How to Test Hypotheses When Exact Values are Replaced by Intervals to Protect Privacy: Case of t-Tests

Vladik Kreinovich[1] and Christian Servin[2]
[1]Department of Computer Science
University of Texas at El Paso
El Paso, Texas 79968, USA
vladik@utep.edu
[2]Information Technology Department
El Paso Community College
919 Hunter, El Paso, TX 79915, USA
cservin@gmail.com

Abstract

Researchers continuously look for possible relations between relevant quantities, e.g., relations which may help in preventing and curing diseases. Once a hypothesis is made about such a relation, it is necessary to test whether it is confirmed by the data. For such hypothesis testing, t-tests are most widely used. For example, a t-test can check, based on two samples, whether it is possible that they come from distributions with the same mean – e.g., whether the average blood pressure after a proposed treatment is the same as before or it is provably smaller – meaning that the tested treatment works.

In traditional statistics, we assume that we know the exact values of the corresponding quantities. In biomedical research, however, it is important to preserve patients' privacy and confidentiality – and, knowing the exact values of all relevant parameters, one can uniquely identify the patient. One of the most efficient ways to preserve privacy is thus to replace the exact values with intervals containing such values. For example, instead of the exact age – which can uniquely identify the patient – we only store an interval containing this age: between 20 and 30, or between 30 and 40, etc.

Different values from the corresponding intervals lead, in general, to different values of the corresponding statistic. In such situations, to make sure that the data confirms the given hypothesis, we need to check that the corresponding statistic is within the desired interval for all possible

values of the input quantities. In other words, we need to make sure that the whole range of possible values of the corresponding statistic is inside the desired interval. Computing this interval is, in general, NP-hard. In this talk, we provide efficient algorithms for computing t-tests under privacy-motivated interval uncertainty.

1 Introduction

Need for t-tests. Biomedical researchers continuously look for possible relations between relevant quantities. Such relations may help in preventing and curing diseases.

Once a hypothesis is made about such a relation, it is necessary to test whether it is confirmed by the data. For such hypothesis testing, *t-tests* are most widely used. A t-test can check whether two samples $x_1, \ldots, x_{n_x}$ and $y_1, \ldots, y_{n_y}$ come from distributions with the same mean, by comparing the value of an appropriate statistic t with a corresponding threshold; see, e.g., [6].

All versions of the t-test are based on sample means $\overline{X} = \dfrac{1}{n_x} \cdot \sum\limits_{i=1}^{n_x} x_i$ and $\overline{Y} = \dfrac{1}{n_y} \cdot \sum\limits_{i=1}^{n_y} y_i$ and sample variances

$$s_X^2 = \frac{1}{n_x - 1} \cdot \sum_{i=1}^{n_x} (x_i - \overline{X})^2 \text{ and } s_Y^2 = \frac{1}{n_y - 1} \cdot \sum_{i=1}^{n_y} (y_i - \overline{Y})^2:$$

- For testing that the actual mean μ is μ_0, we use

$$t = \frac{\overline{X} - \mu_0}{s_X / \sqrt{n_x}}.$$

- For testing that the means are equal ($\mu_x = \mu_y$) in the case of equal sample sizes $n_x = n_y$ and equal variances, we use

$$t = \frac{\overline{X} - \overline{Y}}{s_{XY} \cdot \sqrt{2/n_x}}, \text{ where } s_{XY} = \sqrt{\frac{1}{2} \cdot (s_X^2 + s_Y^2)}.$$

- In the case of unequal sample sizes $n_x \neq n_y$, but equal variances, we use

$$t = \frac{\overline{X} - \overline{Y}}{s_{XY} \cdot \sqrt{\dfrac{1}{n_x} + \dfrac{1}{n_y}}}, \text{ where } s_{XY} \stackrel{\text{def}}{=} \sqrt{\frac{(n_x - 1) s_X^2 + (n_y - 1) s_Y^2}{n_x + n_y - 2}}.$$

- Finally, in the general case, when variances may be unequal, we use

$$t = \frac{\overline{X} - \overline{Y}}{s_{\overline{X}-\overline{Y}}}, \text{ where } s_{\overline{X}-\overline{Y}} = \sqrt{\frac{s_X^2}{n_x} + \frac{s_Y^2}{n_y}}.$$

We can use t-tests, for example, to check whether the average blood pressure decreases after a proposed treatment.

