UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Stenver Jerkku

Paralell Wilcoxon Signed-rank tests

Bachelor's Thesis (6 ECTS)

Supervisor: Sven Laur, PhD

Author: ..... "....." 2012

Supervisor: ..... "....." 2012

Allowed to defence
Professor: ..... "....." 2012

Tartu 2012

Contents

Abstract

The goal of this project is to create a C++ library that can be run from R statistics program and terminal. The library should be able to run tens of thousands of Wilcoxon Signed Ranked Tests in parallel in mere seconds. It should also be able to run these tests accurately, regardless of the sample size(The number of pairs) and take into account that some test might be missing or flawed. The library is developed for BIIT(Bioinformatics, Algorithmics and Data mining group). BIIT is joint research group between the Department of Computer Science (University of Tartu), Quretec, and the Estonian Biocenter. Its main research topics and capabilities include the gene regulation, gene expression data analysis, biological data mining and others.

1 Introduction

For some tasks, the BIIT needs to do huge number of tests with Wilcoxon signed ranked test. However, current implementations of the tests dont run on native code, have performance problems by doing a lot of recomputations and can only be ran 1 at the time.They dont take into an account that the data might contain nulls and they dont support NetCDF file format, which is used by BIIT to hold complex gene data. This project fixes these problems by creating a library that is both accurate and fast even with large number of tests. It uses approximation and avoids unnecessary computation to increase its speed and reliability. The first chapter will cover the different definitions used in this paper. The second chapter will covers

2 Definitions

2.1 Variables

1. N - The sample size in a wilcoxon signed-rank test.
2. $V(N,k)$ - table. We need this table only, for calculating $P(N,k)$. It has no other use for us. It signifies the number of different possible signs in an N sized rank array where $k = W$. So for example, $V(3,0)$ would tell us that if we have 3 variables, then how many different signs could those ranks have, if their sum is 0.
3. P - Value that has been calculated. We need it because we want to know how accurate approximated P value is in practical use, using the V table by using the following formula:
4. $P = \sum_{k=W}^{\infty} \frac{V(N,k)}{2^n}$
5. $P(N,k)$ - table that has P values in it for every N and K. The table gets exponentially larger, the bigger N is. Table with the size $N = 80$ is around 2.5 MB in plaintext file, while $N = 800$ is many gigabytes.
6. P_{approx} - value that has been calculated with the Gaussian Distribution formulae.
7. $P_{approx}(N,k)$ - table that holds every Papprox value for each N and K.
8. X - the high enough N value where using Gaussian Distribution is accurate enough that we can use, instead of a accurate P table.

2.2 NetCDF file format

NetCDF is an open source standard of a set of data formats, programming interfaces, and software libraries that help read and write scientific data files. `http://www.unidata.ucar.edu/software/netcdf/docs/what_is_netcdf.html`[2] Data can be held in a structured manner in a NetCDF file. You can define dimensions, name them, put variables in the dimensions and later retrieve or change them. This is useful, for example, to hold matrix like data in a file and have fast lookups. In addition, you can define helper dimensions for the matrix for extra information. NetCDF file format is used by the BIIT researh group to hold gene data and will be given as an input file for the program.

2.3 Bonferroni correction

If you make multiple hypothesis on a test, then you increase the risk in which you reject null hypotheses when its actually true. The Bonferroni test helps to counteract this in a simple way - for each hypothesis you make on a test, you should use a significance level N times lower than before. So for example, if you make N hypothesis and want want a significance level , then you should run each test at a significance level of $/N$.

If you want to use Bonferron correction with Wilcoxon signed-rank test, then you need to keep in mind that the approximated P value significance level needs to be much more accurate, since the bonferron test divides it by the amount of features tested. This means higher the X value, the more features you add in the test. `http://en.wikipedia.org/wiki/Bonferroni_correction`[3]

2.4 Wilcoxon signed-rank test

The Wilcoxon signed-rank test is a non-parametric(data doesnt have to have characteristic parameters or structure) statistical hypothesis test used when comparing two related samples, matched samples, or repeated measurements on a single sample to assess whether their population mean ranks differ - i.e. it is a paired difference test.

This means that with Wilcoxon test you can look at some measurable feature(eg. gene). If we have made two or more experiments in different conditions on it(eg gene experiments and their confirmation results), we can compare them with the test to see if they are relevant. H_0 means that the difference between pairs is zero. H_1 means it is not.

