Some facts about Gaussian linear models

Michael Wilensky

February 2025

1 Introduction

The purpose of this document is to expose some basic facts about Gaussian linear models.

2 Definitions

Suppose we have some data, denoted with the vector d, and a linear model for the data with Gaussian noise,

$$d = Ax + n, \tag{1}$$

where n is a zero-mean Gaussian noise vector, x is a vector holding the parameters of the linear model (the coefficients), and A is a matrix holding the predictors (basis functions). Given x, A, and the noise covariance, N, and some prior information $\mathcal{I}$, we have

$$d|x, A, N, \mathcal{I} \sim \mathcal{N}(Ax, N) \tag{2}$$

i.e. the data would be normally distributed about a mean Ax with covariance N.

Generally we are interested in the case where x is unknown, which is related to $d|a, A, N, \mathcal{I}$ via Bayes' theorem:

$$P(x|d, A, N, \mathcal{I}) = \frac{P(d|x, A, N, \mathcal{I})P(x|A, N, \mathcal{I}))}{P(d|A, N, \mathcal{I})}. \tag{3}$$

For the remainder of this document, we will take A, N, and $\mathcal{I}$ as fixed prior information, and omit conditioning on them in the notation.

Equation 3 implies a need for a prior distribution, $P(x)$. In theory, this prior can be anything, but in practice it encodes a particular state of knowledge and therefore the choice of prior has an effect on the outcome of the inference. We will focus on Gaussian priors, eschewing discussions of appropriateness to other works. That is to say, *a priori*,

$$x \sim \mathcal{N}(\mu, C) \tag{4}$$

3 Properties of the Posterior

Inferences about x given d are captured by the probability distribution of $x|d$. First, what is its form? By multiplying $P(d|x)$ and $P(x)$, which are two Gaussian densities, we can see by examining the exponent that the resulting density must be Gaussian. What are its parameters? The terms in the exponent are, for real vectors,[1]

$$-\frac{1}{2}\left((d-Ax)^T N^{-1}(d-Ax)+(x-\mu)^T C^{-1}(x-\mu)\right)= \\ -\frac{1}{2}\left(x^T\left(A^T N^{-1}A+C^{-1}\right)x+a(x)+b\right) \quad (5)$$

where $a(x)$ is a linear function of x, and b is a constant. Their exact forms are not important because all we need to do is figure out what the covariance and mean of our new Gaussian are, which can be determined through a variety of methods, some more painful than others. Importantly, we can read off the posterior covariance, C', from Equation 5:

$$C'^{-1} = A^T N^{-1} A + C^{-1}. \quad (6)$$

To get the mean, we take the gradient with respect to x of the left side of Equation 5 to obtain linear equation that defines the location of the extremum (equivalent to the mean). I'm going to skip over formalities, but in general, one should break this down component-wise and verify the following relationship:

$$C'^{-1}\mu' = A^T N^{-1} d + C^{-1}\mu \quad (7)$$

where μ' is the posterior mean. This gives us everything we need to know analytically about the posterior. If this is the extent of the model, then the problem has a fully analytic solution and this is the end of the journey up to explorations of particular instances of the quantities involved. However, often Gaussian linear models appear in larger hierarchical models as conditional distributions of joint posteriors that are not necessarily easily understood analytically. In this case it is useful to know how to generate samples from Gaussian distributions, as part of e.g. a Gibbs sampler.

4 Generating Gaussian samples

In general, if μ' and C' are already calculated, one may draw a standard normal sample, ω, and form a sample, s, via

$$s = \mu' + C'^{1/2}\omega \quad (8)$$

where $C'^{1/2}$ is any matrix, M, satisfying $MM^T = C'$. A common choice is the Cholesky decomposition for its numerical properties but theoretically this is not strictly necessary.

[1] which is a space we can always choose to work in when faced with complex vectors – not sure what happens with e.g. quaternions

In typical sampling applications, C and N are easily invertible, but C'^{-1} is not easily invertible to obtain C. Furthermore, it is also typical in Gibbs sampling applications for A or C to be a function of other model parameters not included in x, meaning an explicit inversion of C'^{-1} may have to happen for every new sample to use Equation 8 naively. This is often slow and numerically unpredictable compared to various alternatives. More often, we take advantage of the fact that C'^{-1} is easily calculated and instead use a fast linear solver[2] to instead solve a linear equation for s. For example, one could solve

$$C'^{-1}s = C'^{-1}\mu' + C'^{-1/2}\omega, \tag{9}$$

which can be written

$$\left(A^T N^{-1} A + C^{-1}\right) s = A^T N^{-1} d + C^{-1}\mu + C'^{-1/2}\omega. \tag{10}$$

Now, we can play one more trick with the fluctuation term (the one involving ω), to get an equivalent set of samples. In particular, we can instead solve

$$\left(A^T N^{-1} A + C^{-1}\right) s = A^T N^{-1} d + C^{-1}\mu + A^T N^{-1/2}\omega_0 + C^{-1/2}\omega_1 \tag{11}$$

where ω_0 and ω_1 are standard normal random vectors. This is usually preferable because, for example, recalculating $A^T N^{-1/2}$ for each sample where A changes is generally easier than recalculating $\left(A^T N^{-1} A + C^{-1}\right)^{1/2}$. To see why this works, first see that

$$C'^{-1}\langle s\rangle = C'^{-1}\mu' \tag{12}$$

i.e.

