STAT 778: Midterm Exam

Tom Wallace

March 25, 2018

Preface: Program Organization and Compilation

Source code is contained in `midterm.c`. The program requires the GNU Scientific Library (GSL), an open-source numerical library. It can be obtained from `www.gnu.org/software/gsl`; or, it can installed from most standard Linux package managers. An example command to achieve the latter is:

```
sudo apt-get install gsl-bin libgsl-dev
```

Compilation of `midterm.c` is best achieved in two steps. First, use the below command to compile the program but not link it. You may need to change the argument passed to the `-I` flag to wherever the `gsl` header files live on your computer.

```
gcc -I/usr/include -c midterm.c
```

This command should create an object file `midterm.o`. Link this object file to relevant libraries with the following command. You may need to change the argument passed to the `-L` flag to wherever `libgsl` lives on your computer.

```
gcc -L/usr/lib midterm.o -o midterm -lgsl -lgslcblas -lm
```

Once successfully compiled, the program can be executed. It does not require any arguments. Output is comma-separated text printed to `stdout`. You likely want to pipe this output to a text file, as per the following command:

```
./midterm > output.csv
```

The following R code is useful for analyzing the output data:

```
library(plyr)
df <- read.csv("output.csv", header = T)
analysis <- ddply(df, c("distribution", "n", "m1", "m2"), summarize, emp_I_t =
mean(I_t), emp_I_w = mean(I_w), emp_II_t = mean(II_t), emp_II_w = mean(II_w))
df2[df2$distribution == 1, "distribution"] <- "Normal"
df2[df2$distribution == 2, "distribution"] <- "Normal (contaminated)"
df2[df2$distribution == 3, "distribution"] <- "Exponential"
```

Introduction

This study seeks to compare the performance of the two-sample t-test and the Wilcoxon rank-sum test. It uses simulation to do so. Data is randomly generated under different scenarios. For each scenario, the two methods are used to test the null hypothesis of no difference of means against the simple alternate hypothesis. The goal is to ascertain which method performs better by various criteria.

The remainder of this document is organized into two sections. The **Methods** section provides more detail on how the two methods were implemented and how their performance was compared. The **Simulation Study** section presents output data and results.

Methods

Tests for Difference of Means

The study compares the performance of two different tests of means. The first is **Welch's t-test**. This test assumes that the two populations are independent (i.e. unpaired), that they have normal distributions, and that they may have unequal variances. The test statistic t is calculated as:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{1}$$

with $\bar{X}_i$, s_i^2, and n_i denoting the sample mean, sample variance, and sample size of group i. The degrees of freedom for the t test statistic are calculated by the Welch-Sattherthwaite equation:

$$df \approx \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{s_1^4}{n_1^2(n_1-1)} + \frac{s_2^4}{n_2^2(n_2-1)}} \tag{2}$$

The second is the **Wilcoxon rank-sum test**. This test makes no parametric assumptions nor any assumptions regarding common variance. Observations from the two groups are pooled, and then ranked in ascending order. The sum of ranks is taken for a group (which does not matter). The u statistic is given by:

$$u_1 = R_1 - \frac{n_1(n_1+1)}{2} \tag{3}$$

where R_1 is the sum of ranks of group 1, and n_1 is the sample size of group 1. This study uses the normal approximation for groups with $n_i \geq 25$:

$$z = \frac{u - \mu_u}{\sigma_u} \tag{4}$$

where $\mu_u = \frac{n_1 n_2}{2}$ and $\sigma_u = \sqrt{\frac{n_1 n_2 (n_1+n_2+1)}{12}}$

Hypothesis Testing

The null hypothesis is no difference in group means, with a two-sided alternate hypothesis:

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

A significance level of $\alpha = 0.05$ is used.

Simulation Study

Approach

The basic approach was to generate simulated data for two groups according to some specification; apply the t-test and Wilcoxon rank-sum test to the simulated data; and assess the empirical type I error rate and empirical type II error rate of each test. Different specifications were used, with variation in group size, distribution (including presence of outliers), and true difference in means. Each specification was simulated 1000 times. The tested specifications included various combinations of:

- **Distribution**
  - Normal
  - Normal, with 2% chance of outlier ($\mu_{\text{outlier}} = 100\mu$)
  - Exponential
- **Group size**
  - $n = 25$
  - $n = 50$
  - $n = 100$
- **True difference in means**
  - None ($\mu_2 = \mu_1$)
  - $\mu_2 = 1.05\mu_1$
  - $\mu_2 = 1.5\mu_1$
  - $\mu_2 = 2\mu_1$

Results

Quantitative results are presented in Table 1.