It was popularized by Sidney Siegel in his book "Nonparametric statistics - for the behavioral sciences". Sidney used the symbol T, instead of W in his book, because it was related, but not exactly the same as W. Because of this the test is sometimes referred to as the Wilcoxon T test and the test statistic is reported as a value of T.

2.4.1 Assumptions

1. Data is from the same population and paired. 2. pairs are independent and random. 3. The data is measured at least on an ordinal scale - i.e. it allows rank order by which data can be sorted. 4. The median has symmetric distribution of differences.

2.4.2 The formulae

The test first ranks the experiment values in an ascending list by subtracting the test experiment value from the original experiment value and then taking absolute value of it. Then it sorts them by size.(pairs $(3,5),(6,3),(5,10)$ would come $-2,3,-5$, their absolute values would be ranked $1,2,3$)

Then it finds out the sign of each experiment pairs(pairs $(3,5),(6,3),(5,10)$ would come $-2,3,-5$, which signs would be $-,+,-$, in another words, $-1,1,-1$)

Then it calculates the statistic W, which is the absolute value of the sum of the signed ranks, and finds out whether the N is large enough to approximate P value or it needs to use P value table to get the P value. If it can approximate P value, the test calculates Z value and compares it to approximated Z value. If the Z value is bigger than Zcritical, reject H_0 . If it cant, then it will just get the P value from built in table and if the P is smaller than 0.05, then reject H_0. The formula for wilcoxon test:

Let N be the sample size, the number of pairs. Thus, there are a total of $2*N$ data points. For $i=1,...,N$, let $x_{1,i}$ and $x_{2,i}$ denote the measurements.

1. For $i=1,..,N$, calculate $|x_{2,i}-x_{1,i}|$ and $sgn(x_{2,1}-x_{1,i})$, where is the sign function.

2. Exclude pairs with $|x_{2,i}-x_{1,i}|=0$. Let N_r be the reduced sample size.

3. Order the remaining N_r pairs from smallest absolute difference to largest absolute difference, $|x_{2,i}-x_{1,i}|$.

4. Rank the pairs, starting with the smallest as 1. Ties receive a rank equal to the average of the ranks they span. Let R_i denote the rank.

5. Calculate the test statistic W, the absolute value of the sum of the signed ranks. $W=|\sum_{i=1}^{N_r}[sgn(x_{2,i}-x_{1,i})*R_i]|$

6. As N_r increases, the sampling distribution of W converges to a normal distribution. Where X depends on how accurate you want your results to be

   (a) For $N_r >= X$, a $z-score$ can be calculated as
   $z=\frac{W-0.5}{\sigma_w}, \sigma_w=\sqrt{\frac{N_r(N_r+1)(2N_r+1)}{6}}$
   If $Z > Zcritical$ then reject H_0, where $Z_{critical}$ is calculated with Gaussian distribution

   (b) For $N_r < X$, W is compared to a $p-value$ that can be calculated from enumeration of all possible combinations of W given N_r.
   If $W >= P_{critical}$, N_r then reject H_0

2.4.3 Example

Given pairs $[6, 8, 3, 3, 2, -3, -3, 3, 1, 3]$.

1. Calculate absolute values and signs: Absolute values $[2, 0, -5, 6, 2]$ Signs $[1, 0, -1, 1, 1]$ 2. Exclude pairs absolute value is 0 3. Order the remaining pairs. Ties receive therank equal to average of the ranks they span $[2 \Rightarrow 2.5, -5 \Rightarrow 1, 6 \Rightarrow 4, 2 \Rightarrow 2.5]$ 4. Calculate the test statistic W $W = 1 * 2.5 + (-1 * 1) + 1 * 4 + 1 * 2.5 = 8$ 5. Since N_r is very small, we use a table to look up the p value. The calculation of p table is in a later chapter. $P(5, 8) = 0.125$ Since $0.125 > 0.05$, reject H_0

2.5 Calculating Exact P table

2.5.1 Theory

The P exact table is calculated by counting the number of different ways one can add or substract ranks with each other so he gets a certain sum.

