Exercises 4: Backfitting, Gibbs sampling

Additive models and backfitting

Consider a multiple-regression problem with outcomes y_i and predictors $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$. An *additive model* takes the form

$$y = \alpha + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p) + \epsilon$$

for general functions f_j, where $E(\epsilon) = 0$ and $\text{var}(\epsilon) = \sigma^2$. Each individual effect can be nonlinear. But just as in linear regression, the effects from each predictor still add together to give the joint effect. The f_k are sometimes called partial response functions.

Suppose that someone hands you a set of good estimates for all f_j, $j \neq k$. Define the kth partial residual as the vector $y^{(k)}$ having elements

$$y_i^{(k)} = y_i - \alpha - \sum_{j \neq k} f_j(x_{ij}) \, .$$

Then we can clearly get a decent estimate for f_k by fitting $y^{(k)}$ versus x_k using the tools already in our kit (e.g. local linear regression).

To fit an additive model by *backfitting*, begin with an initial guess for all the f_j's. Successively refine your estimate for each partial response function f_k by computing the partial residuals $y^{(k)}$, and regressing these on x_k. Stop when the estimates reach some convergence criterion.[1]

The data in `air.csv` contain daily readings on some air-quality measurements for the New York area.[2]

Ozone: Average atmospheric ozone concentration in parts per billion from 1 PM to 3 PM at Roosevelt Island.

Solar.R: Solar radiation in Langleys in the wavelength range 400–770 nanometers from 8 AM to 12 PM at Central Park.

Wind: Average wind speed in miles per hour between 7 AM and 10 AM at LaGuardia Airport

Temp: Maximum daily temperature in degrees F at La Guardia Airport.

Write an R function to fit an additive model, and use it to regress ozone concentration on the other three variables.

At least two issues will need your attention:

1. If you subtract a constant c from f_j, and add that same constant to some other f_k, you will get the same regression function for all values of x. You will therefore need some way to identify them.

2. Should all dimensions have the same smoothing parameter?

[1] It is not obvious, at least to me, that this process converges to a unique solution when the predictor variables are correlated. But it does, for essentially the same reason (and under the same kinds of conditions) that the Gauss–Seidel algorithm works for solving linear systems.

[2] You'll notice that there are some missing days in there; we'll assume that these are missing at random.

Gibbs sampling

Consider a Bayesian analysis of the multiple linear-regression model, where $y = X\beta + \epsilon$. Suppose that the errors are assumed to be i.i.d. Gaussian, $\epsilon \sim \mathrm{N}(0, \sigma^2 I)$. Suppose that we specify an inverse-gamma prior for σ^2, and a simple hierarchical model for the regression coefficients:

$$\begin{aligned} (\beta \mid \tau^2) &\sim \mathrm{N}(0, \tau^2 I) \\ \sigma^2 &\sim IG(a/2, b/2) \\ \tau^2 &\sim IG(c/2, d/2) \end{aligned}$$

for fixed choices of a, b, c, d. Remember that an inverse-gamma prior for a variance v means that $1/v$ has a Gamma prior. It is most convenient to parametrize the Gamma distribution in terms of its shape and rate (not the scale). Thus if $1/v = r \sim \mathrm{Ga}(a, b)$, then $p(r) \propto r^{a-1}e^{-br}$.

(A) Derive the conditional posterior distributions for each model parameter: $p(\beta \mid y, \sigma^2, \tau^2)$; $p(\sigma^2 \mid y, \beta, \tau^2)$; and $p(\tau^2 \mid y, \beta, \sigma^2)$. Note that $p(\beta \mid y, \sigma^2, \tau^2)$ is actually a conditional distribution for the entire block of regression coefficients, rather than each coefficient individually.

(B) *Gibbs sampling*[3] is like a Bayesian version of backfitting: iteratively take a random draw from each parameter's conditional distribution, given the current values of all other parameters. Of course, unlike in backfitting, the draws will never converge to specific values as you run the algorithm for more iterations. Rather, they will build up a Monte Carlo sample from the joint posterior distribution over all parameters.[4]

Load the `diabetes` data set in the `BayesBridge` R package, available from CRAN. This is stored as a list, so to extract the responses and design matrix, you can use commands such as

```
Xd = diabetes$x
yd = diabetes$y
```

The outcome variable is a serum-insulin measurement in diabetes patients. The predictors are the patient's age, sex, BMI, and various other blood measurements. The x matrix has been standardized to have zero mean and unit ℓ^2 norm in each column. Fit a Bayesian linear model via Gibbs sampling to serum insulin versus the other predictors. Start with default values for the hyperparameters on σ^2 and τ^2 of $a, b, c, d = 1$. Remember to center the outcome, or to include a column in your design matrix for an intercept term.

The following two papers might be interesting if you want some more background on choosing priors for variances in hierarchical models: (1) "Prior distributions for variance parameters in hierarchical models," by Gelman (Bayesian Analysis, 2006); and (2) "On the half-Cauchy prior for a global scale parameter," by Polson and Scott (Bayesian Analysis, 2012). Start with the first paper and only bother with the second if you really want to dig deeper here. These should be easy to find on the web.

[3] After Josiah Willard Gibbs, the father of modern thermodynamics. Why it is so named is a story for another day.

[4] This is even less obvious than the fact that backfitting converges. Formally, this process defines a Markov chain whose state space is the parameter space, and whose stationary distribution is (under suitable regularity conditions) the joint posterior. Gibbs sampling is a special case of Markov-chain Monte Carlo methods. A nice reference is *Monte Carlo Statistical Methods*, by Robert and Casella. An even better one is Peter Müller's course here at UT on Monte Carlo methods.