Need to preserve privacy. In traditional statistics, we assume that we know the exact values of the corresponding quantities. In biomedical research, however, it is important to preserve patients' privacy and confidentiality. Knowing the exact values of age, height, weight, etc., one can uniquely identify the patient. One of the most efficient ways to preserve privacy is thus to replace the exact values with intervals containing such values. For example, instead of the exact age, we only store an interval containing this age:

- between 20 and 30, or
- between 30 and 40, etc.

Resulting computational challenge. We want to estimate the value of a statistic. We know how the statistic depends on the sample values $x_1, \ldots, x_{n_x}$. For example, the mean is $\overline{X} = \frac{1}{n_x} \cdot \sum_{i=1}^{n_x} x_i$.

For the t-test, we estimate a statistic t. The hypothesis is confirmed, with given confidence α, if this value is below a certain threshold t_α: $t \in [0, t_\alpha]$.

For privacy-protected data, instead of the exact values x_i, we only know the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$. Different values $x_i \in \mathbf{x}_i$ lead, in general, to different values of the corresponding statistic s. In particular, for different $x_i \in \mathbf{x}_i$ and $y_i \in \mathbf{y}_i$, we have different values $t(x_1, \ldots, x_{n_x}, y_1, \ldots, y_{n_y})$.

To reliably confirm the hypothesis, we need to check that

$$t(x_1, \ldots, y_1, \ldots) \le t_\alpha$$

for all $x_i \in \mathbf{x}_i$ and $y_i \in \mathbf{y}_i$. This is equivalent to $\overline{t} \le t_\alpha$, where

$$\overline{t} \stackrel{\text{def}}{=} \max\{t(x_1, \ldots, y_1, \ldots) : x_i \in \mathbf{x}_i, y_i \in \mathbf{y}_i\}.$$

To reliably reject the hypothesis, we need to check that

$$t(x_1, \ldots, y_1, \ldots) > t_\alpha$$

for all $x_i \in \mathbf{x}_i$ and $y_i \in \mathbf{y}_i$. This is equivalent to $\underline{t} > t_\alpha$, where

$$\underline{t} \stackrel{\text{def}}{=} \min\{t(x_1, \ldots, y_1, \ldots) : x_i \in \mathbf{x}_i, y_i \in \mathbf{y}_i\}.$$

When $t(x_1, \ldots, y_1, \ldots) < t_\alpha$ for some $x_i \in \mathbf{x}_i$ and $y_i \in \mathbf{y}_i$ and $t(x_1, \ldots, y_1, \ldots) > t_\alpha$ for some other values $x_i \in \mathbf{x}_i$ and $y_i \in \mathbf{y}_i$, then, based on the interval inputs $\mathbf{x}_i$ and $\mathbf{y}_i$, we can neither reliably confirm not reliably reject the corresponding hypothesis.

Thus, to check the hypothesis under interval data, we need to compute the values $\underline{t}$ and $\overline{t}$, i.e., we need to compute the *range*

$$[\underline{t}, \overline{t}] = \{t(x_1, \ldots, y_1, \ldots) : x_i \in \mathbf{x}_i, y_i \in \mathbf{y}_i\}.$$

Interval computations. Computation under interval uncertainty about inputs is known as *interval computations*. In general, computing the range is NP-hard; for t-tests, see, e.g., [1]. This means, crudely speaking, that no feasible algorithm can solve all instances of this problem.

In some cases, feasible algorithms are possible. For example, it is easy to compute the range of the mean $X = \frac{1}{n_x} \cdot \sum_{i=1}^{n_x} x_i$. Since this function in monotonic in all x_i, the range is

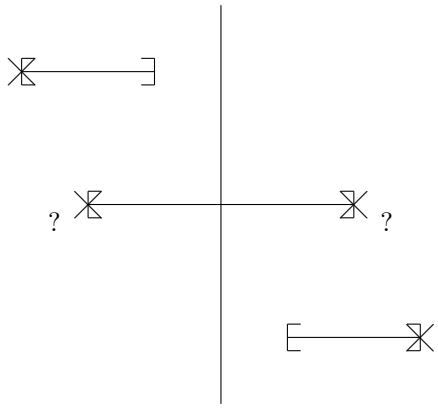
$$[\underline{X}, \overline{X}] = \left[\frac{1}{n_x} \cdot \sum_{i=1}^{n_x} \underline{x}_i, \frac{1}{n_x} \cdot \sum_{i=1}^{n_x} \overline{x}_i\right].$$

In this paper, we provide efficient algorithms for computing t-tests under privacy-motivated interval uncertainty.