For example, when N is 3, given $0, 1, 2$, one can combine them into $0 + 1 + 2$ or $0 + 1 - 2$ or $0 - 1 + 2$ or $0 - 1 - 2$. This gives us one way to get 3, one way to get 1, one way to get -1 and one way to get -3

This counting of possibilities can be continued until infinity and it forms a normal distribution that widens very fast. The edge of the distribution on a certain N is equal to the sum of every value until N

Once the distribution with large enough N is calculated, you can calculate the probability P that a combination of ranks falls into that sum. In other words given distribution table V which shows how many different possibilities there is to combine ranks to get a certain sum given a certain number of ranks, we can calculate the probability that our ranks form that certain sum.

For example, when N is 3, given $0, 1, 2$, we can calculate that the probability that our there is a 0.25% that our value is 3 or lower and 0.5% that our value is 1 or lower.

This distribution is the most accurate table you can compare your wilcoxon test results on - it shows the exact probability that your tests falls into a certain normal distribution range. However, the table is expensive to calculate with $O(n^2)$ complexity.

2.5.2 The V table formula

Given N which shows us the number of ranks starting from 0 and k which shows us the sum that we are interested in. Recursively apply this algorithm until you reach to the $N = 1$. $V(N + 1, k) = V(N, k - 1) + V(N, k + N + 1)$

There are some additional conditions to the formula:

$V(N, k) = 0$ if $k < -N * \frac{N+1}{2}$ or $k > N * \frac{N+1}{2}$

$V(N, k) = 1$ if $k = -N * \frac{N+1}{2}$ or $k = N * \frac{N+1}{2}$

2.5.3 The P table formula

Given V table, N and k, we can calculate the P by $W = |\sum_{T=k}^{\infty}[\frac{V(N,T)}{2^n}]$

To get the P table, we go through all N and K values that we are interested in.

2.5.4 Code sample

You can find a python implementation of a code sample in `https://bitbucket.org/stenver/wilxoni-astaku-test/src/870cc6112d0de4483cd99c71e651c2d06feaaee9/WilcoxonVTable/WilcoxonVTable.py?at=default`

The V table calculation is showcased in the methods V and $calculateVvalues$

The P table calculation is showcased in the method $calculatePValues$

3 Approximating the P value

3.1 Theory

The biggest problem when calculating P values in the other Wilcoxon signed-rank tests is that when they have the samples Z value, then they need to compare it to a $Z_{critical}$ value. If the number of samples N is sufficiently large, the solution is simple - use the Gaussian Distribution formulas to find the $Z_{critical}$. In the BIIT research group, however, the N is usually not sufficiently large. However, once the N gets too small, you cant use the Gaussian Distribution anymore. Furthermore, it is not known where exactly the N limit is, so an arbitrary number is used for that purpose. The currently suggested N is 10.[4] In the case the N gets too small, you cant use Z. You must instead use W and you can calculate the $W_{critical}$ by Calculating the $V(N, k)$ table Calculate the $W_{critical}$ where $W_{critical} = P(N, k)$:

The bottlenecks here are the two steps . Other implementations of the test calculate the tables every time the test is run, thus if you run thousands of tests in a row, a lot of complex recomputation is done. This eventually becomes a performance issue. To speed this up - this project will calculate the entire $P(N, k)$ for every value that is possible where $N < X$. Then the $P(N, k)$ will be hardcoded inside the program for quick lookup of the value. This paper will focus on finding the X value

4 Finding the X value

4.1 Proving that P and P_{approx} get more similar as N increases

First we wanted to confirm that as N grows, P and P_{approx} value will get more and more similar. To do this, we took $P(N,k)$ value for each N where $N < 50$ and $P(N,k) = 0$ and $P(N,k-1)! = 0$. We then compared the $P(N,k)$ and $P_{approx}(N,k)$ for each N where: The results are on Figure 1.

