m: \text{Var} \rightarrow \mathcal{O}$ and an action $a$ in the last transition in $\mathcal{W}$.

For the assignment $m$ of the current state of $w$, we define the evaluation $[e]_m$ of a term $e$ inductively by $[v]_m = m(v)$ and $[f(e_1, \ldots, e_k)] = [f([e_1]_m, \ldots, [e_k]_m)]$.

As in Figure 1, we define a binary relation

$$
\rightarrow \subseteq (\text{Prog} \times \mathcal{W}) \times ((\text{Prog} \times \mathcal{W}) \cup \mathcal{W})
$$

that relates a pair $(C, w)$ consisting of a program $C$ and a possible world $w$ to its next step of execution. If $C$ is terminated, the next step will be a possible world $w'$, otherwise the execution continues to the $(C', w')$.

\[\begin{align*}
\langle \text{skip}, w \rangle & \rightarrow w; (m, \text{skip}), \\
\langle v := e, w \rangle & \rightarrow w; (m[v \mapsto [e]_m], \nu := e), \\
\langle C_1, w \rangle & \rightarrow w' \\
\langle C_1; C_2, w \rangle & \rightarrow \langle C_2, w' \rangle \\
\langle C_1, w \rangle & \rightarrow \langle C'_1, w' \rangle \\
\langle C_1 | C_2, w \rangle & \rightarrow \langle C'_1 | C'_2, w' \rangle \\
\langle \text{if } e \text{ then } C_1 \text{ else } C_2, w \rangle & \rightarrow \begin{cases} \\
\langle C_1, w \rangle & \text{if } e \quad \langle C_2, w \rangle & \text{else } C_2_w \\
\end{cases} \\
\langle \text{loop } e \text{ do } C, w \rangle & \rightarrow \begin{cases} \\
\langle C; \text{loop } e \text{ do } C, w \rangle & \text{if } e \quad \langle C', w \rangle & \text{else } C'_2_w \\
\end{cases} \\
\langle C_1, w \rangle & \rightarrow \langle C'_1, w' \rangle \\
\langle C_1 | C_2, w \rangle & \rightarrow \langle C'_1 | C'_2, w' \rangle \\
\langle C_1, w \rangle & \rightarrow w' \\
\langle C_1 | C_2, w \rangle & \rightarrow \langle C'_1 | C'_2, w' \rangle \\
\langle C_2, w \rangle & \rightarrow w' \\
\langle C_1 | C_2, w \rangle & \rightarrow \langle C'_1 | C'_2, w' \rangle
\end{align*}\]

Figure 1: Rules of execution of programs.

Remark that the semantics of a program contains the trace of commands executed in it. Hence, even if programs finally have the same result, their semantics may be different.
Remark on Parallel Compositions Since parallel compositions are nondeterministic, \( w' \in \mathcal{C}(w) \) may not be unique. However, the resulting world \( w' \) are essentially the same, because for parallel composition \( C_1 || C_2 \), the world \( w' \in \mathcal{C}(C_1 || C_2) \) is convertible to a pair of \( w_1 \in \mathcal{C}(C_1) \) and \( w_2 \in \mathcal{C}(C_2) \) and vice versa.

If we have \( C_b, w \rightarrow^* w; u_b \) for \( b = 1, 2 \), then by \( \text{upd}(C_1) \cap \text{Var}(C_{3-b}) = \emptyset \), we obtain a sequence \( u' \) such that \( C_1 || C_2 \rightarrow^* w; u' \) by combining \( u_1 \) and \( u_2 \).

Conversely, if \( C_1 || C_2 \rightarrow^* w; u \), we can decompose \( u' \) into \( u_1 \) and \( u_2 \) such that \( C_b, w \rightarrow^* w; u_b \) for \( b = 1, 2 \) (for detail, see the full version of this paper).

Then, for any pair of \( \forall \xi \)-free assertions \( \Phi_1 \) and \( \Phi_2 \) satisfying \( \text{upd}(C_b) \cap \text{fe}(\Phi_{3-b}) = \emptyset \) for \( b = 1, 2 \), for any possible world \( w' \in \mathcal{C}(C_1 || C_2)(w) \), we can write \( w' = w; u' \), and decompose \( u' \) into \( u_1 \) and \( u_2 \) as above. We then obtain \( w' \models \Phi_1 \land \Phi_2 \) iff \( w; u_1 \models \Phi_1 \) and \( w; u_2 \models \Phi_2 \).

