ISYE-6420 Midterm Exam

Marshall Bockrath-Vandegrift

March 20, 2016

Contents

1 Master of light

1 (a) One-way Bayesian ANOVA

We will model each measurement as $X \sim \mathcal{N}(\mu_0 + \alpha_{\text{type}}, \sigma^2)$, with improper flat priors across the support of all latent parameters and ensuring identifiability by constraining the α parameters to sum to zero. We implement this model in Stan,

```
data {
  int<lower=1> N;
  int<lower=1> NTYPE;
  int<lower=1, upper=NTYPE> type[N];
  vector[N] meas;
}
parameters {
  vector[NTYPE-1] alpha_raw;
  real mu0;
  real<lower=0> sigma;
}
transformed parameters {
  vector[NTYPE] alpha;
  for (i in 1:(NTYPE-1)) alpha[i] <- alpha_raw[i];
  alpha[NTYPE] <- -sum(alpha_raw);
}
model {
  meas ~ normal(mu0 + alpha[type], sigma);
}
generated quantities {
  real diff[NTYPE, NTYPE];
  for (i in 1:NTYPE)
    for (j in 1:NTYPE)
      diff[i, j] <- alpha[i] - alpha[j];
}
```

We generate samples of the α parameters from this model, yielding the following summary:

```
Inference for Stan model: p1a-anova-1way.
4 chains, each with iter=49000; warmup=24500; thin=1;
post-warmup draws per chain=24500, total post-warmup draws=98000.

          mean se_mean   sd   2.5%    25%   50%   75% 97.5% n_eff Rhat
alpha[1] 15.26    0.02 4.11   7.16  12.51 15.26 18.03 23.28 63321    1
alpha[2] -0.91    0.01 3.77  -8.36  -3.44 -0.92  1.61  6.49 65039    1
```

```
alpha[3] -1.45    0.01 2.89  -7.11  -3.38 -1.47  0.48  4.23 63511    1
alpha[4] -4.29    0.01 3.68 -11.51  -6.76 -4.31 -1.82  2.98 65455    1
alpha[5] -8.61    0.01 3.96 -16.41 -11.23 -8.60 -5.97 -0.84 98000    1

Samples were drawn using NUTS(diag_e) at Fri Mar 18 08:56:52 2016.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

We also plot the 95% equi-tailed credible sets for the α parameters (figure 1). As seen in both the numeric summary and the plot, it appears that the treatment effects `alpha[1]` ("Glass/8") and `alpha[5]` ("Steel/12") differ statistically from the overall mean measurements.

1 (b) One-way treatment comparisons

Samples of the differences between treatment effects (`diff` in the Stan model) yield the following summary:

```
Inference for Stan model: p1a-anova-1way.
4 chains, each with iter=49000; warmup=24500; thin=1;
post-warmup draws per chain=24500, total post-warmup draws=98000.

           mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
diff[1,2] 16.17    0.03 6.35  3.65 11.94 16.15 20.43 28.65 60502    1
diff[1,3] 16.71    0.02 5.54  5.82 13.00 16.72 20.44 27.55 59758    1
diff[1,4] 19.55    0.03 6.27  7.19 15.35 19.55 23.76 31.85 61063    1
diff[1,5] 23.87    0.02 6.54 10.97 19.52 23.85 28.26 36.73 98000    1
diff[2,3]  0.54    0.02 5.13 -9.56 -2.88  0.56  3.95 10.62 63847    1
diff[2,4]  3.38    0.02 5.89 -8.26 -0.57  3.40  7.35 14.89 60300    1
diff[2,5]  7.69    0.02 6.20 -4.51  3.55  7.70 11.84 19.89 98000    1
diff[3,4]  2.84    0.02 5.00 -7.04 -0.51  2.85  6.18 12.67 62641    1
diff[3,5]  7.15    0.02 5.33 -3.33  3.61  7.15 10.69 17.72 98000    1
diff[4,5]  4.32    0.02 6.12 -7.70  0.24  4.31  8.40 16.35 98000    1

Samples were drawn using NUTS(diag_e) at Fri Mar 18 08:56:52 2016.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

We also plot the 95% equi-tailed credible sets for these differences (figure 2). Both the plot and numeric summary suggest that only the "Glass/8" mirror type showed a significant difference, producing statistically larger values for the precision measurements versus all the other types.

