EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH

CERN/EP 89-40
13 March, 1989

FITTING THE RATIO OF TWO DISTRIBUTIONS

Roger Barlow and Peter Hinde

OPAL Collaboration, CERN, Geneva, Switzerland
and

Department of Physics, Manchester University, Manchester, England

Abstract

A maximum likelihood technique is presented for the problem of fitting the ratio of two distributions, as a function of the sampling variable x , which may be multi-dimensional. No binning of the data is involved. A computer program, RATFIT, which implements this algorithm is described.

Submitted to *Computer Physics Communications*

1 The problem

The ratio of two distributions is often of interest. Suppose x is a continuous variable, which may be multi-dimensional, and that two types of events occur in x , S and B (which could stand for ‘signal’ and ‘background’), with distribution functions $S(x)$ and $B(x)$ respectively, normalised to the number of events N_S and N_B of each type. One may be interested not in the S and B distributions themselves, but in their relative strengths as a function of x

$$R(x) = \frac{S(x)}{B(x)} \quad (1)$$

or in the probability $P_S(x)$ that an event at some point x belongs to the signal rather than the background.

$$P_S(x) = \frac{S(x)}{S(x) + B(x)} \quad (2)$$

Such problems often arise when using the Monte Carlo technique to analyse experimental data. The real data will contain a mixture of S and B , in which the provenance of each event is not known. Monte Carlo samples can be generated according to S and B , and information from these finite samples, in which the type of every event is known, can be used to interpret the real data. For simple quantities (e.g. the proportions of S and B removed by a cut on x) the extraction of the appropriate number is straightforward. The question to be addressed is how to estimate more sophisticated quantities, like the form of the above ratios, using the two given samples.

One approach would be to fit the S and B samples separately and then form the required ratio. However it generally happens that the two distributions $S(x)$ and $B(x)$ contain complicated structure which is common to both of them (the effects of the finite detector acceptance, for example) which is difficult to fit. Legitimate errors in the estimates of these forms will lead to unnecessary errors when the ratio is taken.

A second approach is to histogram the observed S and B samples in x , using the same binning, to divide on a bin-by-bin basis, and fit a function through the resulting ratios. This has the advantage of working directly with the ratio, but in order for this to work satisfactorily, one must choose a bin size small enough not to obscure meaningful detail, and yet large enough for a large number of events to be contained in each bin. If this is not possible, as is frequently the case, then the result will depend on the binning technique chosen, and will not be meaningful. Such a choice gets especially difficult if x is multi-dimensional.

The approach presented here avoids these difficulties. The ratio is treated directly, so effects that are common to both samples, no matter how complicated, do not affect the result. No binning is used, so the dilemma of choosing a ‘correct’ bin size does not arise.

Such a problem may also occur in other areas. A linguist might have data recording the observed use in various geographical locations of two different words for the same object, and wish to know the relative variation in their distribution, ignoring the many factors (population of language speakers, where observations had been made, and so on) which are common to both and are not of any interest to the problem. An ornithologist might have similar data on the sightings of two similar species of bird, which could incorporate a third (time) measurement. So although the problem will be discussed here in terms of Monte Carlo distributions called ‘signal’ and ‘background’, these are merely names, and the technique can be applied to any two samples of data of different species.

2 The Method

We propose a method which uses a Maximum Likelihood fit to the individual x values, avoiding the loss of resolution inherent in forming histograms, and which fits the ratio directly.

The combined log likelihood for the two distributions, which is what one would normally maximise in a Maximum Likelihood fit, is:

$$\sum_{i \in S} \ln S(x_i) + \sum_{j \in B} \ln B(x_j)$$

Writing the total distribution of events as $T(x) = S(x) + B(x)$, so that

$$S(x) = \frac{S(x)}{S(x) + B(x)} T(x) = \frac{R(x)}{R(x) + 1} T(x)$$

$$B(x) = \frac{B(x)}{S(x) + B(x)} T(x) = \frac{1}{R(x) + 1} T(x)$$

this can be rewritten as

$$\sum_{i \in S} \left(\ln \left(\frac{R(x_i)}{1 + R(x_i)} \right) + \ln T(x_i) \right) + \sum_{j \in B} \left(\ln \left(\frac{1}{1 + R(x_j)} \right) + \ln T(x_j) \right)$$

It is proposed that

Proposition 1 *$R(x)$ is to be fitted. Then as the $T(x)$ logarithms are not affected by the ratio, they need not be considered in the maximisation*

This means that the likelihood to be maximised is merely

$$\ln L = \sum_{i \in S} -\ln(1 + 1/R(x_i)) + \sum_{j \in B} -\ln(1 + R(x_j)) \quad (3)$$

which is the logarithm of the probability that, given a set of events with these particular x values, they should happen to belong to the S and B samples in the way they do.

