Ze Jia Zhang (3184 7771)

EECS587 Project Proposal: Clustering for Multiscale Gaussian Processes

7 November 2015

Approved: Yes, during office hours

Description: I would like to focus on machine learning for this project, specifically, on supervised regression. Supervised regression involves predicting the output at a new input given a known database of input-output pairs. My research group is developing a new variant of the Gaussian process (GP) regression approach known as a multiscale Gaussian process, or MGP. We intend to use this method when working with large datasets numbering a million or more data points.

Given N known inputs, a traditional GP would construct a $N \times N$ covariance matrix and invert it, costing $O(N^3)$. For large N, this is quite prohibitive, especially if we want to predict the output at M points for an additional cost that is $O(MN)$. For our application, the training cost of $O(N^3)$ is not as important as the testing cost of $O(MN)$, but nevertheless it is desirable to lower both. The goal of MGP is to reduce the effective size of N by performing a clustering operation on the inputs prior to constructing the covariance matrix. Only cluster centres are used in place of the entire training dataset. Typically, N can be reduced by a factor of 5 to 10 in this manner, resulting in significant performance gains. This clustering operation can be thought of as the opposite of K-means, wherein the distance between cluster centres is proscribed, but the total number of clusters is not. It is also performed multiple times on the data for different distances, hence the name "multiscale."

For the purposes of this course, I would like to implement the clustering algorithm in parallel. Up to now, I have been using a serial implementation and have been restricted in using large training datasets by the limited amount of memory available on a single computer. A parallel version that can take advantage of my group's resources on Flux would be very helpful. I can also extend the parallelism to the entire GP algorithm if time allows.

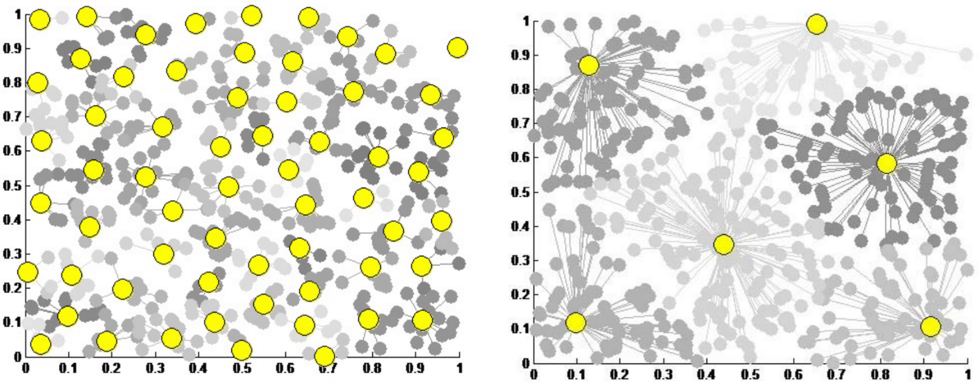

Figure 1: The separation of 500 random points distributed inside a unit square into clusters. The cluster centers are shown in yellow; points belonging to different clusters are in different shades of gray. Left: small h_s. Right: large h_s.

Figure 1 illustrates the clustering of a 2-dimensional dataset. For the purposes of MGP, I want to start with a large separation distance, choose random points that are separated by that distance (or more), record and remove these from the data, then repeat with a smaller distance. The distance h at step s is related by $h_s = h_1\beta^{s-1}$, $s = 1, ..., S$.