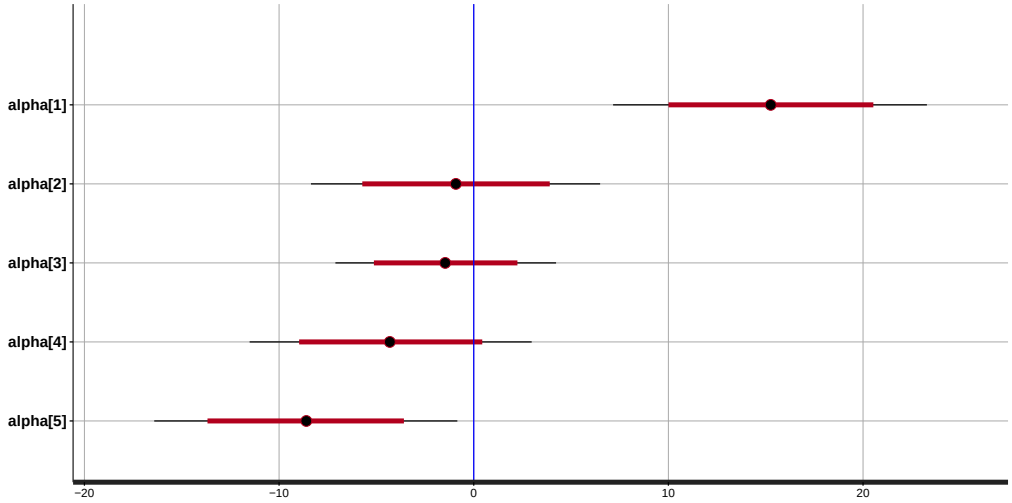

Figure 1: Mirror type effect credible sets

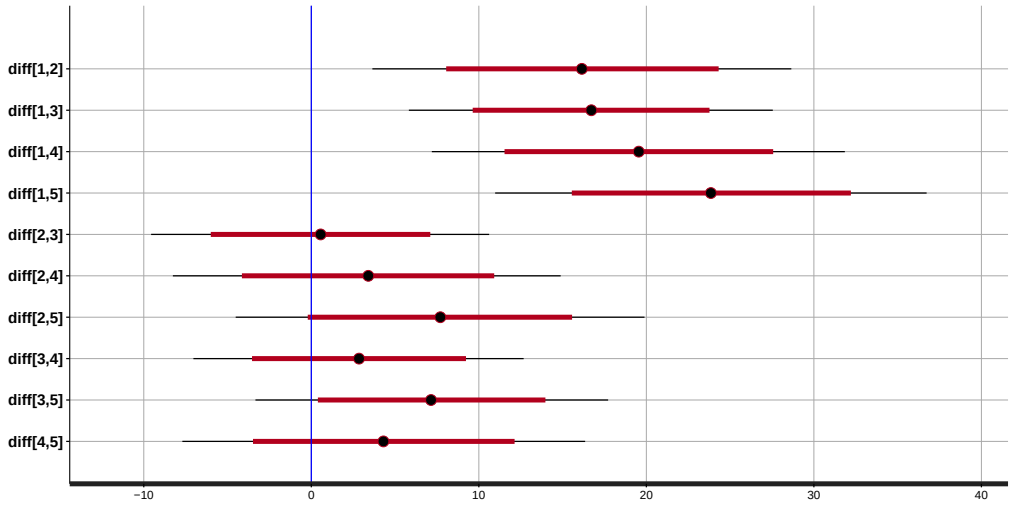

Figure 2: Mirror type effect difference credible sets

1 (c) Two-way Bayesian ANOVA

We will define a similar model for the two-way problem, this time considering each measurement to be $X \sim \mathcal{N}(\mu_0 + \alpha_{\text{mat}} + \beta_{\text{sid}} + \gamma_{\text{mat,sid}}, \sigma^2)$. In Stan,

```
data {
  int<lower=1> N;
  int<lower=1> NMAT;
  int<lower=1> NSID;
  int<lower=1, upper=NMAT> mat[N];
  int<lower=1, upper=NSID> sid[N];
  vector[N] meas;
}
parameters {
  vector[NMAT-1] alpha_raw;
  vector[NSID-1] beta_raw;
  matrix[NMAT-1, NSID-1] gamma_raw;
  real mu0;
  real<lower=0> sigma;
}
transformed parameters {
  vector[NMAT] alpha;
  vector[NSID] beta;
  matrix[NMAT, NSID] gamma;
  for (i in 1:(NMAT-1)) alpha[i] <- alpha_raw[i];
  alpha[NMAT] <- -sum(alpha_raw);
  for (i in 1:(NSID-1)) beta[i] <- beta_raw[i];
  beta[NSID] <- -sum(beta_raw);
  for (i in 1:(NMAT-1))
    for (j in 1:(NSID-1))
      gamma[i, j] <- gamma_raw[i, j];
  for (i in 1:(NMAT-1))
    gamma[i, NSID] <- -sum(gamma_raw[i, :]);
  for (j in 1:(NSID-1))
    gamma[NMAT, j] <- -sum(gamma_raw[:, j]);
  gamma[NMAT, NSID] <- sum(gamma_raw);
}
model {
  vector[N] mu;
  for (i in 1:N)
    mu[i] <- mu0 + alpha[mat[i]] + beta[sid[i]] + gamma[mat[i], sid[i]];