In order to perform this maximisation, it is then proposed that

Proposition 2 *$R(x)$ be parametrised by an exponentiated polynomial*

Therefore we write

$$R(x) = e^{r(x)} = e^{r_0 + r_1 x + r_2 x^2 + \dots + r_n x^n} \quad (4)$$

The coefficients r_q are then adjusted to maximise the likelihood

$$\ln L = \sum_{i \in S} -\ln(1 + e^{-r(x_i)}) + \sum_{j \in B} -\ln(1 + e^{r(x_j)}) \quad (5)$$

which means that the first derivative must be zero for all the coefficients r_q

$$\frac{\partial \ln L}{\partial r_q} = \sum_{i \in S} \frac{x_i^q}{1 + e^{-r(x_i)}} - \sum_{j \in B} \frac{x_j^q}{1 + e^{r(x_j)}} = 0 \quad \text{for } q = 0 \dots n \quad (6)$$

These equations have to be solved numerically.

2.1 Reasons for this approach

The functions $r(x)$, $R(x)$, $P_S(x)$ can all be trivially obtained from one another. The reasons for adopting this approach, i.e. that $r(x)$ is parametrised by a polynomial, rather than R or P_S , are as follows

1. The values of the resulting functions are automatically sensible. If $r(x)$ is a polynomial it can have any value, but no matter what the fit does to the coefficients in r , the resulting exponential $R(x)$ lies in the range 0 to ∞ , and $P_S(x)$ automatically lies between 0 and 1, as they must. Likewise the fitted forms for $\frac{S}{S+B}$ and $\frac{B}{S+B}$ automatically sum to 1. If instead one were to try and fit, for example, $R(x) = R_0 + R_1x + R_2x^2 \dots$ this can give non-physical negative values for R unless a complicated bound is placed on the coefficients.
2. The method and fitting are symmetric in S and B . If one redefines them, and fits B/S instead of S/B , then $R \rightarrow 1/R$ and $r \rightarrow -r$, so the solution of Equation 6 occurs at the same point, with the sign of all the r_q reversed. Thus the estimates $\widehat{B/S}$ and $\widehat{S/B}$ are mutually inverse. By contrast, if one were to fit $S/B = R_0 + R_1x \dots$, and then $B/S = R'_0 + R'_1x \dots$, their product would not be one.
3. The result depends very simply on the numbers of events N_S and N_B . If one has fitted the ratio using some N_S and N_B , this can be applied to any other values by a trivial multiplication of $R(x)$. This means that the number of Monte Carlo events to be generated in the two samples can be chosen according to convenience, and the results then applied to any suggested signal to background ratio in the real data.
4. In many physical situations the sample tends to pure S at one extreme of x , or pure B at the other, or both. P_S is then flat in these regions. A curve which is flat is very hard to parametrise by a polynomial, whereas if P_S tends to zero or one, then r tends to plus or minus infinity, which is easily accommodated at the extremes of x .
5. All the above works just as well if x has 2 or more dimensions. The polynomial gets more complicated...

$$\ln R(x_1, x_2) = r_{00} + r_{10}x_1 + r_{01}x_2 + r_{11}x_1x_2 + r_{12}x_1x_2^2 + r_{21}x_1^2x_2 \dots$$

but the likelihood function can be formed and solved as before.

2.2 Connection with the method of moments

The q_{th} moment of x from the S events in the data is

$$M_q^S = \sum_{i \in S} x_i^q$$

Given the probability function $P_S(x)$ and a sample of mixed S and B events where one didn't know which was which, one would estimate M_q^S as

$$M_q'^S = \sum_{k \in S, B} P_S(x_k) x_k^q = \sum_k \frac{S(x_k)}{S(x_k) + B(x_k)} x_k^q$$

Now, Equation 6 can be rewritten as

$$\sum_{i \in S} x_i^q \frac{B}{S+B} - \sum_{j \in B} x_j^q \frac{S}{S+B} = 0$$

whence, using $\frac{S+B}{S+B} = 1$

$$\sum_{i \in S} x_i^q = \sum_{i \in S} x_i^q \frac{S}{S+B} + \sum_{j \in B} x_j^q \frac{S}{S+B}$$

which, in terms of the two types of moments described above, gives

$$M_q^S = M_q'^S$$

and the same is of course true for M_q^B . Such equations hold for $q = 0, 1, 2 \dots n$.

