A Poisson Parable: Bias in Linear Least Squares Estimation

Wm. A. Wheaton

ABSTRACT A standard problem in high-energy astronomy data analysis is the decomposition of a set of \( I \) observed counts, \( n_i \), described by Poisson statistics, for \( i = 1, \ldots, I \), according to some known \( J \)-component linear model,

\[
\bar{n}_i = \text{E}[n_i] = \sum_{j=1}^{J} A_{ij} r_j,
\]

with underlying physical count rates \( r_j \) or fluxes which are to be estimated from the data, the \( A_{ij} \) being known experiment constants. This problem is often solved by Linear Least Squares (LLSQ), but limited to situations where the number of counts per bin \( i \) is not too small.

For the simplest possible case, \( J = 1 \), which is just a counting experiment with no background, it is interesting to attempt a direct application of the weighted average formula using \( n_i \), instead of the observed \( n_i \), the expected count, \( \text{E}[n_i] = \sigma_i^2 = r t_i \) in the weighting, where \( t_i \) is the observing time in bin \( i \), it turns out that the unknown rate \( r \) cancels from the weighted average sums, and we recover the obviously correct estimate \( \hat{r} = N/T = \sum n_i / \sum t_i \).

41.1 Introduction

The problem of estimation from Poisson data has been solved by variations of linear least squares (LLSQ, often termed the “minimum chi-square method” or something similar), for many years in nuclear physics, high energy physics, and high-energy astronomy. In this paper I wish to draw attention once again to the problems caused by use of the observed counts to directly weight the LLSQ equations. I do so by examining the simplest possible Poisson estimation problem, and show that directly weighting each bin with \( 1/\sigma_i^2 \) from the observed data \( n_i \) gives a manifestly wrong result.
41.2 A paradox

Thus, consider the problem of estimating a single count rate without any background at all, when the data have been binned into \( I \) bins, \( i = 1, \ldots, I \), each with \( n_i \) counts observed in livetime \( t_i = t_i = \text{total bin time - deadtime} \). It is known that \( N = \sum n_i \) and \( T = \sum t_i \) are “sufficient statistics” ([Leh59], pp 17–20) for this problem. That is, the maximally efficient estimator for the true rate \( r \) is a function of \( N \) and \( T \) only, so that the extra information due to the binning is superfluous. Nevertheless it is interesting to compare algorithms for handling binned data in this simple situation, for which equation (41.1) reduces to

\[
\hat{n}_i = t_i r. \tag{41.2}
\]

41.2.1 Weighted averaging

Consider first the weighted average of the count rate estimates for each bin, \( \hat{r}_i = n_i / t_i \), with weights \( w_i = 1 / \sigma_r^2 \). This is a plausible approach, since it is known that the weighted average, with weights \( 1 / \sigma_r^2 \), is the optimal (minimum variance) average. For a single sample, \( \sigma = \hat{r} / \sqrt{N} \). Thus we take the weights to be \( w_i = t_i / n_i \). The weighted average formula gives

\[
\hat{r} = \frac{\sum t_i \hat{r}_i}{\sum w_i} = \frac{\sum t_i}{\sum (t_i / n_i)}, \tag{41.3}
\]

which looks somewhat strange.

41.2.2 “Modified chi-square” method

What [EDJ+71] term the “modified \( \chi^2 \) method” of LLSQ is the minimization of

\[
\chi^2 \approx \sum_{i=1}^{I} (n_i - m_i)^2 / \sigma_i^2, \tag{41.4}
\]

where \( m_i \) are the fitted or model counts in bin \( i \), using the approximation that \( \sigma_i = \sqrt{n_i} \), the observed data. Considering equations (41.2) as an \( I \times J \) LLSQ system in the trivial case where \( J = 1 \), with one equation for each bin, we begin by weighting each equation by \( 1 / \sigma_i \approx 1 / \sqrt{n_i} \). Thus, after multiplication of both sides by the transpose of the weighted matrix (bringing in another factor of \( 1 / \sigma_i \), note), we obtain:

\[
\sum t_i = \sum \left\{ \frac{t_i^2}{n_i} \right\} r. \tag{41.5}
\]

and obtain for our estimate of \( r \):

\[
\hat{r} = \frac{\sum t_i}{\sum (t_i^2 / n_i)}. \tag{41.6}
\]