  meas ~ normal(mu, sigma);
}
```

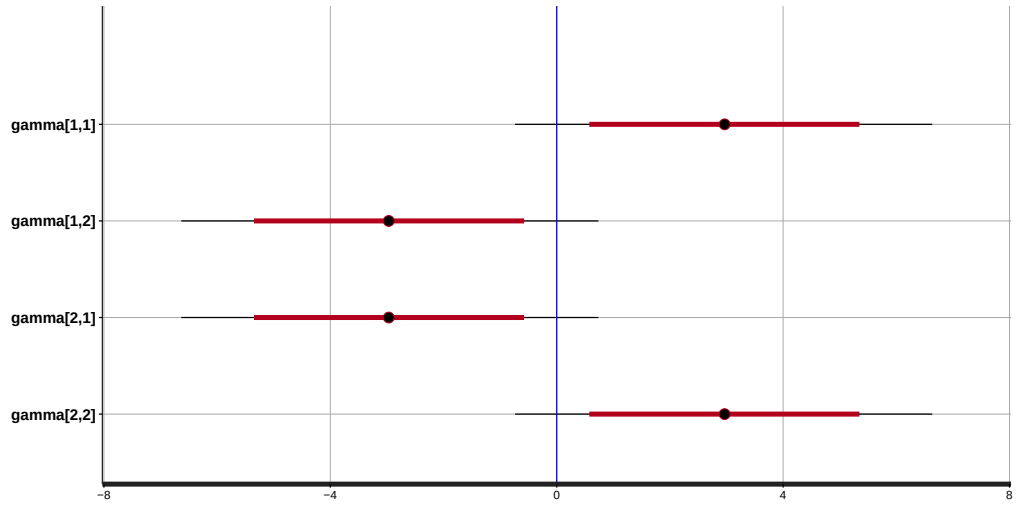

Figure 3: Material-sides interaction effect credible sets

We generate samples of the γ parameters from this model, yielding the following summary:

```
Inference for Stan model: p1c-anova-2way.
4 chains, each with iter=49000; warmup=24500; thin=1;
post-warmup draws per chain=24500, total post-warmup draws=98000.

           mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
gamma[1,1]  2.96    0.01 1.87 -0.74  1.73  2.97  4.21  6.63 72041    1
gamma[1,2] -2.96    0.01 1.87 -6.63 -4.21 -2.97 -1.73  0.74 72041    1
gamma[2,1] -2.96    0.01 1.87 -6.63 -4.21 -2.97 -1.73  0.74 72041    1
gamma[2,2]  2.96    0.01 1.87 -0.74  1.73  2.97  4.21  6.63 72041    1

Samples were drawn using NUTS(diag_e) at Fri Mar 18 08:57:05 2016.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

We also plot the 95% equi-tailed credible sets for the γ parameters (figure 3). As seen in both the numeric summary and the plot, the interaction effect does not meet our 95% confidence requirement for significance, but comes fairly close. More investigation may be necessary.

3 Amanita muscaria

3 (a) Gibbs sampler with gamma prior on τ

The distributions specified for the problem model lead to the standard conjugate relationships and marginal distributions.