Thus: the proposed Maximum Likelihood method is equivalent to ensuring that the moments of order up to n for the S and B data obtained from the combined sample using the fitted form for the probability are equal to the true moments obtained by taking the samples separately. This is a comforting confirmation that the approach is sensible.

3 Errors on the results

The second derivative of the log likelihood is

$$M_{qp} = \frac{\partial^2 \ln L}{\partial r_q \partial r_p} = \sum_{i \in S} \frac{x_i^{q+p} e^{r(x_i)}}{(1 + e^{r(x_i)})^2} + \sum_{j \in B} \frac{x_j^{q+p} e^{r(x_j)}}{(1 + e^{r(x_j)})^2} \quad (7)$$

The powers and minus signs conveniently cancel out so that the summation is the same over both data samples.

As usual for Maximum Likelihood estimates, in the limit of large samples the inverse of this matrix gives the errors on the estimated r_q

$$Cov(r_q, r_p) = (M^{-1})_{qp}$$

A proof is given in appendix A.

4 Adequacy of the polynomial

As is usual for Maximum Likelihood estimation, and in contrast to the χ^2 method, there is no clear parameter indicating whether the estimate found is a good or bad fit to the data. This means that there is no easy way of knowing whether the order n of the polynomial is adequate to describe the data, or whether a higher power is required.

Unless one has *a priori* knowledge, it is best to increase the value of n and observe the behaviour of the results. Useful indicators for this purpose are the “ $\mathcal{F}$ -numbers”, given by:

$$\mathcal{F}_1 = \sqrt{1 + \frac{(\overline{w_S^2} - \overline{w_S}^2) + (N_B/N_S)(\overline{w_B^2} - \overline{w_B}^2)}{(\overline{w_B} - \overline{w_S})^2}} \quad (8)$$

$$\mathcal{F}_2 = \sqrt{1 + \frac{(\overline{w_S^2} - \overline{w_S}^2) + (N_B/N_S)\overline{w_B^2}}{\overline{w_S}^2}} \quad (9)$$

Briefly, these arise as follows – for a full description and explanation see [1]. The numbers of S and B events in a mixed data sample can be estimated by forming a weighted sum, where the weight function is, on average, large for S events and small for B events. The best such weight function is just the probability $P_S(x)$, as given by Equation 2. The errors on resulting numbers that are due to Poisson fluctuations on N_S are multiplied by a number $\mathcal{F}$, which is always greater than or equal to one, and expresses the extra contribution to the error arising because the sample of signal events is not pure but adulterated by background. The $\mathcal{F}_1$ form applies if there is no prior information about the expected number of B events, the $\mathcal{F}_2$ form applies if the expected number of B events is known. In the above equations $\overline{w_S}$ is the average of the weight function $P_S(x)$ for the signal sample, and $\overline{w_B}$, $\overline{w_S^2}$, and $\overline{w_B^2}$ are defined similarly. The numbers of S and B events N_S and N_B , here and in Equation 2 when used to define $w(x)$, need not be the same as those of the data samples used in the fit (as discussed in section 2.1 point 3).

What concerns us here is that the $\mathcal{F}$ numbers provide a convenient measure of the extent to which $P_S(x)$ differs from a constant, i.e. to which the algorithm has found and fitted a difference in the two distributions. If an increase in n produces a decrease of the $\mathcal{F}$ numbers towards unity, then the extra parameter is doing a useful job. When the numbers are stable this indicates that an adequate polynomial has been used.

5 Several distributions

The method can be extended to the case where there are not just two samples S and B, but three (call them A,B,C). It may be useful to indicate this, although it is not provided in the present program.

One now maximises the likelihood

$$\ln L = \sum_A \ln \frac{A}{A+B+C} + \sum_B \ln \frac{B}{A+B+C} + \sum_C \ln \frac{C}{A+B+C}$$

and parametrises $A/B = e^{\alpha(x)}$, $B/C = e^{\beta(x)}$, $A/C = e^{\gamma(x)}$, where α, β, γ are polynomials as before.