Comment. These algorithms were first announced in [3].

2 Analysis of the Problem

Intuitive idea. All expressions for t have the form $\frac{\overline{X}}{s}$. Thus, the smallest value $\underline{t}$ of t is attained when s is the largest. This means that for each i, we select x_i which are as far from the mean as possible:

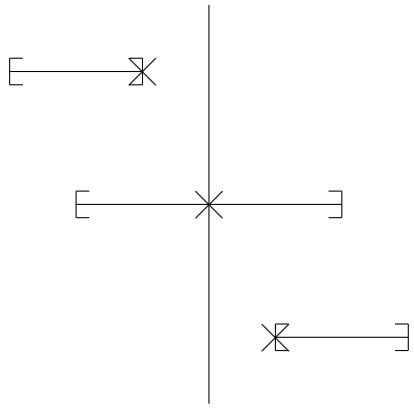


$\overline{X}$

Here:

- For intervals $[\underline{x}_i, \overline{x}_i]$ containing $\overline{X}$, we have two options: $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$.
- For all other intervals $[\underline{x}_i, \overline{x}_i]$, we have only one option.

Similarly, the largest value $\overline{t}$ of t is attained when s is the smallest. This means that for each i, we select x_i which are as close from the mean as possible:



$\overline{X}$

For each interval $[\underline{x}_i, \overline{x}_i]$, we have only one option:

- if $\overline{x}_i < \overline{X}$, then $x_i = \overline{x}_i$;
- if $\overline{X} < \underline{x}_i$, then $x_i = \underline{x}_i$;
- if $\underline{x}_i \le \overline{X} \le \overline{x}_i$, then $x_i = \overline{X}$.

Towards algorithm for $\overline{t}$. In general, a differentiable function $f(x)$ attains its maximum on $[\underline{x}, \overline{x}]$:

- either inside the interval, then $\dfrac{df}{dx} = 0$;
- or for $x_i^M = \underline{x}$, then $\dfrac{df}{dx} \le 0$;
- or for $x^M = \overline{x}$, then $\dfrac{df}{dx} \ge 0$.

So, for every i, the maximum $t = \overline{t}$ is attained:

- either when $\underline{x}_i < x_i^M < \overline{x}_i$ and $\dfrac{\partial t}{\partial x_i} = 0$;
- or when $x_i^M = \underline{x}_i$ and $\dfrac{\partial t}{\partial x_i} \le 0$;
- or when $x_i^M = \overline{x}_i$ and $\dfrac{\partial t}{\partial x_i} \ge 0$.

Here, the derivative $\dfrac{\partial t}{\partial x_i}$ is proportional to $x_i - c$ for some expression c which is independent on i. This expression c is a ratio whose numerator is quadratic in x_i and denominator is linear in x_i.

For example, for $t = \dfrac{\overline{X} - \mu_0}{s_X / \sqrt{n_x}} = \dfrac{\sqrt{n_x} \cdot \left(\overline{X} - \mu_0\right)}{s_X}$, we have

$$\frac{\partial t}{\partial x_i} = \frac{\sqrt{n_x}}{s_X^2} \cdot \left(\frac{1}{n_x} \cdot s_X - \left(\overline{X} - \mu_0\right) \cdot \frac{2x_i}{n_x - 1} \right) = -\frac{\sqrt{n_x}}{s_X^2} \cdot \frac{2\left(\overline{X} - \mu_0\right)}{n_x - 1} \cdot (x_i - c),$$

where we denoted $c \stackrel{\text{def}}{=} \dfrac{(n_x - 1) \cdot s_X}{2n_x \cdot \left(\overline{X} - \mu_0\right)}$.

Let us consider all possible locations of c in relation to the interval $[\underline{x}_i, \overline{x}_i]$.