```
sampler_gibbs_g <- function(x=spores,
                            mu0=12, sigma0=2, alpha=2, beta=4,
                            mu=NULL, tau=NULL,
                            warmup=1000, keep=100000) {
  tau0 <- sigma0 ^ -2
  n <- length(x)
  sum_x <- sum(x)
  rmu <- function(tau) {
    mean <- (tau0 * mu0 + tau * sum_x) / (tau0 + n * tau)
    sd <- (tau0 + n * tau) ^ -0.5
    rnorm(1, mean=mean, sd=sd)
  }
  rtau <- function(mu) {
    shape <- alpha + (n / 2)
    rate <- beta + sum((x - mu) ^ 2) / 2
    rgamma(1, shape=shape, rate=rate)
  }
  if (is.null(mu)) mu <- rnorm(1, mean=mu0, sd=sigma0)
  if (is.null(tau)) tau <- rgamma(1, shape=alpha, rate=beta)
  rtarget <- function() {
    mu <<- rmu(tau)
    tau <<- rtau(mu)
    c(mu=mu, tau=tau)
  }
  chain <- iter(rtarget)
  for (i in 1:warmup) nextElem(chain)
  do.call(rbind, as.list(ilimit(chain, keep)))
}
```

This sampler yields squared error loss point estimates of $\mu \approx 10.12$ and $\tau \approx 0.46$.

3 (b) Gibbs sampler with inverse gamma prior on σ^2

The distributions for this problem model are also standard conjugate, with the resulting standard marginal distributions.

```
sampler_gibbs_ig <- function(x=spores,
                             mu0=12, sigma0=2, alpha=4, beta=2,
                             mu=NULL, sigma2=NULL,
                             warmup=1000, keep=100000) {
  sigma02 <- sigma0 ^ 2
  n <- length(x)
  sum_x <- sum(x)
  rmu <- function(sigma2) {
    mean <- ((mu0 / sigma02 + sum_x / sigma2) /
               (1 / sigma02 + n / sigma2))
    sd <- (1 / sigma02 + n / sigma2) ^ -0.5
    rnorm(1, mean=mean, sd=sd)
  }
  rsigma2 <- function(mu) {
    shape <- alpha + (n / 2)
    rate <- beta + sum((x - mu) ^ 2) / 2
    1 / rgamma(1, shape=shape, rate=rate)
  }
  if (is.null(mu)) mu <- rnorm(1, mean=mu0, sd=sigma0)
  if (is.null(sigma2))
    sigma2 <- 1 / rgamma(1, shape=alpha, rate=beta)
  rtarget <- function() {
    mu <<- rmu(sigma2)
    sigma2 <<- rsigma2(mu)
    c(mu=mu, tau=(1 / sigma2))
  }
  chain <- iter(rtarget)
  for (i in 1:warmup) nextElem(chain)
  do.call(rbind, as.list(ilimit(chain, keep)))
}
```

The samples from this second model yield squared error loss point estimates of $\mu \approx 10.12$ and $\tau \approx 0.52$.

These results differ from those of the first model due to the change in hyperparameters. Repeating with the original hyperparameters but the second model's inverse gamma prior on σ^2, we calculate our new estimates to be $\mu \approx 10.12$ and $\tau \approx 0.46$. These coincide with our estimates from the first model, reflecting the identical likelihoods resulting from the simple paramaterization difference between the two models.

3 (c) Metropolis-Hastings sampler with gamma prior on τ

To produce a Metropolis-Hastings sampler we will need the density of our joint target distribution up to a normalizing constant. We can calculate the contribution of the normal density $p(\mathbf{x}|\mu,\tau)$ directly as $\prod_{i=1}^{n} p(x_i|\mu,\tau)$, but this is inefficient. We know from standard results in statistical theory that for the normal distribution $\bar{X}$ is a sufficient statistic for location parameter μ, S^2 is sufficient for precision parameter τ, and by Basu's theorem $\bar{X}$ and S^2 are independent. This allows us to calculate the density for the sample as the product of the densities of $\bar{X} \sim \mathcal{N}\left(\mu, [n\tau]^{-1}\right)$ and $(n-1)S^2 \sim \tau^{-1}\chi^2(n-1)$. For the density of $(n-1)S^2$ we must include the contribution of the non-constant transform of the χ^2 random variable via the transform inverse absolute derivative $|(g^{-1})'(x)| = \tau$.

For our proposal distribution we will take uncorrelated Gaussian steps in both dimensions, making our sampler a Metropolis random walk sampler. This can generate proposed values outside of the support of τ, but we will assign such values a density of 0 and reject them as part of the normal proposal-evaluation process.