$$\langle s\rangle = \mu'. \tag{13}$$

where this expectation is over *realizations of* ω_0 *and* ω_1. Then see that the fluctuation terms obey

$$\mathrm{Cov}\left(A^T N^{-1/2}\omega_0 + C^{-1/2}\omega_1, A^T N^{-1/2}\omega_0 + C^{-1/2}\omega_1\right) = C'^{-1} \tag{14}$$

implying

$$\mathrm{Cov}\left(C'^{-1}s, C'^{-1}s\right) = C'^{-1}\mathrm{Cov}\left(s,s\right)C'^{-1} = C'^{-1} \tag{15}$$

further implying

$$\mathrm{Cov}\left(s,s\right) = C'. \tag{16}$$

In some instances, further speedups can be made if $C'-1$ is sparse, or can be made to be sparse with preconditioning (rumor is that this is usually done by finding an approximate inverse that is easy to compute or does not need to be recomputed ever).

[2]for example, `NUMPY`'s linear algebra package wraps `LAPACK`, which is a well-understood high-performing `FORTRAN`-based linear algebra package

5 Model selection

Consider two linear models, A_1 and A_2. So long as we have proper priors for the parameters of these models, we can compare the marginal likelihoods of the models fully analytically. If we are willing to assign prior probabilities to the models, we can perform Bayesian model selection. In particular, omitting a bunch of things for brevity, and defining for the ith model

$$C_i' \equiv N + A_i C_i A_i^T \tag{17}$$

where C_i is the prior covariance for the ith model. Let μ_i be the prior mean for the ith model. Then define

$$\chi_i^2 \equiv (d - A_i\mu_i)^T C_i'^{-1} (d - A_i\mu_i) . \tag{18}$$

This lets us see that

$$\log \frac{P(A_1|d)}{P(A_2|d)} = -\frac{1}{2}\left(\log \frac{|C_1'|}{|C_2'|} + \chi_1^2 - \chi_2^2\right) . \tag{19}$$

Roughly speaking, this will favor model 1 if it has smaller volume and smaller squared deviations with respect to the prior parameters μ_1 and C_1 compared to the equivalent quantities involving model 2. To investigate further, we experiment numerically with the above formalism.

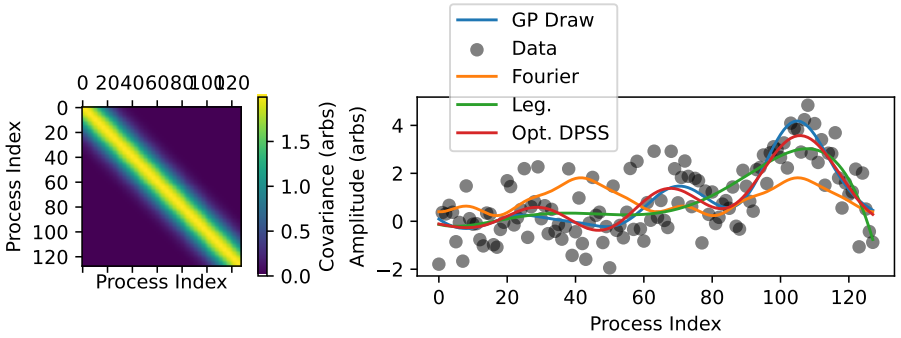

Figure 1: Left: Covariance matrix of a stationary Gaussian process with Gaussian covariance function. The correlation length is set to 5 samples. Right: A draw from a Gaussian process (blue) with added white noise (black data), and fits using three different bases with 6 basis functions each: Fourier (orange), Legendre (Green), and a tailored Discrete Porlate Spheroidal Sequence based on what maximizes the marginal likelihood (red). The tailored basis does the best.

Figure 1 shows the result of fitting a smooth Gaussian process in the presence of white noise, with an SNR per "time" sample of roughly $\sqrt{2}$. We then experiment with least-squares fitting the series to different choices of basis functions, some of which are very commonly used.

1. Fourier modes
2. Legendre
3. Discrete Prolate Spheroidal Sequence (DPSS)

For the legendre and DPSS bases, we use six basis functions. For the Fourier basis, we make sure these are centered on the DC mode and use seven basis functions to ensure symmetry in the Fourier domain. The DPSS basis has a tunable parameter, which is the bandwidth of the signal. We have chosen the bandwidth that maximizes the marginal likelihood of the data for fixed number of basis functions (set to six). By eye, the closest approximation to the actual Gaussian draw is this tuned DPSS basis.