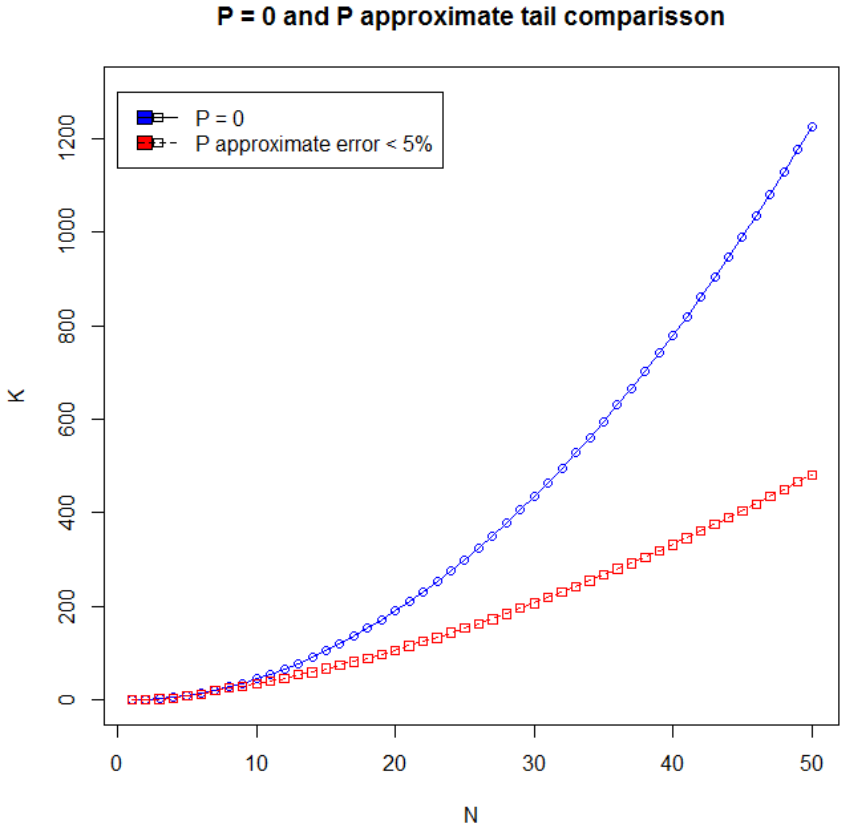

Figure 1: Approximate and accurate comparisson

Much to our surprise, as N grew, the gap between approximate and accurate P values grew bigger. This meant that as N grows, the Gaussian distribution will get more inaccurate toward the tails of the distribution. Previously we thought that with the growth of N, Gaussian would surely get more and more accurate. From Figure 2 we can see that Gaussian distribution is a little bit bigger and gets

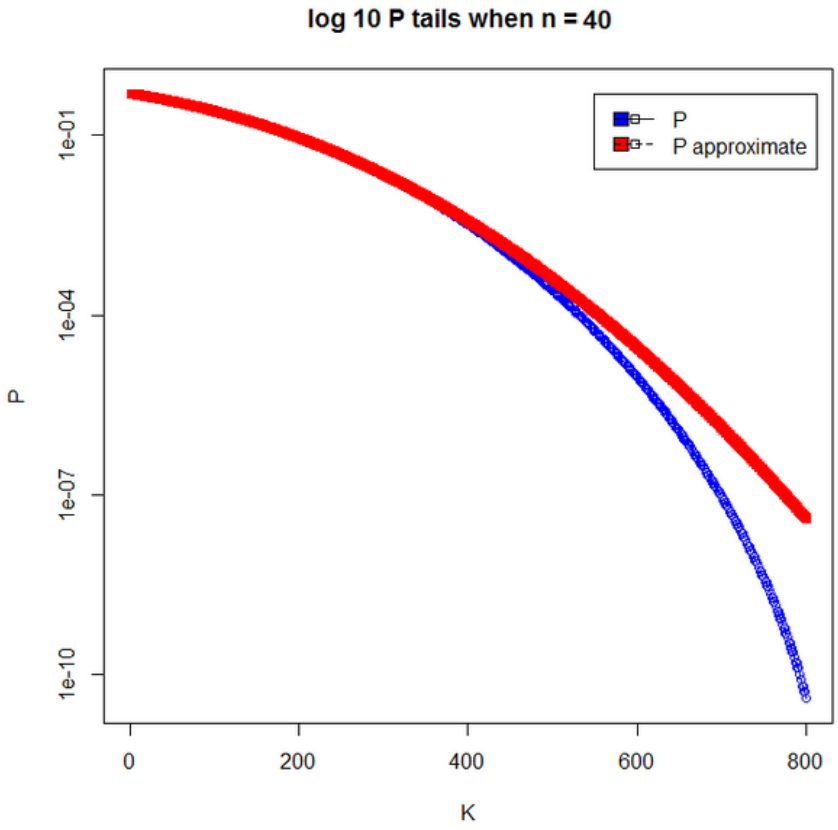

Figure 2: Relative error increasing toward edge of tail

bigger as K grows. Figure 2 illustrates the difference in position of between last accurate P_{appox} and $P = 0$. Accurate means that the relvative error is under 5%

To further investigate this finding, we decided to find out the minimum P value that you can get for each N while maintaining a certain error threshold. The thresholds chosen were $5\%, 10\%, 20\%, 50\%$. The formula used was the same as before, except the thresholds were replaced as needed.

