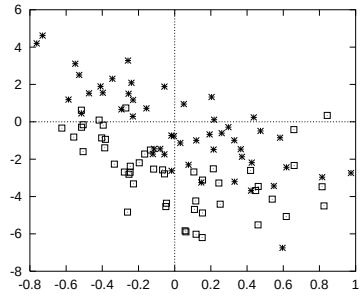

Figure 1: Toy data from two populations differing by an offset `dy`

1 Inferring the distance to the Virgo cluster

Introduction

The most accurate means of measuring the distances to galaxies involves the measurement of the period and luminosity of a class of supergiant variable stars known as Cepheids. Empirically, the luminosity and log period of nearby Cepheids show a linear relationship (with a small amount of scatter), so the measurement of the magnitude and period of a number of Cepheids in a distant galaxy give direct distance information, in the form of a constant offset between their magnitude–period line and the magnitude–period line of the nearby Cepheids.

In Nature vol 371 p.757-762, Freedman *et. al.* report such a measurement of the distance to the Virgo cluster galaxy M100, and a deduction of the value of the Hubble constant. The details of their data analysis are not given there, but it is presumed that a least-squares fit of a straight line model is used. This data fitting procedure corresponds to a model that assumes uniform Gaussian scatter from an exact straight line relationship. However, one might argue that these assumptions are inappropriately strong. It is not known that an exact straight line relationship holds, and nor do we know that the scatter is Gaussian and uniform. A model that made less strong assumptions might yield different error bars on the inferred offset. Larger error bars might help reduce the controversy over the value of the Hubble constant.

We have not yet received the original data from Freedman *et. al.*, so here we use toy data (figure 1) to compare the inferred offset between two scatterplots when (a) a straight line assumption is replaced by a polynomial of degree K; (b) a uniform Gaussian noise level is replaced by a noise level that varies as a function of x.

Method

Data in the form of (x, t) pairs are assumed to be obtained from two populations. The underlying relationship is described by y, x pairs, where $y(x) = \sum_h w_{h=1}^{K_W} \phi_h(x)$, and the basis functions ϕ are Legendre polynomials. Similarly, the varying noise level is written as $\tau(x) = \exp\left(\sum_{h=1}^{K_B} b_h \phi_h(x)\right)$. The parameters of the model are thus $\{w_h\}_1^{K_W}$ and $\{b_h\}_1^{K_B}$. Gaussian priors on these parameters are specified with single hyperparameters α_w and α_b that have broad gamma priors. For convenience I introduce $K = \max(K_W, K_B)$ and if $K_B < K$ I introduce extra parameters $\{b_h\}_{K_B+1}^{K}$ that are set to zero (this allows the use of a single matrix $\phi_h(x_n)$). The data from the two populations are stored in vectors $\mathbf{x}$ and $\mathbf{t}$ such that population 'a' gave rise to $\{(x_n, t_n)\}_{n=1}^{N_a}$, and population 'b' gave rise to $\{(x_n, t_n)\}_{n=N_a+1}^{N}$. The offset between the two populations, the quantity of interest, is termed `dy`.

We obtain the special case of uniform noise by setting $K_B = 1$ and the special case of a straight line relationship by setting $K_W = 2$.

Here is the file `astro10.bug`:

```
model astro10;

const
  N=100,        # Number of data
  Na=50,        # Number of data in first population
  KW=10,        # Number of basis functions for y(x)    [alternative: 2]
  KB=10,        # Number of basis functions for tau(x)  [alternative: 1]
  K=10,         # Please ensure K=MAX(KW,KB)
  x0=0.0,dx=1.0;

var
      x[N], dy , y[N], t[N],  w[K], b[K],
      phi[K,N],
      alphab, alphaw, tau[N],sigma[N];

data  in "astro2.dat" ;
inits in "astro1.in"  ;

{
  dy ~ dnorm( 0.0, 0.0001 ) ;  # offset between two populations

  for (m in 1:N) {   # recurrence relation to define Legendre polynomials
      phi[1,m] <- 1.0 ;
      phi[2,m] <- (x[m]-x0)/dx ;
```

```
    for (h in 3:K) {
          phi[h,m] <- ( ( 2 * h - 3 ) * (x[m]-x0)/dx * phi[h-1,m]
                      - ( h - 2 ) * phi[h-2,m] ) / ( h - 1 ) ;
    }
  }
  for (h in 1:KW) {
    w[h] ~ dnorm( 0.0 , alphaw ) ;
  }
  for (h in 1:KB) {
    b[h] ~ dnorm( 0.0 , alphab ) ;
  }
  for (h in KW+1:K) {
    w[h] <- 0.0 ;
  }
  for (h in KB+1:K) {
    b[h] <- 0.0 ;
  }

  for (m in 1:N) {
    t[m]       ~  dnorm( y[m] , tau[m] ) ;
    tau[m]   <- exp( inprod( b[] , phi[,m] ) ) ;
    sigma[m] <- 1.0 / sqrt( tau[m] ) ;
  }
  for (m in 1:Na) {
    y[m]        <- inprod( w[] , phi[,m] ) ;
  }
  for (m in Na+1:N) {
    y[m]        <- dy + inprod( w[] , phi[,m] ) ;
  }

  alphab  ~ dgamma( 1.0E-3 , 1.0E-3 ) ;
  alphaw  ~ dgamma( 1.0E-3 , 1.0E-3 ) ;
}
```

And here is the file `astro1.in`:

```
      list ( alphab=1.0 , alphaw=1.0 , dy=0 )
```

Results

The toy data was generated using true values $K_B = K_W = 3$ and `dy` $= 3.0$.

A burn–in period of 500 iterations was followed by 1000 iterations with statistics on `dy` being recorded.

dy K_B	1			10		
K_W	mean	sd	2.5%:97.5%	mean	sd	2.5%:97.5%
2	2.442	0.3005	1.857:3.020	2.761	0.2603	2.250:3.240
10	2.542	0.3074	1.983:3.156	2.714	0.2710	2.180:3.231

The over–simple model $K_B = 1, K_W = 2$ gives a 95% confidence interval for `dy` that only just includes the true value. Changing from the over–simple model to the model with more parameters in the interpolant (moving down from the top left corner) produces a slight increase in the uncertainty of `dy`. The increase is only slight because there are two opposing effects: first, for any particular value of noise level $1/\tau$, the more flexible interpolant is less well determined and the uncertainty in `dy` increases; but second, the greater flexibility of the interpolant allows it to fit the curving shape of the data and makes smaller noise levels $1/\tau$ probable. Small noise levels give more accurate inferences. A similar effect occurs as we increase the number of terms in the representation of $\tau(x)$ (going from left to right). The estimation of `dy` can become more precise, in intuitive terms, because the model is able to discover that some values of x give more reliable measurements than others, so that the inference of `dy` can be based on them, ignoring the more noisy measurements. The net effect is that when we change from the over–simple model to the most flexible model (bottom right), the confidence interval becomes smaller and more accurate. Whether this will happen for the real Cepheid data remains to be seen.

DJCM December 9, 1994