The marginal likelihoods reward good fits (low χ^2 in Equation 19), and penalize excessive prior volume. This penalty is related to the determinant terms in 19 in two different ways. First, an excessive prior width at fixed number of model parameters is discouraged, and second, extra parameters are penalized *exponentially* in certain cases. This second point assumes a couple things: that each parameter has access to the same amount of parameter space by assuming $C \propto I$, and that the prior covariance is much broader than the (white) noise covariance. In this case, we can write for Equation 17,

$$C_i' = \sigma_n^2 I + \sigma_p^2 A_i A_i^T \tag{20}$$

The matrix $M \equiv A_i A_i^T$ has some number of nonzero eigenvalues equal to its rank, which is the number of linearly independent basis functions (not necessarily orthogonal!). Denote the jth eigenvalue of this matrix (including any zeros) as $\lambda_i^{(j)}$. The eigenvalues C_i' are then in one-to-one correspondence[3] with these and are equal to $\sigma_p^2 \lambda_i^{(j)} + \sigma_n^2$. We the have that

$$\log |C_i'| = \sum_j \log \left(\sigma_p^2 \lambda_i^{(j)} + \sigma_n^2 \right) \tag{21}$$

Suppose we rank order these, and for some cutoff, J, all of the $\lambda_i^{(j)}$ are 0. If M is the length of the data, then this gives

$$\log |C_i'| = (M - J) \log(\sigma_n^2) + \sum_{j=1}^{J} \log \left(\sigma_p^2 \lambda_i^{(j)} + \sigma_n^2 \right) \tag{22}$$

If A_i is orthogonal, we can see something nice (the nonzero eigenvalues will all be 1), but we'll show numerically that this roughly holds anyway:

$$\log |C_i'| = (M-J) \log(\sigma_n^2) + J \log(\sigma_p^2 + \sigma_n^2) = M \log(\sigma_n^2) + J \log \left(1 + \frac{\sigma_p^2}{\sigma_n^2} \right) \tag{23}$$

[3] Any linear combination of vectors with degenerate eigenvalue can also make a new eigenvector with that eigenvalue, including in the nullspace.

The leftmost term is a constant of the problem because it is defined in terms of the noise variance, assumed known, and the length of the series in consideration. The right-hand term is clearly proportional to the number of basis vectors i.e. the size of the parameter space. The overall minus sign in Equation 19 tells you that this is a *penalty* for the model defined by A_i. So the penalty is *exponential* in the size of the model.

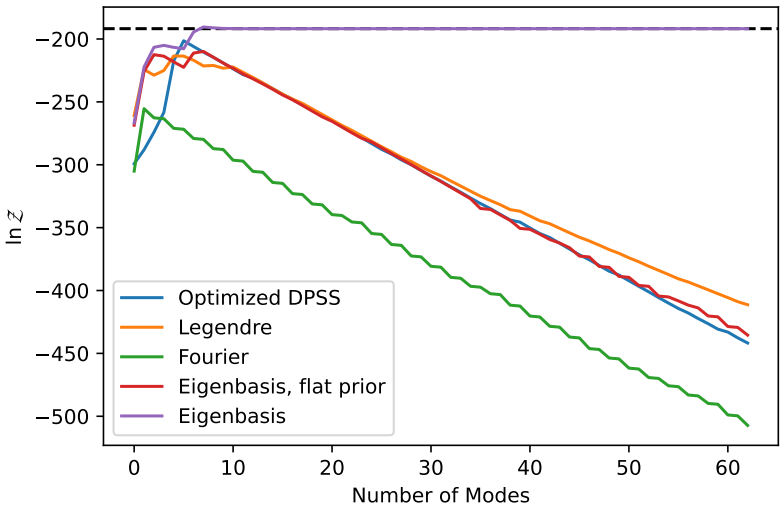

Figure 2: Marginal likelihoods for different choices of bases based on totaly number of modes included. The first four bases (blue, orange, green) assume an approximately flat prior. These bases (with ∼flat priors!) have exponential tails, and peak within a handful of basis functions. The optimized DPSS basis with 6 basis functions wins among these four. The fourth (red) is the eigenbasis of the signal covariance, but with a flat prior. The horizontal dashed line is the marginal likelihood using the true covariance of the process. The purple line uses the eigenbasis of the true signal covariance, and the prior covariance for the purple line is diagonal with variances equal to the eigenvalues. In other words, succeeding approximations to the true covariance approach the evidence of the true covariance and roughly plateau.

For an illustration of this concept, see Figure 2. Here we use an approximately flat prior for the three bases enumerated above and vary the number of basis functions. We also do this for a fourth basis set, defined by the eigenvectors of the covariance matrix in Figure 1. In one case we use an approximately flat prior to show that it behaves very similarly to the other bases. In another case we use the eigenvalues to generate successive approximations of the signal covariance. The flat prior cases all have exponential tails, and the successive approximations of the signal covariance tend to the value used when the true

signal covariance is input into the marginal likelihood calculation. The Fourier basis has a stair step pattern, which I believe is attributed to the fact that it only includes one of the positive/negative frequency exponentials per increment in the number of basis functions, but I have not tested this. The tuned DPSS basis at 6 basis functions outperforms all the other flat prior choices. Note that this logarithmic scale is extremely brutal, with the magnitude of overperformance being roughly 10 e-folds, which is itself roughly $2^{15} \approx 32000$. This can be interpreted as the posterior odds in favor of the best DPSS basis against the nearest competitor, assuming both models were equally likely *a priori*.