The likelihood becomes

$$\ln L = - \sum_A \ln(1 + e^{-\alpha} + e^{-\gamma}) - \sum_B \ln(1 + e^{\alpha} + e^{-\beta}) - \sum_C \ln(1 + e^{\gamma} + e^{\beta})$$

This preserves the advantages before mentioned, the solutions are physical, the normalisation is correct, and the approach is symmetric in A, B, and C

However, as $(A/B)(B/C) = (A/C)$, we must have $\alpha(x) + \beta(x) = \gamma(x)$, and as this is true for all x it is true for every coefficient; $\alpha_q + \beta_q = \gamma_q$. One of the three polynomials can be replaced explicitly, e.g.

$$\ln L = - \sum_A \ln(1 + e^{-\alpha} + e^{-\alpha-\beta}) - \sum_B \ln(1 + e^{\alpha} + e^{-\beta}) - \sum_C \ln(1 + e^{\alpha+\beta} + e^{\beta})$$

or, if one fears that numerical effects in the solution might bias the answer and lose the symmetry, one can keep them independent and impose the constraint using Lagrangian Multipliers.

This can be extended to many sample; if there are M species there are $M(M-1)/2$ ratios, though only $M-1$ of these are independent.

6 The RATFIT program

A program to execute this algorithm has been written in FORTRAN77 and is called RATFIT (RATio FITting). It is available from the authors. It uses the NAG Library routine [2] E04LAF to find the maximum of the likelihood (Equation 5), for which the required first and second derivatives are given by Equations 6 and 7. The routine F01AAF is used to invert the matrix of second derivatives to give the errors.

The program produces the set of coefficients r_q and their corresponding error matrix. Some practical details in the fitting are as follows

1. One has to avoid exponential overflows when r is large; this can always be done but requires care. For instance, if r is negative for one of the data points then the term as given in Equation 7 should be used for the second derivative, whereas if r is positive the expression

$$\frac{x_i^{q+p} e^{-r(x_i)}}{(1 + e^{-r(x_i)})^2}$$

which is algebraically equivalent, should be used instead.

2. Scaling of the x before fitting makes the problem of finding the maximum easier. Thus all the data is scaled to the interval -1 to 1 before the fit.
3. Rather than using the polynomial expansion

$$r_0 + r_1 x + r_2 x^2 \dots$$

it is preferable to expand in Tschebycheff Polynomials

$$t_0 + t_1 T_1(x) + t_2 T_2(x) \dots$$

and the program uses these internally.

4. In several dimensions the exponential terms take the form:

$$\sum_{q=0}^n \sum_{r=0}^n \sum_{s=0}^n \dots t_{qrs\dots} T_q(x_1) T_r(x_2) T_s(x_3) \dots$$

and the bookkeeping for the sums and products which are to be performed a variable number of times, and coefficients with a variable number of indices, is non-trivial and leads to some rather complicated code.

5. The order n can be varied for each variable and so all terms up to $T_{n_1}(x_1) T_{n_2}(x_2) \dots T_{n_D}(x_D)$ are included. Thus the number of separate terms in the sum, and the number of coefficients is $\prod_{i=1}^D (n_i + 1)$.

6.1 Running the program

The array sizes used in the program are dictated by the parameters `MAXDAT`, `MAXDIM`, `MAXC1`, `MAXCN` which give the maximum number of events allowed, the maximum number of dimensions in the data, the maximum number of coefficients in each singular dimension, and the maximum number of coefficients in the multi-dimensional fit respectively. These parameters are given values in the present program by the statement:

```
PARAMETER (MAXDAT=2000, MAXDIM=10, MAXC1=8, MAXCN=256)
```

Information is passed to the RATFIT program via the FORTRAN COMMON blocks `RFDATA` and `RFINFO`, and the coefficients determined are returned in `RFCOFS` and `RFCOFN`. `RFDATA` is defined as follows:

```
COMMON/RFDATA/NDIM,NS,NB,SIG(MAXDAT,MAXDIM),BAK(MAXDAT,MAXDIM)
```

where:

`SIG(i,j)` (REAL) is the datum in the j^{th} dimension of the i^{th} signal event.

`BAK(i,j)` (REAL) is the datum in the j^{th} dimension of the i^{th} background event.

`NDIM` (INTEGER) is the maximum j filled in the data `SIG` and `BAK`.

`NS` (INTEGER) is the total number of events filled in the array `SIG` (maximum i filled).