```
sampler_mh <- function(x=spores,
                       mu0=12, sigma0=2, alpha=2, beta=4,
                       mu=NULL, tau=NULL,
                       warmup=10000, keep=100000) {
  n <- length(x)
  x_bar <- mean(x)
  x_ssd <- (n - 1) * var(x)
  dtarget_log <- function(theta) {
    mu <- theta[1]; tau <- theta[2]
    if (tau <= 0) {
      -Inf
    } else {
      dnorm(x_bar, mean=mu, sd=((n * tau) ^ -0.5), log=TRUE) +
        dchisq(tau * x_ssd, df=(n - 1), log=TRUE) + log(tau) +
        dnorm(mu, mean=mu0, sd=sigma0, log=TRUE) +
        dgamma(tau, shape=alpha, rate=beta, log=TRUE)
    }
  }
  rproposal <- function(theta) {
    xi <- rnorm(length(theta), mean=theta, sd=c(1.5, 0.1))
    names(xi) <- names(theta)
    xi
  }
  if (is.null(mu)) mu <- rnorm(1, mean=mu0, sd=sigma0)
  if (is.null(tau)) tau <- rgamma(1, shape=alpha, rate=beta)
  theta <- c(mu=mu, tau=tau)
```

```
  rtarget <- function() {
    xi <- rproposal(theta)
    rho <- exp(min(0, dtarget_log(xi) - dtarget_log(theta)))
    theta <<- if (runif(1) <= rho) xi else theta
    theta
  }
  chain <- iter(rtarget)
  for (i in 1:warmup) nextElem(chain)
  do.call(rbind, as.list(ilimit(chain, keep)))
}
```

This sampler also yields the expected squared error loss point estimates of $\mu \approx 10.12$ and $\tau \approx 0.46$.

4 Squids

4 (a) Linear regression

We define a linear regression model in Stan, accounting for the assumed-MCAR x_1 and y values, supporting sampling-time prediction for new inputs, and calculating the Ibrahim-Laud L criteria and the predictive log-likelihood necessary for LOO estimation.

Parameter and predictive point estimates for this model show little sensitivity to priors, but their credible sets do, under entirely non-informative priors generating nonsensically wide intervals (e.g., including negative squid weights). Following advice in the Stan manual and Gelman et al. (2008),[1] we regularize the regression coefficients by centering and scaling each of the x variables to a common range then placing on the coefficients a common Cauchy prior scaled to the distribution of y.

This results in the following model, which while somewhat complicated we will use without further modification for the remaining parts of this problem.

```
functions {
  matrix rescale(matrix x, row_vector x_bar, row_vector s) {
    return ((x - rep_matrix(x_bar, rows(x)))
            ./ rep_matrix(s, rows(x)) ./ 2);
  }
}
data {
  int<lower=1> K;
  int<lower=1> N_cc;
  int<lower=0> N_mis_x1;
  int<lower=0> N_mis_y;
  int<lower=0> N_pred;
  row_vector[K] x_bar;
  row_vector[K] s;
  matrix[N_cc, K] x_cc;
  matrix[N_mis_x1, K] x_mis_x1;
  matrix[N_mis_y, K] x_mis_y;
  matrix[N_pred, K] x_pred;
  vector[N_cc] y_cc;
  vector[N_mis_x1] y_mis_x1;
}
transformed data {
  int N_obs;
  int P;
```

[1] Gelman, A., Jakulin, A., Pittau, M. G., and Su, Y.-S. (2008). A weakly informative default prior distribution for logistic and other regression models. *Annals of Applied Statistics*, 2(4):1360–1383.