- When $\underline{x}_i \le c \le \overline{x}_i$, we cannot have $x_i^M = \underline{x}_i$ and $x_i^M = \overline{x}_i$, so x_i^M is in between, thus $\dfrac{\partial t}{\partial x_i} = 0$ and $x_i^M = c$.
- Similarly, when $\overline{x}_i \le c$, we have $x_i^M = \overline{x}_i$.
- Finally, when $c \le \underline{x}_i$, we have $x_i^M = \underline{x}_i$.

In all three cases, x_i^M is the point from the interval $[\underline{x}_i, \overline{x}_i]$ which is the closest to c.

Algorithm for computing $\overline{t}$. To use the above observations, let us sort all endpoints of all given x-intervals into an increasing sequence:

$$x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n_x)}$$

To this sequence, let us add the infinities $x_{(0)} = -\infty$ and $x_{(2n_x+1)} = +\infty$. Then, the zones $[x_{(k)}, x_{(k+1)}]$ cover the whole real line. Therefore, the value c is in one of the zones $[x_{(k)}, x_{(k+1)}]$.

For each zone k, for each i, we either know x_i^M, or we know that $x_i^M = c$. Substituting these values $x_i = x_i^M$ and similar values $y_i = y_i^M$ into the expression $c(x_1, \ldots, x_{n_x}, y_1, \ldots, y_{n_y})$ for c, we get a quadratic equations for c.

For example, for $c = \dfrac{(n_x - 1) \cdot s_X}{2n_x \cdot \left(\overline{X} - \mu_0\right)}$, if we plug in the values $x_i = c$ for appropriate indices i into the equivalent equation $(n_x - 1) \cdot s_X = 2n_x \cdot \left(\overline{X} - \mu_0\right) \cdot c$ in which $\overline{X}$ is a linear function of c and s is quadratic in c, we get the equation which is indeed quadratic in c.

For each zone, we form the corresponding quadratic equation. After solving the quadratic equation, we find c.

Once we know c, we thus know all the maximizing values x_i^M. Based on these values, we compute the value t corresponding to the case when c is in the k-th zone.

We repeat this for each pair of X- and Y-zones. The largest of the computed values t is the desired maximum $\overline{t}$.

This algorithm is feasible. For each sample of size n, we have $2n$ bounds, so we have $2n + 1 = O(n)$ zones. Thus, we have $O(n) \cdot O(n) = O(n^2)$ pairs of zones.

For each pair of zone, we need $O(n)$ computational steps. Thus, overall, we need $O(n^2) \cdot O(n) = O(n^3)$ steps. So, our algorithm is indeed feasible.

Towards an algorithm for computing $\underline{t}$. According to calculus, a function $f(x)$ attains its minimum on an interval $[\underline{x}, \overline{x}]$:

- either inside the interval, then $\dfrac{df}{dx} = 0$;
- or for $x^m = \underline{x}$, then $\dfrac{df}{dx} \ge 0$;
- or for $x^m = \overline{x}$, then $\dfrac{df}{dx} \le 0$.

So, for every i, when the minimum $t = \underline{t}$ is attained:

- either when $\underline{x}_i < x_i^m < \overline{x}_i$ and $\dfrac{\partial t}{\partial x_i} = 0$;

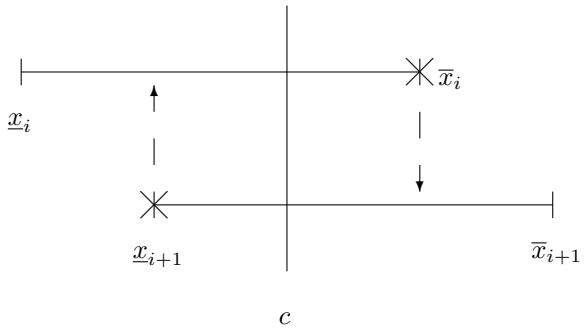
- or when $x_i^m = \underline{x}_i$ and $\dfrac{\partial t}{\partial x_i} \geq 0$;
- or when $x_i^m = \overline{x}_i$ and $\dfrac{\partial t}{\partial x_i} \leq 0$.