From figure 3 we can see that as the N grows, the graph becomes stable, meaning that the distribution becomes a normal distribution at around $N > 10$ and $N < 25$. We can see that the minimum P value under error threshold does get smaller, as N increases, so this further proves that Gaussian distribution does get more accurate as N grows. This is because as N increases, you can take a P value closer to the tail of the distribution and still get an accurate value.

The figure 4 shows us the relative error as P grows when $N = 20$. The P values are logged on this graph. There are 2 noteworthy things here, however. First, around $P = 0.05$, the relative error actually get a little small tip toward

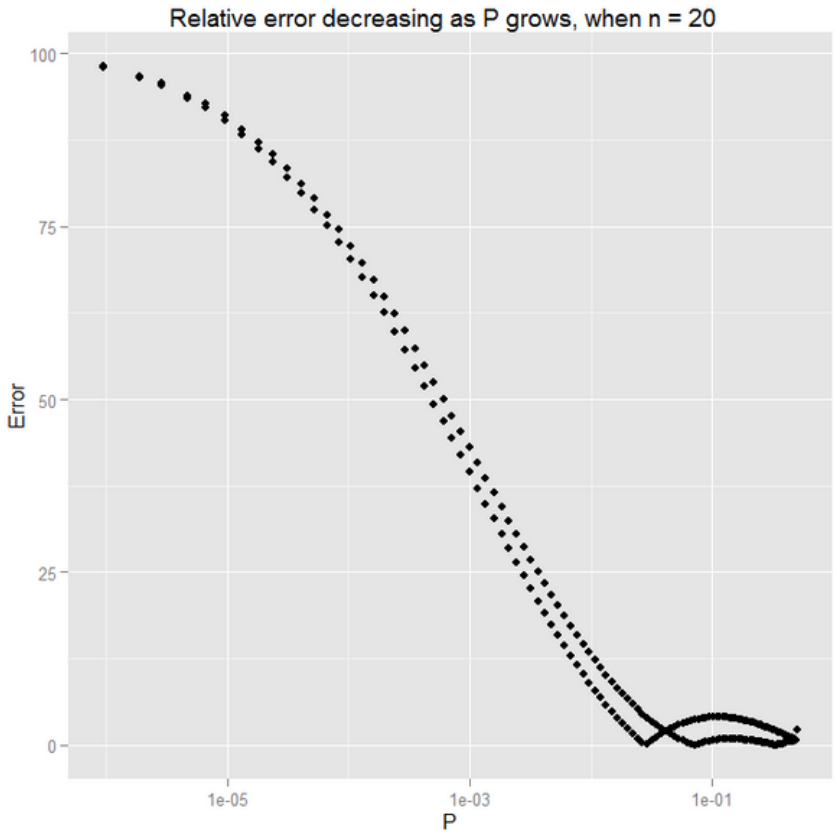

Figure 3: Comparing approximate probability relative error as sample size increses

accuracy. Second, it can be seen that there are two accuracy paths that the error takes, depending on weather in $P_{approx}(N,k)$ the K is even or odd number. This is a computational artefact caused by the recurrence of the $V(N,K)$ values and we will not make any conclusions of that.

4.2 Investigating the $P(N,k)$ and $P_{approx}(N,k)$ similarities

To further prove that we can start using Gaussian Distribution at a certain N, we needed to prove that Papprox will get more accurate as the N increases toward the tail of the distribution aswell.

To achive this, we found out the largest relative error between P and P_{approx} for each N and for each P where $P = (0.5 : 0.1), P = (0.1, 0.01), P = (0.001, 0.0001), ..., P = (10-14, 10-15)$

As can be seen from figure 5, the relative error increases exponentially. When $P = (10-5, 10-6)$ the relative error is massive. When $N = 200$, then the error is