Procedures of Hypothesis Testing Finally, we present the interpretation of a program \( f_{A}(\psi)(y) \) for a hypothesis test \( A \equiv \{ \phi \} C \{ \psi \} \) with a null hypothesis \( \phi \), a test statistic \( t \), and a test size \( s \). For a dataset \( y \) and an assignment \( m \), \( [f_{A}(\psi)(y)]_m \) represents the p-value:

\[
[f_{A}(\psi)(y)]_m = \Pr_{d \sim D_{\psi}(\phi)} [d \mathcal{L}(t(m(y))],
\]

which is the probability that a data \( d \) is at most as likely as the test statistic \( t(m(y)) \) when it is sampled from \( D_{\psi}(\phi) \) in the world where the null hypothesis \( \phi \) is true.

7 Belief Hoare Logic for Hypothesis Testing

We introduce belief Hoare logic (BHL) for formalizing and reasoning about statistical inference using hypothesis tests. Then we describe the reasoning about the multiple comparison problem and p-value hacking using BHL.

7.1 Hoare Triples

An environment is defined as a pair \( \Gamma = (\Gamma^{\text{inv}}, \Gamma^{\text{obs}}) \) consisting of an invisible environment \( \Gamma^{\text{inv}} \) and an observable environment \( \Gamma^{\text{obs}} \) that assign types to invisible variables and to observable variables, respectively. We denote by \( \text{Var}(\Gamma) \) the set of all variables occurring in an environment \( \Gamma \), and by \( \text{Env} \) the set of all possible environments.

A judgment is of the form \( \Gamma \vdash \{ \psi \} C \{ \phi \} \) where \( \Gamma \in \text{Env}, \psi, \phi \in \text{Fml} \), and \( C \in \text{Prog} \). Intuitively, this represents that when the precondition \( \psi \) is satisfied, executing the program \( C \) results in satisfying the postcondition \( \phi \).

We say that a judgment \( \Gamma \vdash \{ \psi \} C \{ \phi \} \) is valid iff for any model \( M \) and any possible world \( w \), if \( M, w \models \psi \), then \( M, w' \models \phi \) for all \( w' \in \mathcal{C}(w) \).

We write \( \Gamma \models \phi \) if \( M, w \models \phi \) for any model \( M \) and any world \( w \) respects the type information in \( \Gamma \) (i.e., the type of \( w(v) \) being \( \Gamma(v) \) for any variable \( v \in \text{Var} \)).

7.2 Inference Rules

Next, we present the inference rules for belief Hoare logic (BHL). The rules are classified into those for basic command constructs (Figure 2) and for hypothesis testing constructs (Figure 3). The latter includes axiom schemas that can be instantiated to a variety of concrete hypothesis test methods; we present such instantiation in Section 7.3.

The rules for basic constructs in Figure 2 are standard; the readers are referred to a standard textbook on Hoare logic (Winskel 1993) for details. We remark the features:

- In the rules \text{IF} and \text{LOOP}, the guard condition \( e \) is a Boolean expression implicitly used as a logical predicate in the preconditions and the postconditions as usual.
- The rule \text{CONSEQ} is used to weaken the precondition and strengthen the postcondition of a triple. The relation \( \models \phi \) defined above is used in this rule.

Schemas for Single Hypothesis Testing In Figure 3 the axiom schemas \( \text{Two}-T, \text{Low}-T, \) and \( \text{Up}-T \) correspond to two-tailed, lower-tailed, and upper-tailed tests, respectively.

In these schemas, a dataset \( y \) is sampled from a population \( x \), which follows a statistical model \( P(\xi; \theta) \) with unknown true parameters \( \xi \in \Phi \) and known parameters \( \theta \in \Theta \).

To reason about the unknown parameter \( \xi \), we perform a hypothesis test \( A_{x \xi} \) with a null hypothesis \( \phi_0 \). Let \( \phi \), \( \phi_U \), and \( \phi_U$\}$ be the alternative hypotheses for the lower-tailed, upper-tailed, and two-tailed tests, respectively.