`NB` (INTEGER) is the total number of events filled in the array `BAK` (maximum i filled).

The steering parameters are passed via `RFINFO` :

```
COMMON/RFINFO/I1FITS(MAXDIM),INFITS(MAXDIM),LUNCOF,  
+NCOEF1(MAXDIM),NCOEF2(MAXDIM),NFIT,IPRINT,BTOS,EXPLIM,ICOMBC
```

`I1FITS` (INTEGER) specifies which dimensions of the data are to be fitted in 1-dimension. A user can select a dimension to be fitted merely by inserting its index into this array; these are treated in turn until an unfilled (zero) element is found, whereupon the program moves to the multidimensional fit. The default is to fit all dimensions.

`INFITS` (INTEGER) is the array containing the indexes to the dimensions of the data to be fitted in many dimensions. If `ICOMBC` (see below) is set so that coefficients from 1-dimensional fits are used as starting values, then these dimensions are fitted in 1-dimension first, unless they have already been done. By default no multi-dimensional fitting is done.

`LUNCOF` (INTEGER) is the FORTRAN logical unit number to which the indexes for forming the polynomial and coefficients of the multi-dimensional fit will be written (if this parameter is not positive no writing will occur).

`NCOEF1(i)` (INTEGER) is the number of coefficients used to start the 1-dimensional fitting of dimension i . (i.e., a polynomial of order `NCOEF1(i)-1` is used). The default for any i is 2.

NCOEF2(i) (INTEGER) is the number of coefficients used for dimension i in the many dimensional fitting, and the final number of coefficients used in 1-dimensional fits if any are done. The default for any i is 4.

NFIT (INTEGER) is the number of dimensions used in the final fit, i.e., the number of filled elements in the array **INFITS**.

IPRINT (INTEGER) is a flag which if positive turns on printing of the function obtained in a particular iteration of the minimisation. Its default value is zero.

BTOS (REAL) is the expected background to signal ratio in the 'real' experiment, used in forming the $\mathcal{F}$ -numbers. The default ratio is 1.

EXPLIM (REAL) gives the maximum absolute power of the exponential used in the fitting and testing routines of the program. It can be set to prevent numbers exceeding a machine limit. The default is 170.0 which corresponds to a value of about $10^{\pm 73}$. If this gives problems, it probably means that the signal and background distributions are completely separate and distinct (or almost so) and you are trying to fit a step function.

ICOMBC is a flag indicating whether or not to combine the coefficients obtained in 1-dimensional fits to initialise the multi-dimensional fit. The default is 1, which indicates that coefficients will be combined.

Having set up the data in the common blocks, all that is required is the statement

```
CALL RATFIT
```

The polynomial coefficients and their error matrix are then available in the common blocks **RFCOFS** and **RFCOFN** as follows:

```
COMMON/RFCOFS/COEFFS(MAXC1,MAXDIM),ERMATS(MAXC1,MAXC1,MAXDIM)
COMMON/RFCOFN/COEFFN(MAXCN),ERMATN(MAXCN,MAXCN)
DOUBLE PRECISION COEFFN,ERMATN
```

After fitting, the polynomial coefficients from the 1-dimensional fits are contained in **COEFFS** and their corresponding error matrices in **ERMATS**. **COEFFN** contains coefficients from the full fit with error matrix **ERMATN**. The given values have been converted from Tschebycheff to simple polynomial, and the scaling factors removed.

Some notes on the usage of the program are

1. The default starting values for the coefficients assume that the S and B distributions are the same, i.e. all the r_q are zero, except for r_0 which is $\ln(N_S/N_B)$. Although this is the only sensible first guess, it may be a long way from the true solution, and even for 1-dimensional data it may well be that for large n the fit will tend to be slow in converging and/or find false maxima. This should be avoided by the sensible use of the steering parameters in **NCOEF1** and **NCOEF2**: the program will first fit dimension i with a polynomial of **NCOEF1(i)** coefficients, then it will fit with a polynomial one degree higher using the coefficients from the previous fit as its starting point (and zero for the new term), and the process is repeated until **NCOEF2(i)** coefficients are used.

