1 Gibbs Sampling

We are interested in Gibbs sampling for normal linear regression with one independent variable. We assume we have paired data (y_i, x_i), $i = 1, \dots, N$. We wish to find the posterior distributions of the coefficients β_0 (the intercept), β_1 (the gradient) and of the precision τ, which is the inverse variance. The model can be written as

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \tau^{-1})$$

where τ is the precision (inverse variance).
The likelihood for this model may be written as the product over N i.i.d. observations is

$$p(y|X, \beta_0, \beta_1, \beta_2, \tau) = \prod_{i=1}^{N} \mathcal{N}(y_i|\beta_0 + \beta_1 x_i + \beta_2 x_i^2, \tau^{-1})$$

We place conjugate priors on $\beta_0, \beta_1, \beta_2$ and τ. We choose

$$\begin{aligned}
\beta_0 &\sim \mathcal{N}(\mu_0, \tau_0^{-1}) \\
\beta_1 &\sim \mathcal{N}(\mu_0, \tau_1^{-1}) \\
\beta_2 &\sim \mathcal{N}(\mu_0, \tau_2^{-1}) \\
\tau &\sim \text{Gamma}(\alpha, \beta)
\end{aligned}$$

Implement a Gibbs sampler for this model.
Submit your final code to CATe via the LabTS system.

The general approach to deriving an update for a variable is:

1. Write down the log-joint distribution over all variables
2. Throw away all terms that do not depend on the current sampling variable
3. Pretend this is the density for your variable of interest and all other variables are fixed. What distribution does the log-density remind you of?
4. That is your conditional sampling density.

1.1 Task 1 (20 marks)

Implement a sampler for the conditional distribution

$$p(\beta_0|\beta_1, \beta_2, \mu_0, \tau_0, \tau, X, y)$$

Hint: This conditional is Gaussian.

1.2 Task 2 (20 marks)

Implement a sampler for the conditional distribution

$$p(\beta_1|\beta_0, \beta_2, \mu_1, \tau_1, \tau, X, y)$$

Hint: This conditional is Gaussian.

1.3 Task 3 (20 marks)

Implement a sampler for the conditional distribution

$$p(\beta_2|\beta_0, \beta_1, \mu_2, \tau_2, \tau, X, y)$$

Hint: This conditional is Gaussian.

1.4 Task 4 (20 marks)

Implement a sampler for the conditional distribution

$$p(\tau|\beta_0, \beta_1, \beta_2, \alpha, \beta, X, y)$$

Hint: This conditional is Gamma.

1.5 Task 5: Gibbs sampler (20 marks)

Implement the Gibbs sampler using the conditional distributions from Tasks 1–4.

2 Large-Scale Gaussian Process Challenge (not assessed)

In this task, you can explore large-scale Gaussian processes. You can use any kind of GP package, e.g., GPflow[1], which will also require you to install Tensorflow[2]. GPflow's interface works pretty much like GPy, which you used in the labs last week. You can use the Microsoft Azure[3] budget that we allocated to you for this course (currently 500 USD per person) for your computations.

2.1 Problem Setting

We consider a large-scale non-stationary data set reporting flight arrival and departure times forevery commercial fight in the US from January to April 2008.[4] The data set can be downloaded using the `get_data.sh` script and parsed using the `parse_plane_data.py` python script. This data set contains information about almost 6 million flights. The goal is to predict the flight delay (in minutes) at arrival. Compute the RMSE and the negative log-likelihood per test point.
If you normalize the data, you can only use the training data to determine the normalization terms (mean and standard deviation). When you compute errors (NLL and RMSE), please compute them in the original y-space. You can make predictions to normalized y-values, but then you have to un-do the normalization:

[1] `https://github.com/GPflow/GPflow`
[2] `https://www.tensorflow.org/`
[3] `https://azure.microsoft.com/`
[4] `http://stat-computing.org/dataexpo/2009/`

```
# transform (Gaussian) predictions of normalized variables
# into predictions  in original data space
def unNormalizePrediction(m, S, xmean, xstd):
    # un-do the normalization in the prediction
    # x = xNormalized*xstd + xmean
    # where xNormalized \sim N(x|m, S)
    # then x \sim N(x| m*xstd, xstd^2*S)
    mu = m*xstd + xmean
    Sigma = S*xstd**2
    return mu, Sigma
```

2.1.1 Task 1: 700K

Select the first 700,000 data points to train the mode, and the following 100,000 to test it.

2.1.2 Task 2: 5M

Select the first 5,000,000 data points and test on the following 100,000.

Reporting Results Add your results (RMSE and NLL) to the benchmark overview spreadsheet[5].

2.2 Remarks

Models To tackle this challenge, you can explore any large-scale GP model of your choice, including some of the models that we discussed in the course ((generalized) product of experts, (robust) Bayesian committee machine).

Potentially useful Tricks

- Consider normalizing the data (numerical stability, ..)
- Use some good heuristics to initialize the hyper-parameters before training (see slides on GPs)
- At test time, do not test at the full test set at once, but do this in batches (e.g., size 1,000).

[5] https://docs.google.com/spreadsheets/d/12f-Niih4lnYtxKFBV78sepFQ3ZSinTcF1Pio-JDfsSk/edit?usp=sharing