When we consider both the lower-tail \( \phi_L \) and upper-tail \( \phi_U \), possible before performing the test, we express them by \( P(\phi_L) \) and by \( P(\phi_U) \) in the precondition. Then we apply the schema \text{Two}-T. When the two-tailed test \( f_{A_{x \xi}}(y) \) returns the p-value \( \alpha \in [0, 1] \), we obtain a statistical belief on the alternative hypothesis \( \phi_T \), namely, \( K(\phi_T \mid \phi_U) \).

When we consider only the lower-tail \( \phi_L \) possible, we apply \text{Low}-T (resp. \text{Up}-T) and obtain a statistical belief on \( \phi_L \) (resp. \( \phi_U \), namely, \( K(\phi_L \mid \phi_U) \)).
Figure 3: Axiom schemas and rules for hypothesis tests. TWO-T, LOW-T, and UP-T are schemas for two-tailed, lower-tailed, and upper-tailed tests, respectively. MULT-V is the rule for the Bonferroni’s method, and MULT-Λ is for the simultaneous tests without correction of α.

7.3 Instantiation to Concrete Test Methods

The axiom schemas for hypothesis tests in Figure 3 are instantiated with concrete examples of tests as follows. We first show the case of the two-tailed Z-test (Example 1).

Example 5 (Z-test). The axiom for the Z-test comparing means of two populations with datasets \(y_1, y_2\) of sample sizes \(n_1, n_2\) and a known variance \(\sigma^2\) is given by instantiating TWO-T with the following parameters:

\[
P_b(\mu_{bpl}, \sigma^2) \overset{def}{=} N(\mu_{bpl}, \sigma^2) \quad \text{for } b = 1, 2
\]

\[
\varphi_0 = (\mu_{\text{bpl}1} = \mu_{\text{bpl}2}) \quad \varphi_T = (\mu_{\text{bpl}1} \neq \mu_{\text{bpl}2})
\]

\[
\xi^{(T)} = \{ (r_1, r_2) \in \mathbb{R} \times \mathbb{R} \mid |r_1| > |r_2| \}
\]

\[
t(y_1, y_2) = \frac{\text{mean}(y_1) - \text{mean}(y_2)}{\sigma \sqrt{1/\text{size}(y_1) + 1/\text{size}(y_2)}}
\]

\[
D_{b, \varphi_0} = N(0, 1) \quad (\text{the standard normal distribution}).
\]

Next we show the instantiation to the classical likelihood ratio test with a simple null hypothesis \(\xi = \xi_0\) and a simple alternative hypothesis \(\xi = \xi_1\), namely, in the setting of the Neyman-Pearson lemma.

Example 6 (Likelihood ratio test). The goal of the (simplest version of) the likelihood ratio test is to determine which of two candidate distributions \(D_p, D_q\) is better to fit a dataset \(y = (y_1, \ldots, y_n)\) of sample size \(n\). The test can be reformulated with the statistical model \(x \approx P(\xi)\) defined by

\[
P(\xi) = \begin{cases} 
D_q & \text{if } \xi = \xi_0 \\
D_p & \text{if } \xi = \xi_1 
\end{cases}
\]
and the following null and alternative hypotheses:
\[ \varphi_0 \equiv (\xi = \xi_0), \quad \varphi_L \equiv (\xi = \xi_1). \]
The likelihood function \( L \) for this test is given by
\[ L(y|\xi_0) = \prod_{i=1}^{n} q(y_i), \quad L(y|\xi_1) = \prod_{i=1}^{n} p(y_i). \]
Then, the likelihood ratio \( t(y) \) is given by
\[ t(y) = \frac{L(y|\xi_0)}{L(y|\xi_1)} = \frac{\prod_{i=1}^{n} q(y_i)}{\prod_{i=1}^{n} p(y_i)} \]
where \( p \) and \( q \) are the density functions of \( D_p \) and \( D_q \) respectively.
In the likelihood ratio test, for a given \( \alpha \) and a threshold \( k \) such that \( P_{D_1,\ldots,D_n|D_q}[t((d_1, \ldots, d_n)) \leq k] \leq \alpha \), if we have \( t(y) \leq k \), the likelihood \( L(y|\xi_0) \) is too small to accept the distribution \( D_q \). We then conclude that the other candidate \( D_p \) is better to fit \( y \) (thus this test is lower-tailed).