2. Similarly, in a fit of D dimensions, a first guess is provided by fitting each dimension separately. The product of each of these fits then gives the coefficients for the starting point of the general fit. If the distribution function factorises into its different dimensions then the starting point is the true solution; in general it of course does not, but usually does so approximately, so the first guess is not too far from the eventual solution. Such a first guess may be avoided if desired by use of the ICOMBC flag.
3. The 1-dimensional coefficients in RFCOFS are easy to access, whereas for those for several dimensions in RFCOFN the book-keeping is complicated. For this reason they may be written onto a file, in which the powers corresponding to each coefficient are specified. A FORTRAN function RFWGHT is available to reconstruct the weight for a many dimensional combination using this file: The following line obtains the weight W :

$$W = \text{RFWGHT}(\text{LUN}, X, \text{BTOS})$$

where LUN is the FORTRAN Logical Unit Number to read a file output from RATFIT, X is the data point (an array of length NFIT which must have its elements in the same order as was stated in the array INFITS for the fitting), and BTOS is the expected background to signal ratio.

6.2 A simple example

As a simple test to show how the program works, and to compare it with the method of histogramming, 100 'signal' and 100 'background' values of the single variable x were generated, both proportional to the same probability distribution

$$x^2 e^{-15x}$$

The data are displayed in Figure 1. This type of skew distribution is chosen as it typifies many experimental samples.

The coefficients obtained from the RATFIT program, and from histogramming and fitting the ratios are given in the table. (The 'true' values are zero for both coefficients. It should also be noticed that r_0 and r_1 are strongly correlated.)

RATFIT	r_0	r_1
	0.40 ± 0.30	-0.19 ± 1.30

Histogram: no. of bins	r_0	r_1
2	0.56 ± 1.69	-2.14 ± 3.82
7	0.12 ± 1.02	-0.87 ± 3.30
10	0.30 ± 0.79	-1.76 ± 2.62
30	-0.04 ± 0.53	-1.05 ± 1.88
50	-0.84 ± 0.55	-1.11 ± 2.17
100	-0.56 ± 0.49	1.05 ± 2.12

It can be seen that the results from the histogramming technique vary greatly according to the choice of bins, and have errors larger than the RATFIT results.

6.3 A more complicated example

In High Energy Physics one can often form many variables, each of which gives some separation of the signal from the background. As an example of the use of the method in a more complicated situation we here show its application to a possible problem in physics at the LEP accelerator: the separation of a Higgs Boson signal in the channel

$$e^+e^- \rightarrow Z^0 \rightarrow H^0 \mu^+ \mu^-$$

with a Higgs of mass $30 \text{ GeV}/c^2$ which subsequently decays to b quarks. The signature for this is two high energy muons separated from hadron jets, so we select events by requiring two oppositely charged muons each with an energy above 2 GeV (since this is the typical energy required for a muon to traverse a large iron calorimeter before passing through a specialised muon detector).

Background arises from the reaction

$$e^+e^- \rightarrow Z^0 \rightarrow b\bar{b}$$

as these b quarks can also give rise to high energy muons.

The signal data was generated using the program of Behrends and Kleiss [3] and the Lund Monte Carlo program [4] was used to fragment the quarks for both the signal and background samples. To increase the useful Monte Carlo statistics we force the b -quarks in the background to decay to muons.

We use 3 relevant variables: the invariant mass $m_{\mu\mu}$ of the muons and the cosines of the angular separations α_1, α_2 of the muons with the nearest charged particle (ordered so that α_1 is the least angularly separated muon). The distributions of these variables for signal and background are given in Figure 2. The relevant $\mathcal{F}$ -number is $\mathcal{F}_2$ since we know the expected background contamination - we expect about 2,000 such background events per 10^6 Z^0 's (which we scale to 3,000 to (hopefully) overestimate the extra muon pairs we obtain in such a sample due to hadronic faking of muons in a detector), as opposed to about 10 signal events.

Forming the weight $w(x)$ for each variable in a 2-coefficient fit we obtain the following:

Variable	$\mathcal{F}_2$	r_0	r_1
$m_{\mu\mu}$	2.423	-5.286	0.1723
$\cos \alpha_1$	1.430	29.70	-33.92
$\cos \alpha_2$	1.509	100.7	-103.4