As we have mentioned, th i-th partial derivative $\dfrac{\partial t}{\partial x_i}$ is proportional to $x_i - c$ for some expression c which is independent on i. Let us consider all possible locations of the value c in relation to the interval $[\underline{x}_i, \overline{x}_i]$.

- When $c < \underline{x}_i$, we cannot have $x_i^m = \underline{x}_i$ and $\underline{x}_i < x_i^m < \overline{x}_i$, so $x_i^m = \overline{x}_i$.
- Similarly, when $\overline{x}_i < c$, we have $x_i^m = \underline{x}_i$.
- Finally, when $\underline{x}_i \leq c \leq \overline{x}_i$, we can have both $x_i^m = \underline{x}_i$ and $x_i^m = \overline{x}_i$.

For privacy data, intervals $[\underline{x}_i, \overline{x}_i]$ can be sorted so that $\underline{x}_i \leq \underline{x}_{i+1}$ and $\overline{x}_i \leq \overline{x}_{i+1}$. Let us show that min is attained when $x_i^m \leq x_i^{m+1}$.

Indeed, the only possibility for $x_i^m > x_{i+1}^m$ is when both intervals contain c, $x_i^m = \overline{x}_i$, and $x_{i+1}^m = \underline{x}_{i+1}$. In this case, since t is symmetric w.r.t. all x_i we can swap these values and take $x_i^m = \underline{x}_{i+1}$, and $x_{i+1}^m = \overline{x}_i$:



We see that the resulting tuple is not minimizing.

This show that indeed, $x_i^m \leq x_{i+1}^m$ for all i. Thus, there exists k for which the minimizing sequence x_i^m has the form

$$(\underline{x}_1, \ldots, \underline{x}_{k_x}, \overline{x}_{k_x+1}, \ldots, \overline{x}_{n_x}).$$

We have such thresholds k_x and k_y for both samples.

Resulting algorithm for computing $\underline{t}$. For each pair of thresholds k_x and k_y, we consider sequences

$$(\underline{x}_1, \ldots, \underline{x}_{k_x}, \overline{x}_{k_x+1}, \ldots, \overline{x}_{n_x})$$

and

$$(\underline{y}_1, \ldots, \underline{y}_{k_y}, \overline{y}_{k_y+1}, \ldots, \overline{y}_{n_y}).$$

Based on these sequences, we compute the value t. The smallest of these values t is the desired value $\underline{t}$.

This algorithm is feasible. There are n^2 pairs of such thresholds. For each pair, we know the values x_i and thus, we can compute t by using time $O(n)$. So, the above algorithm takes time $O(n^2) \cdot O(n) = O(n^3)$ and is, thus, feasible.

This algorithm can be further improved. How can we make computations faster?

When we change from k to $k+1$, only one value changes x^m_{k+1}, from $\underline{x}_{k+1}$ to $\overline{x}_{k+1}$. Thus, we can change $\overline{X}$, $\overline{Y}$, s_X, and s_Y in $O(1)$ steps. With this improvement, we can compute $\underline{t}$ is time $O(n^2)$.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721.

The authors are thankful to Scott Ferson and to all the participants of the Dynamica Expo Conference (El Paso, Texas, November 14–15, 2014) for valuable discussions.

References

[1] M. Černy and M. Hladík, "The complexity of computation and approximation of the t-ratio over one-dimensional interval data", *Computational Statistics and Data Analysis*, 2014, Vol. 80, pp. 26–43.

[2] S. Ferson, V. Kreinovich, J. Hajagos, W. Oberkampf, and L. Ginzburg, *Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty*, Sandia National Laboratories, Report SAND2007-0939, May 2007.

[3] V. Kreinovich and C. Servin, "How to test hypotheses when exact values are replaced by intervals to protect privacy: case of t-tests", *Abstracts of the Proceedings of the Dynamica Expo Conference*, El Paso, Texas, November 14–15, 2014, p. 3.

[4] H. T. Nguyen, V. Kreinovich, B. Wu, and G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Springer Verlag, Berlin, Heidelberg, 2012.

[5] C. Servin and V. Kreinovich, *Propagation of Interval and Probabilistic Uncertainty in Cyberinfrastructure-Related Data Processing and Data Fusion*, Springer Verlag, Berlin, Heidelberg, 2015.

[6] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC Press, Boca Raton, Florida, 2011.