Conversely, the \( p \)-value of this test is defined by
\[ \Pr_{d_1,\ldots,d_n\sim\sim D_q}[t((d_1, \ldots, d_n)) \leq t(y)]. \] (6)

The axiom for the likelihood ratio test is given by instantiating Low-T with the above model \( P(\xi), \varphi_0, \varphi_L, t(y) \) and
\[ \varphi_0 \equiv (\xi = \xi_0), \quad \varphi_L \equiv (\xi = \xi_1). \]

Bayesian hypothesis test is given in an analogous way.

**Example 7** (Bayesian hypothesis test). Consider the Bayesian likelihood ratio test with a dataset \( y \) of sample size \( n \), prior distributions \( D_{p'}, D_{q'} \in \mathbb{D} \) with density functions \( p' \) and \( q' \), and posterior distributions \( D_{p(z)}, D_{q(z)} \in \mathbb{D} \) with density functions \( p(-z) \) and \( q(-z) \).

The goal of this test is to determine whether the dataset \( y \) is sampled from \( D_{q(z)} \) when \( z \) follows \( D_{q'} \). The null hypothesis is that \( y \) is sampled from \( D_{p(z)} \) when \( z \) follows \( D_{p'} \).

A statistical model \( x \approx P(\xi) \) of this test is defined as follows: First, we introduce the statistical models \( z \approx P_0(\xi) \) and \( x \approx P_1(\xi, z) \) of prior and posterior distributions by
\[ P_0(\xi) = \begin{cases} D_{q'} & \text{if } \xi = \xi_0, \\ D_{p'} & \text{if } \xi = \xi_1. \end{cases} \]
Next, for each \( \xi = \xi_0, \xi_1 \), we define the probability measure \( P(\xi) \in \mathbb{D} \) by for any measurable subset \( A \subseteq \mathbb{R} \),
\[ (P(\xi))(A) = \int_A h_{\xi,A} d\mu_\xi \]
where \( \mu_\xi = P_0(\xi) \) and \( h_{\xi,A} : \mathbb{R} \rightarrow \mathbb{R} \) is a measurable function defined by \( h_{\xi,A}(z) = (P_1(\xi, z))(A) \).

The axiom for this test is given by instantiating Low-T with the above model \( P \) and the following parameters:
\[ \varphi_0 \equiv (\xi = \xi_0), \quad \varphi_L \equiv (\xi = \xi_1). \]

Unlike the (classical) likelihood ratio test, the **Bayes factor** is the ratio of the following marginal likelihoods:
\[ L(y|\xi_0) = \int q(y_i) \prod_{i=1}^{n} p(y_i | z) dz \]
\[ L(y|\xi_1) = \int p(y_i) \prod_{i=1}^{n} p(y_i | z) dz \]

### 7.4 Reasoning About Multiple Comparison

We illustrate how BHL reasons about the multiple comparison in Example 2, where the derivation of (3) guarantees that the hypothesis tests are applied appropriately in \( C_{\text{drug}} \) in (1). (Details are shown in the Supplementary Material.)

In the derivation, we obtain the following judgments:
\[ \Gamma \vdash \{ \psi_{\text{pre}} \} C_{12} \{ \psi_{\text{pre}} \land K_y^{0.12} \varphi_{12} \} \]
\[ \Gamma \vdash \{ \psi_{\text{pre}} \land K_y^{0.12} \varphi_{12} \land \alpha_{12} \leq 0.05 \} C_{13} \{ \varphi_{12}^{\text{bel}} \rightarrow \varphi_{12}^{\text{bel}} \} \]
\[ \Gamma \vdash \{ \psi_{\text{pre}} \land K_y^{0.12} \varphi_{12} \land \alpha_{12} > 0.05 \} \text{ skip } \{ \varphi_{12}^{\text{bel}} \rightarrow \varphi_{12}^{\text{bel}} \} \]

