COUNTING STATISTICS:
THE SIGNIFICANCE OF ERRORS
AND THE PROPERTIES OF RARE EVENTS

1. INTRODUCTION

Rare events (*e.g.* earthquakes, uncommon diseases, winning the lottery...) have some very specific statistical properties. In this series of experiments we will use prototypical rare event: Radioactive decay. There is no way for you to make a nucleus decay, nor can you stop one from decaying, therefore there is no way for you to affect the outcome of the experiment. By removing the most significant source of bias and error (the experimenter) we are left with a clean system to study. The statistical properties are fully defined and completely outside your control or influence, so if you fail to observe those properties (within the uncertainties that are *also* fully defined by your experimental conditions) then you have done something wrong. The uncompromising nature of this situation means that we can focus on the analysis and worry much less about the mechanics of the actual experiment.

There are several purposes behind the design of this experiment.

(1) Get you to integrate everything that you just learned about the arduino and python into doing something useful. You know how to run an arduino and you know how to use python to extract data from it. Now you get to use them both together.

(2) Taking data over long periods of time without becoming bored is something that computers do far better than people. Reducing extremely large data sets to something more manageable is another great task for a computer. This experiment relies on both.

(3) At a more serious level, this experiment will demonstrate how the quality of data affects the significance of the conclusions that can be drawn from it. The analysis draws heavily on the ideas of basic statistics, so a review of such material is strongly advised before starting the experiment. The experiment will produce a vast amount of information, therefore the manner in which you present this information is extremely important if you are to avoid swamping the reader. The problem of presentation is an integral part of any experiment.

The experimental measurement and the analysis procedure have been specifically chosen because the uncertainties associated with the measurements are purely statistical in origin and are readily determined. There is essentially no room for subjectivity in setting the error bars and therefore a rigorous statistical treatment is possible.

(4) Rare events have unusual spatial and temporal distributions that often lead people

to see a causal relationship where none exists. If a rare event happens, when should you expect the next one to occur? If you look at the spatial distribution of rare events that have no causal link, and pick one (at "random"), how far away would you expect the nearest one to be?

2. EXPERIMENTAL OBJECTIVES

When a γ–ray photon strikes a Geiger counter it responds by producing a 'click' sound. Switch on the one provided and bring a radioactive source close to it. It is immediately clear that the source does not produce a steady stream of photons, they arrive in bunches of various sizes with gaps of variable length between them. Radioactive decay is probably the best example of a truly random process.

PROBLEM: how many photons strike the Geiger counter in a particular time interval – one second for example? If you repeat the observation you will almost certainly obtain a different answer. Despite this variability, the average is a well defined quantity, so is the frequency distribution of the observations – for a given experimental arrangement you will obtain the same average count rate and distribution of counts.

Most of the initial tasks needed to collect data were completed at the end of the python component of the course. The geiger counter signal can be used as the interrupt source (instead of the button) in your most recent code, and minor modifications will allow it to send the number of events in a unit of time as a string to your python code where it can be saved to a text file ready for further analysis.

- You will need to program the arduino to count the number of events that occur in some (user-defined) time frame (*e.g.* one second). This will require the use of the timer and interrupt functions.
- You will need to write a python program to collect and save the data produced by the arduino code.
- You will need to write a program to extract data from the previous file and generate histograms.
- You will need to be able calculate a mean and standard deviation for a histogram.
- You will need to be able to establish uncertainties for the histogram bins.
- You will need to be able to compare your histograms to test distributions.
- You will need to...

You get the idea.

Try comparing your data to standard forms (e.g. Gaussian or Poisson) by plotting your data with the corresponding calculated form on top of it. Can you say which is a better representation of the data? What other distributions are possible? Are they any better? What distribution do you expect? Why?

3. DATA ANALYSIS

In order to show that the measured data follow one form better than some others it is necessary to have a more objective test of similarity. This test must answer three questions: (i) how similar is the data set to the expected distribution? (ii) how significant is this similarity given the uncertainty in the data? (iii) how likely is it that you would obtain data like yours if the underlying distribution had the assumed form?

Many statistical tests exist with characteristics that make them more or less attractive/suitable for a specific problem and you are free to use as many as you wish to obtain your results. One standard method is the χ^2 or chi–squared test which compares the difference between the observed and expected values at each point, to the uncertainty in the measurement of that point. For data in the form of a histogram a common definition of the reduced χ^2 for a data set is given by:

$$\chi^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{(O_i - E_i)^2}{E_i}$$

where the sum runs over all of the n bins in the histogram. O_i and E_i are the observed and expected values respectively in bin i. This form make a number of assumptions about the data and its statistical properties, and may not always be valid. A more general form that makes fewer assumptions about the data and may be used for non–histogram data:

$$\chi^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{(O_i - E_i)^2}{\sigma_i^2}$$

where σ_i^2 is the variance of the observed value. What assumption is avoided by using this form?

USE THE SECOND DEFINITION IN ALL OF YOUR CALCULATIONS. Why?

O_i and E_i are readily available, they are your data and the distribution you want to compare it with. σ_i^2 is more difficult – how to determine the error on the contents of a bin in

your histogram? To solve this problem we turn to an invaluable technique for establishing errors or reliability – replication.

If you repeat (or replicate) the same experiment a large number of times, then, if the replicas are truly independent, the scatter in your results should reflect the uncertainties in the experiment. Furthermore, if you obtain a larger (or smaller) scatter than you expect from your original estimation of the uncertainties in your measurements, it is highly likely that either your estimates are wrong (it is common to overestimate the accuracy with which a measurement can be made), or that there are additional sources of error in the procedure. In principle, if the analysis is done correctly, and all sources of error are included, the two procedures (replication and error propagation) will give the same result. However, there is no way of being sure. Wherever possible, replication, rather than guesstimates, should be used to establish limits of precision.

