Assignment #5

Due: 23:59pm November 22, 2013

Problem 1 (50pts)

In this problem, you will implement some simple computations for Gaussian processes with a one-dimensional input space. You can assume that the GP is zero mean and has the "squared exponential" covariance function (with a little bit of diagonal "jitter"):

$$K(x, x') = \alpha \exp\left\{-\frac{1}{2\ell^2}(x - x')^2\right\} + 10^{-6}\delta(x, x').$$

1. Create a grid with about 100 points or so in an input space between zero and five. Make several plots, each with different values for α and ℓ. Sample ten functions from the Gaussian process for each plot.

2. Recall from the midterm that if one has two Gaussian variates $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, both with marginal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, then

   $$\boldsymbol{x}_3 = \boldsymbol{x}_1 \cos\theta + \boldsymbol{x}_2 \sin\theta$$

   also has marginal distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ for any θ.

   For each of the above plots, take two of the independent samples and blend them as in the equation above. Use a range of points for $\theta \in (-\pi, \pi)$. You can see that varying θ results in a smooth blending of these two functions, while still producing a function that is marginally from the same Gaussian process. This is the trick that motivates the *elliptical slice sampling* algorithm.

3. Come up with three or four "training data", i.e., $\{x_n, y_n\}$ pairs. For each of your hyperparameter variations, plot several functions from the posterior distribution implied by these training data.

4. Using the same training data, plot the 95% marginal envelope for the function. This is like the grey area shown in Figure 2.2b in the Rasmussen and Williams book.

5. Compute and report the log marginal likelihoods for each of the hyperparameter settings you have looked at. Discuss how these marginal likelihoods reflect how well the hyperparameters fit the training data you invented.

Problem 2 (50pts)

In this problem, you'll build a basic implementation of a Dirichlet process mixture model.

1. Warmup: Make 200 draws from several different instantiations of a Chinese restaurant process, using different concentration parameters α. For each of your concentration parameters, produce a bar graph that shows how many "customers" have been assigned to each "table".

2. Implement MCMC for a Dirichlet process mixture of Gaussians. Two papers (among many) to check out for your implementation are:

   - Carl Edward Rasmussen. *The Infinite Gaussian Mixture Model*. NIPS, 1999. [Don't bother implementing adaptive rejection sampling for the hyperparameter; use slice sampling instead.]
   - Radford M. Neal. *Markov Chain Sampling Methods for Dirichlet Process Mixture Models*. Journal of Computational and Graphical Statistics. 9:249-265, 2000. [Neal's related technical report is also good to check out.]

   This is a good opportunity to try out Geweke-style validation of your hyperparameters. That is, there is a hyperparameter α, parameters θ and data $\mathcal{D}$. Your model specifies a joint distribution

   $$p(\alpha, \theta, \mathcal{D}) = p(\alpha)\, p(\theta \,|\, \alpha)\, p(\mathcal{D} \,|\, \theta).$$

   Your inference will sample from the conditional distribution $p(\alpha, \theta \,|\, \mathcal{D})$, but you can also fantasize data from $p(\mathcal{D} \,|\, \theta)$. Augment your MCMC with fantasy data and then examine the samples that you get for α. Make sure that their histogram looks like the prior.

3. Apply your implementation to data of your choice and report the results. As a minimum, I suggest applying it to some two-dimensional synthic data and visualizing the clusters it discovers. One tool you might check out for generating such synthetic data can be found here: `http://hips.seas.harvard.edu/content/synthetic-pinwheel-data-matlab`.