where \( \varphi_{12}^{\text{bel}} \equiv K_y^{0.05} \varphi_{12} \) and \( \varphi_{12}^{\text{bel}} \equiv K_y^{0.12} \varphi_{12} \land \varphi_{13} \). The second judgment is derived by the two-tailed test axiom, **MULT-\land**, and **CONSEQ**. The last judgment is derived from \( \Gamma \models ( \psi_{\text{pre}} \land K_y^{0.12} \varphi_{12} \land \alpha_{12} > 0.05 ) \rightarrow \neg \varphi_{12}^{\text{bel}} \) and \( \Gamma \models \neg \varphi_{12}^{\text{bel}} \rightarrow ( \varphi_{12}^{\text{bel}} \rightarrow \varphi_{12}^{\text{bel}} ) \) and **CONSEQ**. Applying **IF** to the last two judgments, we have \( \Gamma \vdash \{ \psi_{\text{pre}} \land K_y^{0.12} \varphi_{12} \} \) if \( \alpha_{12} \leq 0.05 \) then **skip** \( \{ \varphi_{12}^{\text{bel}} \rightarrow \varphi_{12}^{\text{bel}} \} \); composing it with the first judgment by applying **SEQ**, we have the judgment in (3).

In contrast, the program \( C_{12} \parallel C_{13} \) in (4) shows the multiple comparison problem. Since the alternative hypothesis \( \varphi_{12} \lor \varphi_{13} \) is disjunctive, we apply the rule **MULT-\lor** to drive the belief \( K_y^{0.12+\alpha_{13}} ( \varphi_{12} \lor \varphi_{13} ) \), with a \( p \)-value (larger than \( \alpha_{12} \) and \( \alpha_{13} \)) at most \( \alpha_{12} + \alpha_{13} \).

### 7.5 Reasoning About \( p \)-Value Hacking

We informally describe how our framework can be applied to reason about a program for \( p \)-value hacking, i.e., a scientifically malignant technique to obtain a low \( p \)-value. The following program \( c_{\text{pHack}} \) is an example of \( p \)-value hacking that conducts a hypothesis test on different datasets \( y_1 \) and \( y_2 \), and ignores the experiment showing a higher \( p \)-value to report only a lower \( p \)-value:
\[ (\alpha_1 := f_{\alpha_0}^{\gamma_0}(y_1); \alpha_2 := f_{\alpha_0}^{\gamma_0}(y_2)); \]
\[ \text{if } \alpha_1 < \alpha_2 \text{ then } \alpha := \alpha_1 \text{ else } \alpha := \alpha_2 \]

We write \( \varphi_{\text{alt}} \) for the alternative hypothesis of this test.

For the reported \( p \)-value \( \alpha \) to be an actual \( p \)-value, \( K_y^{\gamma_0}(y_1,y_2) \varphi_{\text{alt}} \) needs to hold as a postcondition of \( c_{\text{pHack}} \). Then
must hold at the end of the first line of c_{p Hack} due to the rules UpDVar and If. However, this is not implied by the post-conditions of Mult-V or Mult-∧. Since K_{(y_1, y_2)}^{\leq a_1} (\varphi_{alt}) is a postcondition of c_{p Hack}, the total p-value \alpha_1 + \alpha_2 should be reported without ignoring any experiments.

8 Discussion

In this section, we provide a whole picture of the justification of statistical belief inside and outside BHL.

A statistical belief derived in a program relies on the following three issues: (i) the validity of hypothesis test methods themselves, (ii) the satisfaction of the empirical conditions required for the hypothesis tests, and (iii) the appropriate usage of hypothesis tests in the program. In our framework, these are respectively addressed by (a) the validity of axioms and rules, (b) the (manual) confirmation of the pre-conditions in a judgment, and (c) the proof for the judgment.

8.1 Validity of Hypothesis Test Methods

The validity of hypothesis test methods is not ensured by mathematics alone. The philosophy of statistics has a long history of argument on the proper interpretation of hypothesis testing. One of the most notable examples is the argument between the frequentist and the Bayesian statistics, which still has many issues to be discussed (Sober 2008).

We also remark that statistical methods occasionally involve some approximation of numerical values. However, we may not always confirm the validity of approximation rigorously, i.e., whether the approximation is valid to the specific situation we apply the statistical methods.