Place the source so that you get an average of 7–10 counts in whatever time interval you have chosen and collect a large block of data. Several thousand events will do for a start.

You can now extract many, independent 100-element histograms and since the replicas are equivalent and independent, you can determine the mean and variance for each bin from the data set. Use the set of variances to perform a χ^2 test on each replica by comparing each one in turn with Gaussian and Poisson distributions. The mean and variance for the test distributions may be calculated from the set of means that you obtained above. Can you decide between the two forms for your data?

The test fails because your data are too noisy – the uncertainties are too large. You may improve the data by including more events in each histogram. You created the first series by using blocks of 100. Try re-extracting in blocks of 200, or 400, or ... Is it now clear why you first gathered the data into a file and then did the processing in a different program?

Repeat the analysis above. Can you now distinguish the two distributions? If you plot out one replica as a histogram you should see that it is much smoother – smaller apparent variances (Why?). Repeat the collapse of the data set and compare again.

How good does the data set have to be before you can tell the difference between Gaussian and Poisson distributions? Can you predict the required quality?

Finally, take means for each column determined in the first step and perform a χ^2 test on this set. What should you use as the variances for this data set? (It is *not* the variance you calculated with this set of means.)

You have shown that the output from a radioactive source follows a particular distribution. What can you say about the average number of γ–rays arriving at your detector in your chosen time interval? What is the uncertainty on this estimate?

It is an interesting, illuminating and *definitely non–optional* exercise to repeat the whole

analysis with the position of the source and the counting interval set so that the mean of your distribution is higher, 20 for example, and lower, 3–5 for example. As you increase the mean, the Poisson distribution becomes more symmetric and more difficult to distinguish from the Gaussian form. To see how close the distributions get to each other, try comparing calculated forms using the expressions given below.

4. TEMPORAL DISTRIBUTION OF DECAYS

You can also use the arduino and a modified version of your code to investigate the temporal distribution of decays. Use one event to start a clock, and the next one to stop it and write out the time difference. The 'stop' now becomes a 'start' and you wait for a new 'stop' event. Repeat for a long time and form a histogram of the time delays. What does it look like? Can you fit it?

5. DEFINITIONS OF STATISTICAL PARAMETERS

Mean:

$$< \mathrm{x} > = \frac{1}{n} \sum_{i=1}^{n} x_i$$

where the x_i are the data.

For data in a histogram:

$$< \mathrm{x} > = \frac{1}{\sum f_i} \sum x_i f_i$$

where the f_i are the frequencies of the x_i.

Variance:

$$\sigma^2 = \frac{1}{n}[\Sigma(x_i)^2] - < x >^2$$

This is equivalent to the usual definition:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - < x >)^2$$

but is more efficient for computer calculation since it only requires a single pass through the data. Generate your own version for data in a histogram.

The standard deviation is simply σ, *i.e.* the square root of the variance.

The standard error is generally, but not necessarily, two standard deviations. (Why?) The form of error used and the method used to estimate it should always be specified when reporting results.

The probability of getting ν counts in a time interval for a process following a Poisson distribution given by:

$$P_\mu(\nu) = e^{-\mu}\frac{\mu^\nu}{\nu!}$$

where ν is the bin number and μ is a positive parameter. For this distribution, the mean and variance are the same and equal to μ. One parameter specifies the full distribution form.

Similarly, if the process follows a Gaussian distribution:

$$P_{\mu,\sigma}(\nu) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(\nu-\mu)^2/2\sigma^2}$$

where μ is the mean and σ^2 is the variance of the distribution.

NOTE (1) both of these distributions are normalised to unit area. To compare them with your data you will have to multiply by the number of points in your data set. For data with a small mean, comparison with the Gaussian distribution may require a different normalisation.

NOTE (2) watch out for overflows and underflows when calculating these distributions. You my want to use Stirling's approximation (or some other means):

$$\ln(n!) \sim n\ln n - n$$

6. PREPARING THE REPORT

There are several questions raised in this hand out. The report must contain answers to them as it would not be possible to complete this experiment without knowing the answers. The most obvious question is: what distribution did you observe?

This guide gives (or implies) no justification for (i) the particular selection of distributions, (ii) the validity of the χ^2 test, (iii) the physical origin of the actual underlying distribution. These and many other points must be dealt with in the report.

Your report should describe how you obtained and analysed your data, but your primary concern is what you actually learned. Therefore, the discussion is an essential part of the report. It should identify clearly what form you found the distribution to take, how reliable this conclusion is (that requires you to explain how the χ^2 test works), why this result is expected from the physics/mathematics of the process.

7. SUGGESTED REFERENCES

John R. Taylor, "An Introduction to Error Analysis"

Phillip R. Bevington, "Data Reduction and Error Analysis for the Physical Sciences"

Louis Lyons, "Data Analysis for Physical Science Students"

Les Kirkup, "Experimental Methods"

William H. Press *et al.* "Numerical Recipes" (Mainly chapter 13)

This last book is an extremely useful general reference work for scientific computing, and well worth adding to your collection.

8. SOME ADVICE

Break the work into small, manageable sub-tasks so that you can test ideas and code before building on them.

For example: Generating a data file with just 50 or 100 entries in it will only take a few minutes. You can use that file to test almost all of the code you have to write.

- Can you create a histogram?
- Can you calculate the mean and standard deviation for it?
- Can you calculate the test distributions?
- Can you do a chi^2 test?

On a small data set you can check your code by using another program (even excel...) or by hand. That way when you move onto analysing a much larger file, you can be confident that everything is working.