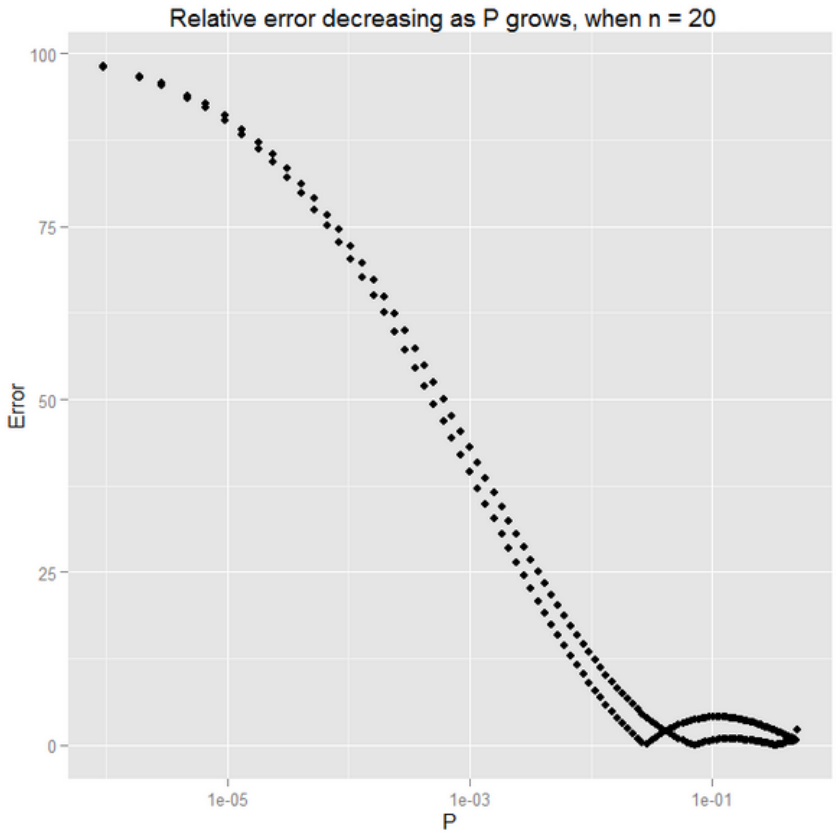

Figure 4: Showing the relative error as probabilty increases when sample size is 20

still around 40%. The bigger the N gets, the bigger the error at the tail. However, it can also be seen that the error decreases steadily as N grows for each of the P range chosen.

This means that even though the error of the distribution tail edge does increase as N grows, overall, both distributions get more and more similar to the gaussian distribution

To further investigate the relation between $P(N,k)$ and $P_{approx}(N,k)$, we wanted to see the difference between all P values when $P = 50$ and when $P = 25$

As we can see from 6 or 7, when the $N = 50$ or $N = 25$, then the $P(N,k)$ and $P_{approx}(N,k)$ values are almost the same with slight differences - the $P_{approximate}$ is a little bit larger. When $N = 25$, the graph is a little less smooth, which tells us that the differences are not as similar as they are when $N = 50$.

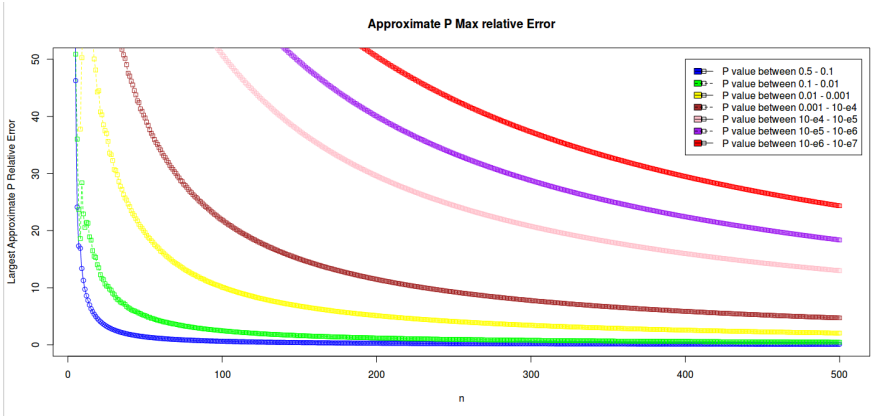

Figure 5: Showing the maximum approximate relative error in probability ranges, as sample size increases

4.3 Determining the X value

Judging by all the data we have gathered, we can clearly see that as N grows, $P(N,k)$ and $P_{approx}(N,k)$ tail edges grow apart but overall the distribution gets more and more similar. Determining the X value depends on the amount of data we have and the accuracy of the result we want. The most helpful graphs to help us determine the necessary X value is figure 5. Table 1 illustrates the number of samples(N) needed for a certain relative accuaracy. So the X value depends on the number of measurements and the required accuracy where $X = N$:

Table 1 showing the required N for measurements size while maintaining a relative error

As pointed out in the beginning, when the N grows, the $P(N,k)$ table size grows exponentially. As such, we needed to find a balance between accuaracy and precalculated table thats still within acceptable size.