For these reasons, we do not attempt to formalize the “justification” for hypothesis test methods within BHL, and left them for future work. Instead, we define axiom schemas for hypothesis tests that are commonly used in practice and explained in textbooks, e.g., (Kanji 2006). Then we focus on the logical aspects of the appropriate usage of hypothesis tests, which has been a long-standing, practical concern but has not been formalized using a symbolic logic before.

One of the advantages of this approach is that we do not adhere to a specific philosophy of statistics, but can model both the frequentist and the Bayesian statistics by introducing an axioms-rule corresponding to each hypothesis test.

8.2 Clarification of Empirical Conditions

The hypothesis test methods usually assume some empirical conditions on the unknown population from which the dataset is sampled. Typically, many parametric tests require that the population follows a normal distribution. For instance, the Z-test in Example 5 assumes that the population follows a normal distribution with known variance, but this cannot be rigorously confirmed or justified in general.

In some cases, such conditions on the unknown population may be confirmed approximately or partially by some exploratory observations on the sampled data and by prior knowledge of some properties on the population (outside the statistical inference). As far as we know, there has been no general method for justifying such empirical conditions rigorously. Thus, the formal justification of those conditions themselves would require further research in statistics.

In the present paper, the empirical conditions on the unknown population remain to be assumptions from the viewpoint of formal logic. Hence, we describe empirical conditions as the preconditions of a judgment in BHL.

Explicit specification of the preconditions would be useful for non-experts to prevent errors in the choice of statistical methods. Furthermore, when we formalize empirical science in future work, it would be crucial to clarify the empirical conditions that justify scientific conclusions.

8.3 Epistemic Aspects of Statistical Inference

One of our contributions is to show that epistemic logic is useful to formalize statistical inference. Although the outcome of a hypothesis test is a knowledge determined by the test action, it may form a false belief; i.e., a rejected null hypothesis may be true, and a retained one may be false. Hence the formalization of statistical inference deals with both truth and beliefs, for which epistemic logic is suitable.

The key to formalizing statistical beliefs is to introduce a Kripke semantics with a possible world w_0 where a null hypothesis is true (Section 5.2). This possible world w_0 may not be the real world that we actually apply the hypothesis test on an observed dataset.

In the Kripke model, a transition between states is used to model the update of statistical beliefs by a hypothesis test. Since the world records the executions of all tests, BHL does not allow for hiding some tests to manipulate the statistics (e.g., p-value in multiple comparison and p-value hacking).

Furthermore, the choice of two-tailed or one-tailed tests requires describing a prior belief using the possibility modality P. Without this modality, we cannot express the belief that both lower-tail and upper-tail are possible before applying the test, since this belief may not be true.

Without this Kripke semantics, the formalization of hypothesis testing would deal with only the purely mathematical propositions satisfied in the possible world w_0 where a null hypothesis is true, hence could not reason about the appropriate usage of hypothesis tests in the real world.

9 Conclusion

In this work, we proposed a new approach to formalizing and reasoning about statistical inference in programs. Specifically, we introduced belief Hoare logic (BHL) for formalizing and checking the requirement for hypothesis tests to be employed appropriately. Then we showed that BHL is useful to reason about practical issues in statistics. We also discussed a whole picture of the justification of statistical inference. We emphasize that this is the first attempt to introduce a program logic for the appropriate application of hypothesis tests.

In ongoing and future work, we are extending our framework to other kinds of statistical methods. We plan to investigate the relative completeness of BHL and to develop a verification tool based on this framework. Another possible research would be to study justification logic for statistics.
Acknowledgments

The authors are supported by ERATO HASUO Metamathe-matics for Systems Design Project (No. JPMJER1603), JST. In particular, we thank Ichiro Hasuo for providing the opportunity for us to meet and collaborate in that project. Yusuke Kawamoto is supported by JST, PRESTO Grant Number JP-MJPR2022, Japan, and by JSPS KAKENHI Grant Number 21K11028, Japan. Tetsuya Sato is supported by JSPS KAK-ENHI Grant Number 20K19775, Japan. Kohei Suenaga is supported by JST CREST Grant Number JMPCR2012, Japan.

References