```
  matrix[N_cc, K] x_cc_std;
  matrix[N_mis_x1, K] x_mis_x1_std;
  matrix[N_mis_y, K] x_mis_y_std;
  matrix[N_pred, K] x_pred_std;
  vector[N_cc + N_mis_x1] y_obs;
  real scale_beta;
  real sst;
  N_obs <- N_cc + N_mis_x1;
  P <- K + 1;
  x_cc_std <- rescale(x_cc, x_bar, s);
  x_mis_x1_std <- rescale(x_mis_x1, x_bar, s);
  x_mis_y_std <- rescale(x_mis_y, x_bar, s);
  x_pred_std <- rescale(x_pred, x_bar, s);
  y_obs <- append_row(y_cc, y_mis_x1);
  scale_beta <- 2.5 * sd(y_obs);
  sst <- dot_self(y_obs - mean(y_obs));
}
parameters {
  real<lower=0> sigma;
  real beta0;
  vector[K] beta;
  vector[N_mis_y] y_mis_y;
  vector[N_mis_x1] x1_imputed;
}
transformed parameters {
  matrix[N_obs, K] x_obs_std;
  x_obs_std <- append_row(x_cc_std, x_mis_x1_std);
  if (N_mis_x1 > 0)
    x_obs_std[(N_cc+1):, 1] <- (x1_imputed - x_bar[1]) / s[1] / 2;
}
model {
  beta ~ cauchy(0, scale_beta);
  if (N_mis_x1 > 0) x1_imputed ~ normal(3.0/2.0, 0.5);
  if (N_mis_y > 0)
    y_mis_y ~ normal(beta0 + x_mis_y_std * beta, sigma);
  y_obs ~ normal(beta0 + x_obs_std * beta, sigma);
}
generated quantities {
  vector[N_obs] log_lik;
  vector[N_pred] y_pred;
  real BR2;
  real L2;
```

```
  {
    vector[N_obs] Z;
    for (i in 1:N_obs) {
      real mu;
      mu <- beta0 + x_obs_std[i] * beta;
      log_lik[i] <- normal_log(y_obs[i], mu, sigma);
      Z[i] <- normal_rng(mu, sigma);
    }
    L2 <- dot_self(Z - y_obs);
    BR2 <- 1 - (N_obs - P) * sigma^2 / sst;
    for (i in 1:N_pred)
      y_pred[i] <- normal_rng(beta0 + x_pred_std[i] * beta, sigma);
  }
}
```

Using the resulting samples, we estimate this model to have a Bayesian R^2 of 0.95. We estimate the missing $x_1 \approx 1.38$ and the missing $y \approx 9.39$.

4 (b) Prediction

Our model predicts y for the specified x values to be 3.29 with squared error loss, and a 95% equi-tailed credible set of (0.1, 6.49).

4 (c) Model selection

We re-fit the model five times, each time removing one of the x predictor variables and recording the resulting L^2 and log-likelihood values. We then use these values to compute for each reduced model the Ibrahim-Laud $L = \sqrt{L^2}$ criteria and the deviance-scale PSIS-LOO[2] estimated leave-one-out cross validation expected log-likelihood.

	full	x_1	x_2	x_3	x_4	x_5
L	4.85	4.74	4.99	4.72	5.19	6.06
looic	57.35	55.00	55.20	53.94	59.32	66.02

In this case both metrics agree, taking lowest values for the model eliminating variable x_3.

For these posterior predictive checks, we have elected to include the observation missing x_1 under multiple imputation, but exclude the observation missing y. It does not seem to make sense to check the performance of a model versus an observation predicted via the model. That said, we generated the predictive performance metrics under each combination of including/excluding the missing-data observations and found the same model (that excluding x_3) most effective under all combinations.

[2]Vehtari, A., Gelman, A., and Gabry, J. (2016). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. arXiv preprint: `http://arxiv.org/abs/1507.04544`