If the joint distribution were to factorise it would be the product of these three, which is

$$\exp(125.1 + 0.1723 m_{\mu\mu} - 33.92 \cos \alpha_1 - 103.4 \cos \alpha_2)$$

which gives a separation

$$\mathcal{F}_2 = 1.041$$

After the full 3-dimensional fit the exponential term above becomes

$$\begin{aligned} \exp(& - 1263 + 98.46 \, m_{\mu\mu} + 1200 \, \cos \alpha_1 \\ & - 94.97 \, m_{\mu\mu} \cos \alpha_1 + 1556 \, \cos \alpha_2 - 107.2 \, m_{\mu\mu} \cos \alpha_2 \\ & - 1505 \, \cos \alpha_1 \cos \alpha_2 + 103.9 \, m_{\mu\mu} \cos \alpha_1 \cos \alpha_2) \end{aligned}$$

and we obtain an improvement in separation to

$$\mathcal{F}_2 = 1.010$$

6.4 An example program

```

PROGRAM EXAMPLE
C
  INTEGER MAXDAT,MAXDIM,MAXC1,MAXCN
  PARAMETER (MAXDAT=2000, MAXDIM=10, MAXC1=8, MAXCN=256)
  COMMON/RFDATA/NDIM,NS,NB,SIG(MAXDAT,MAXDIM),BAK(MAXDAT,MAXDIM)
  REAL SIG,BAK
  INTEGER NS,NB,NDIM
  COMMON/RFINFO/I1FITS(MAXDIM),INFITS(MAXDIM),LUNCOF,
+NCOEF1(MAXDIM),NCOEF2(MAXDIM),NFIT,IPRINT,BTOS,EXPLIM,ICOMBC
  REAL COEFFS,BTOS,EXPLIM
  INTEGER NCOEF1,NCOEF2,NFIT,IPRINT,INFITS,I1FITS,LUNCOF
  INTEGER I,J
C Each data point is composed of 5 values
  NDIM=5
C Of these 5, fit dimensions 1,2, and 4 separately
  I1FITS(1)=1
  I1FITS(2)=2
  I1FITS(3)=4
C For dimensions 1 and 2, try a linear fit (default) then a quadratic
  NCOEF2(1)=3
  NCOEF2(2)=3
C For dimension 3, start with a quadratic then try a cubic (default)
  NCOEF1(3)=3
C Then do a joint fit to two of the variables
  NFIT=2
C namely, dimensions 1 and 2
  INFITS(1)=1
  INFITS(2)=2
C but do not use the results from individual fits as starting values
  ICOMBC=0
C For the F numbers, suppose background 1000 times greater than signal
  BTOS=1.E3
C Logical unit number to which coefficients will be written
  LUNCOF=15
C Read in background data
  DO 10 J=1,MAXDAT
    READ(8,END=20) (BAK(J,I),I=1,NDIM)
  10  CONTINUE
  NB=MAXDAT
  GOTO 30
  20  NB=J-1
  30  CONTINUE
C Read in signal data
  DO 40 J=1,MAXDAT
    READ(9,END=50) (SIG(J,I),I=1,NDIM)
  40  CONTINUE
  NS=MAXDAT
  GOTO 60
  50  NS=J-1
  60  CONTINUE
C Execute the fitting
  CALL RATFIT
  END

```

A The errors on the estimates

Consider for simplicity the case of a single parameter, a , whose true value is a_0 . The derivation is similar to that for the usual Maximum Likelihood procedure in the large sample limit.

The estimate $\hat{a}$ from a data sample $(x_1, x_2, \dots, x_N)$ is obtained by setting to zero the quantity

$$f(a) = \sum_i \frac{\partial \ln P(x_i, a)}{\partial a} \quad (10)$$

where we have

$$\begin{aligned} P(x_i) &= \frac{S(x_i)}{S(x_i) + B(x_i)} && \text{for the S sample} \\ &= \frac{B(x_i)}{S(x_i) + B(x_i)} && \text{for the B sample} \end{aligned} \quad (11)$$

Thus $f(\hat{a}) = 0$. At the true value a_0 the value of f is in general non-zero, because the x_i were obtained by sampling at random. It is distributed with some variance $V(f)$. We assume that the overall numbers of S and B events have been chosen *a priori*, and that therefore $\int S dx$ and $\int B dx$ are two fixed constants which sum to 1. This will happen in many cases of this type, for example in the study of events by the Monte Carlo technique one will usually generate some specific number of each type. This means that variation in the final coefficients will arise due to the random distribution of the x_i within each sample, but that there is no component of the variation caused by the possible variation of the total numbers of S and B events being studied, either as absolute numbers or due to binomial variation between the two samples. If the conditions of the experiment are such that the numbers may vary then the errors on the estimates will, naturally, be larger. However for large samples this is probably a small effect.