The t-test and Wilcoxon rank-sum test perform about the same on normally-distributed data. The power of the tests is low for small true difference in means and improves at a similar rate as the true difference in means becomes greater.

The Wilcoxon rank-sum test greatly outperforms the t-test on normal data with outliers. The power of the t-test is low and scarcely improves with increased true difference in means; in contrast, the Wilcoxon rank-sum test is much more powerful. This simulation finding makes theoretical sense: the t-test is based on means (which are sensitive to outliers) whereas the Wilcoxon rank-sum test is based on medians (which are much less sensitive to outliers). This finding is true for all tested sample sizes.

The t-test seems to be more powerful than the Wilcoxon rank-sum test on exponential data, particularly as sample size increases. This is surprising as normality is an assumption of the t-test. The Wilcoxon test performs about the same on the exponential data as it does on other distributions; this is theoretically expected as the test makes no parametric assumptions.

Table 1: Simulation Results

Distribution	**n**	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$	$\bar{\boldsymbol{\alpha}}_t$	$\bar{\boldsymbol{\alpha}}_u$	$\bar{\boldsymbol{\beta}}_t$	$\bar{\boldsymbol{\beta}}_u$
Normal	25	1.00	1.00	0.04	0.05	-	-
			1.05	-	-	0.95	0.96
			1.50	-	-	0.58	0.59
			2.00	-	-	0.07	0.07
	50	1.00	1.00	0.05	0.05	-	-
			1.05	-	-	0.94	0.94
			1.50	-	-	0.31	0.34
			2.00	-	-	0.00	0.00
	100	1.00	1.00	0.05	0.05	-	-
			1.05	-	-	0.93	0.94
			1.50	-	-	0.06	0.07
			2.00	-	-	0.00	0.00
Normal (contaminated)	25	1.00	1.00	0.02	0.04	-	-
			1.05	-	-	0.98	0.95
			1.50	-	-	0.84	0.62
			2.00	-	-	0.67	0.11
	50	1.00	1.00	0.01	0.05	-	-
			1.05	-	-	0.98	0.94
			1.50	-	-	0.91	0.34
			2.00	-	-	0.85	0.00
	100	1.00	1.00	0.04	0.06	-	-
			1.05	-	-	0.96	0.94
			1.50	-	-	0.94	0.09
			2.00	-	-	0.91	0.00
Exponential	25	1.00	1.00	0.04	0.05	-	-
			1.05	-	-	0.96	0.95
			1.50	-	-	0.73	0.76
			2.00	-	-	0.39	0.48
	50	1.00	1.00	0.06	0.06	-	-
			1.05	-	-	0.94	0.95
			1.50	-	-	0.50	0.59
			2.00	-	-	0.09	0.15
	100	1.00	1.00	0.04	0.04	-	-
			1.05	-	-	0.94	0.94
			1.50	-	-	0.19	0.33
			2.00	-	-	0.00	0.01

Each specification simulated 1000 times
$\sigma^2 = 1$ for all normal and contaminated normal distributions
$\bar{\alpha}_t$ = empirical Type I error probability, t-test
$\bar{\alpha}_u$ = empirical Type I error probability, Wilcoxon rank-sum test
$\bar{\beta}_t$ = empirical Type II error probability, t-test
$\bar{\beta}_u$ = empirical Type II error probability, Wilcoxon rank-sum test

Comments Regarding Course

Overall, I have a positive impression of the class thus far. My C programming skills have gotten much sharper. I particularly appreciate that the instructor focuses on more on students learning useful skills than on grades. The purpose of graduate school is to prepare students to do professional research, and so I feel that grades are relatively superfluous. They can be a useful barometer of whether students are learning the skills needed for research, but the relationship is not that strong. In some other courses, the pressure to do well on frequent, difficult graded assignments actually hurt my progress as a statistician. I got excellent grades, but did so by focusing more on learning to crank out correct answers than on truly understanding the material. I really appreciate the different approach of this class. I put in just as much work, but am free to do so in a way that aids my long-term progress.

I hope that in the second half of the course we spend more time on algorithms. Techniques such as jackknife, bootstrap, EM, and the like are fundamental to modern statistics. Most of our work in the first half of the class has been on learning C, not learning statistical algorithms—we have been implemented very basic procedures such as Kaplan-Meier. I look forward to learning more advanced algorithms.