We chose the size to be $N = 80$, since the plaintext file of the precalculated table with that N is around 2.5 MB.

This will allow us to run tests on 10 experiments while maintaining a 10% accuracy.

Future research could look into approximating the accurate $P(N,k)$ table and potentially decrease the size of the file, increase the precalculated table size while keeping the accuracy relatevly the same.

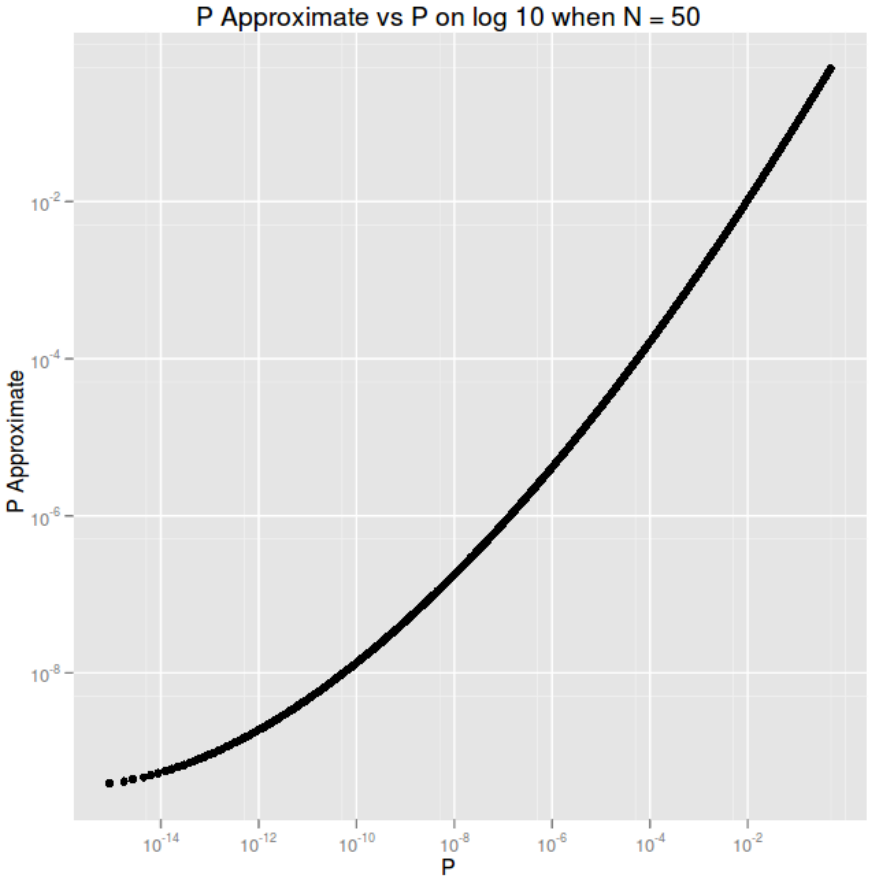

Figure 6: Relative value of all actual probabilities and approximated probabilities when sample size is 50

5 The implementaion

Repository that contains everything about our findings can be found in `https://bitbucket.org/stenver/wilxoni-astaku-test/src/870cc6112d0d?at=default` repository

The repository contains a number of folders.

5.1 RcppWilcoxonTest

An interface that connects our implementation of the optimized Wilcoxon algorithm to the R. Note that you must have installed our implementation to use it. The interface must be compiled with separate R commands from the command line. You need Rcpp packages for R to use it. It can be installed in R console by