The variance of $\hat{a}$ is given by the simple law of combination of errors

$$V(\hat{a}) = \frac{V(f)}{\left(\frac{\partial f}{\partial a}\right)^2} \quad (12)$$

The differential in the denominator should be taken at $a = a_0$, which is in fact equal to its expectation value.

$$\left(\frac{\partial f}{\partial a}\right)\Big|_{a=a_0} = \left\langle \frac{\partial f}{\partial a} \right\rangle = N \left\langle \frac{\partial^2 \ln P}{\partial a^2} \right\rangle \quad (13)$$

The numerator can be written out as

$$\begin{aligned} V(f) &= \langle f^2 \rangle - \langle f \rangle^2 \\ &= N \left\langle \left(\frac{\partial \ln P}{\partial a} \right)^2 \right\rangle - N^2 \left\langle \frac{\partial \ln P}{\partial a} \right\rangle^2 \end{aligned} \quad (14)$$

This is then simplified using formulae obtained starting from the obvious statement

$$\int S(x) dx + \int B(x) dx - \int (S(x) + B(x)) dx = 0 \quad (15)$$

Differentiating this with respect to a gives

$$\int S(x) \frac{\partial}{\partial a} \left(\ln \left(\frac{S(x)}{S(x) + B(x)} \right) \right) dx + \int B(x) \frac{\partial}{\partial a} \left(\ln \left(\frac{B(x)}{S(x) + B(x)} \right) \right) dx = 0 \quad (16)$$

This quantity is also, by inspection (referring to Equations 10 and 11) the expectation value of the function f , being the sum of the contributions from the two samples. Thus $\langle f \rangle = 0$ can be inserted in Equation 14. A further differentiation gives

$$\begin{aligned} & \int S(x) \frac{\partial^2}{\partial a^2} \left(\ln \left(\frac{S(x)}{S(x) + B(x)} \right) \right) dx + \int B(x) \frac{\partial^2}{\partial a^2} \left(\ln \left(\frac{B(x)}{S(x) + B(x)} \right) \right) dx + \\ & \int \frac{\partial S(x)}{\partial a} \frac{\partial}{\partial a} \left(\ln \left(\frac{S(x)}{S(x) + B(x)} \right) \right) dx + \int \frac{\partial B(x)}{\partial a} \frac{\partial}{\partial a} \left(\ln \left(\frac{B(x)}{S(x) + B(x)} \right) \right) dx = 0 \end{aligned}$$

The sum of the first pair of integrals is the expectation value of the second derivative of P . The second pair, with a little manipulation, can be written as

$$\int S \left(\frac{S'B - B'S}{S(S+B)} \right)^2 dx + \int B \left(\frac{S'B - B'S}{B(S+B)} \right)^2 dx \quad (17)$$

which is the expectation $\langle \left(\frac{\partial \ln P}{\partial a} \right)^2 \rangle$. Thus we have

$$\left\langle \frac{\partial^2 \ln P}{\partial a^2} \right\rangle = - \left\langle \left(\frac{\partial \ln P}{\partial a} \right)^2 \right\rangle \quad (18)$$

Equation 18 can be inserted in Equation 14, and the result used in the numerator of Equation 12, with Equation 13 in the denominator, to give the desired result

$$V(\hat{a}) = - \left\langle N \frac{\partial^2 \ln P}{\partial a^2} \right\rangle^{-1} \quad (19)$$

In an actual application this is of course unknown, but the value for the observed sample at $a = \hat{a}$ can be used within the validity of the approximation.

$$V(\hat{a}) = - \left(\frac{\partial^2 \sum \ln P}{\partial a^2} \right)^{-1} \quad (20)$$

References

- [1] R.J. Barlow *Event Classification using Weighting Methods* Journal of Computational Physics **72** (1987) p 202
- [2] NAGLIB version 11. Supplied by the Numerical Algorithm Group, Oxford, UK.
- [3] F.A.Behrends and R.Kleiss *Initial State Radiation at LEP Energies and the Corrections to Higgs Boson Production* Nuclear Phys. **B260** (1985) p 32
- [4] T.Sjöstrand *The Lund Monte Carlo for Jet Fragmentation* Computer Phys. Comm. **39** (1986) p 347

Figure 1: Data for signal and background, both from the same random distribution.

Figure 2: Histograms of data for signal (a) and background (b) in the 3 variables.