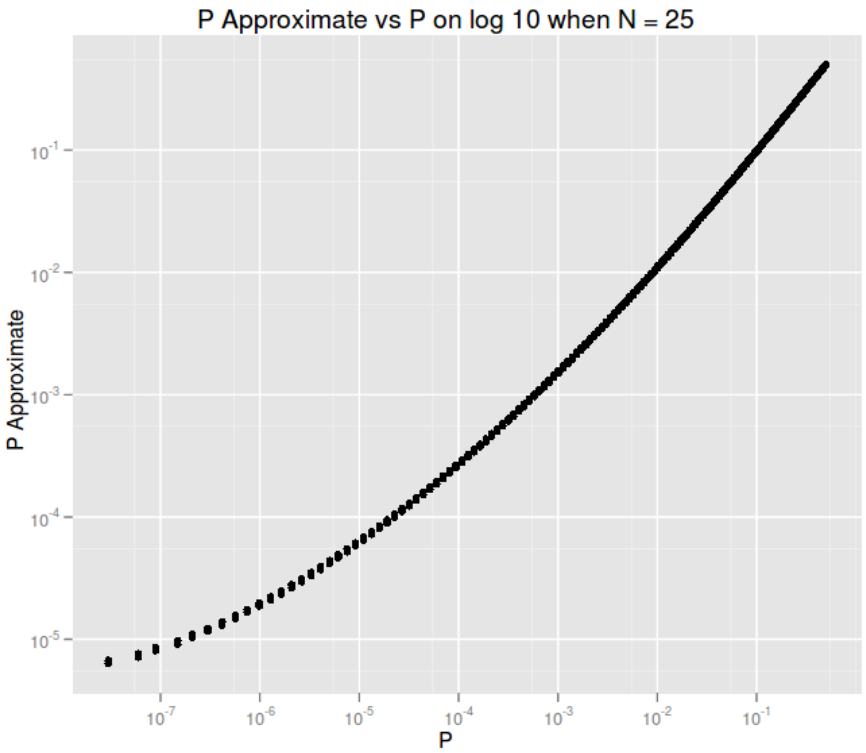

Figure 7: Relative value of all actual probabilities and approximated probabilities when sample size is 25

running:

```
$install.packages("Rcpp")
```

After that, the package can be installed to R by running the command in the command line in the folder:

```
$R CMD INSTALL .
```

You can now load our library in R by calling the

```
>library('RcppWilcoxonTest')
```

and you invoke the function by calling

```
>RcppWilcoxonTest::WilxTest(dataMatrix,
dataXsize, dataYsize, testIndexes, controlIndexes)
```

5.2 TerminalWilcoxonTest

An interface that connects our implementation of the optimized Wilcoxon algorithm to the R. Note that you must have installed our implementation to use it.

Table 1: Showing the minimal required sample size for a certain relative error and number of measurements

Measurements	Error	Required N
10	5%	25
100		250
1000		500
10000		> 500
100000		> 500
10	10%	80
100		200
1000		> 500
10000		> 500
100000		> 500
10	20%	50
100		100
1000		280
10000		500
100000		> 500
10	50%	25
100		50
1000		100
10000		150
100000		200

The interface can be compiled by going to the folder, compiling and running help for further help.

```
$make
WilcoxonTest    help
```

Currently only supports NetCDF file as input data.

5.3 WilcoxonTestLibrary

Our implementation for the optimized wilcoxon test. You library can be installed by going to the folder and running

```
make install
```

5.4 WilcoxonVTable

Python program that can calculate V and P tables, print them, create files of the tables and create a number of graphs on the tables.

5.5 Seminar_paper

The folder that contains this paper and all images attached to it. It also contains R programs to create those images.

6 Conclusion

We found out that Gaussian distribution tail edge grows larger apart from the $P(N, K)$ as N increases while the overall distribution grows more similar. In addition, we confirmed that if we want to run thousands of parallel tests, then using approximation is not reliable and instead an accurate table of P values must be used. Calculating that table is very expensive and thus it is advised to precalculate the table for the program. However, our research shows that if we want to have 10 000 measurements run under 5% relative error, then the hardcoded N should be over 1000. Table of that size would take many gigabytes of space and is impractical to use. For the time being, we will keep the precalculated table $N = 80$. This will allow us to make 10 parallel measurements with 10% error. In the future, we will look into optimizing the precalculated table to use larger data N sizes.

Overall the research included some surprising results and can be considered a success. The library is at its final stages of development and already shows positive results.

7 References

1 Wikipedia Foundation Inc, Wilcoxon Signed-rank test (http://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test), 14 August 2013

2 The Unidata Program Center, NetCdf (http://www.unidata.ucar.edu/software/netcdf/docs/what_is_netcdf.html), 25 September 2013

3 Wikipedia Foundation Inc, Bonferroni correction (http://en.wikipedia.org/wiki/Bonferroni_correction), 18 September 2013

4 Lowry, Richard. "Concepts & Applications of Inferential Statistics". (http://vassarstats.net/textbook/ch12a.html), 24 March 2011.