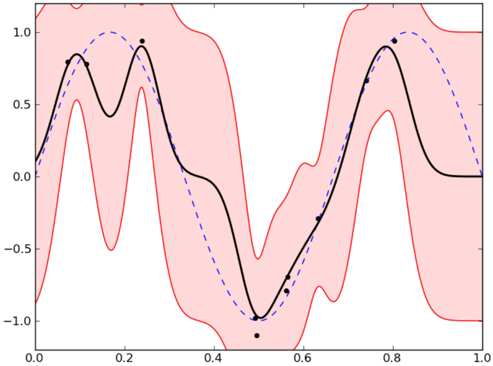
Gaussian Process Regression Using the Improved Fast Gauss Transform

Chi Feng

18.336
May 15, 2013

Motivation for Gaussian Process Regression

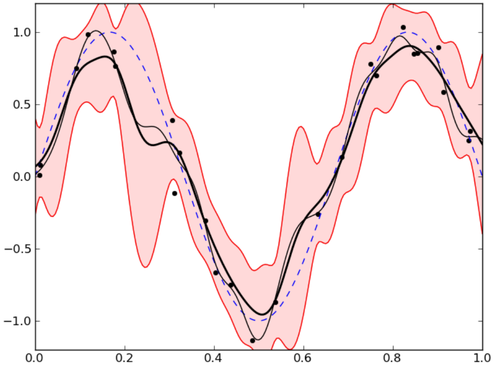
Non-parametric regression that quantifies uncertainty (pink regions represent 95% CL)



$N = 10$ points

Motivation for Gaussian Process Regression

Non-parametric regression that quantifies uncertainty (pink regions represent 95% CL)



$N = 25$ points

Computational Challenges in Gaussian Process Regression

- The *covariance matrix*

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \cdots & k(x_N, x_N) \end{bmatrix} \tag{1}$$

- $k(x, x')$ is the *covariance function*:

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2\ell^2}\right] + \sigma_n^2 \delta(x, x'), \tag{2}$$

- Can predict mean and variance of y_* at x_*:

$$\overline{y}_* = K_* K^{-1} \mathbf{y} \tag{3}$$

$$\mathrm{var}(y_*) = K_{**} - K_* K^{-1} K_*^T \tag{4}$$

Conjugate gradient method for matrix inversion

Solving

$$\hat{K}|x\rangle = |b\rangle \implies |x\rangle = \hat{K}^{-1}|b\rangle$$

is equivalent to minimization of

$$f(|x\rangle) = \frac{1}{2}\langle x|\hat{K}|x\rangle - \langle x|b\rangle$$

CG method requires evaluation of $\hat{K}|p\rangle$ each iteration, if $\hat{K}$ is matrix of Gaussian kernels, $\hat{K}|p\rangle$ is equivalent to the *Discrete Gauss Transform*

$$G(y_j) = \sum_{i=1}^{N} p_i e^{-||y_j - x_i||^2/h^2} \tag{5}$$

The Improved Fast Gauss Transform

- The *Discrete Gauss Transform*

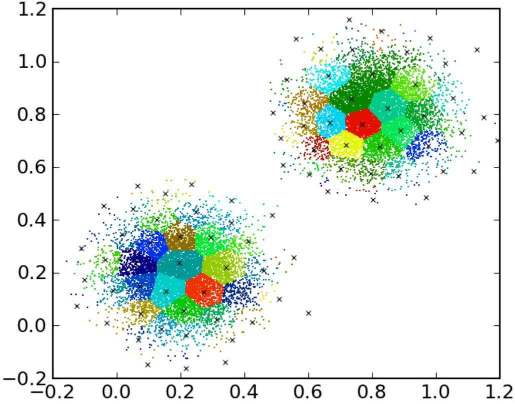
$$G(y_j) = \sum_{i=1}^{N} q_i e^{-||y_j - x_i||^2/h^2} \tag{6}$$

- q_i are weight coefficients,
- x_i are the centers of the Gaussians ("source" points),
- h is the bandwidth of the Gaussians.

Normally, with N "source" points, and M "target" points, we need to evaluate and sum $N \times M$ square exponentials. The Improved Fast Gauss Transform is an ϵ-exact approximation that reduces complexity from $O(NM)$ to $O(M + N)$.

The Improved Fast Gauss Transform

Use k-center clustering (farthest point algorithm), greedy, $O(N)$



Efficient partitioning of space vs. multilevel grids from FMM, esp. in high dimensions

The Improved Fast Gauss Transform

Sum of Gaussians approximated as (multinomial expansion as sum of monomials)

$$G(y_j) = \sum_{i=1}^{N} q_i e^{-||y_j - x_i||^2/h^2}$$

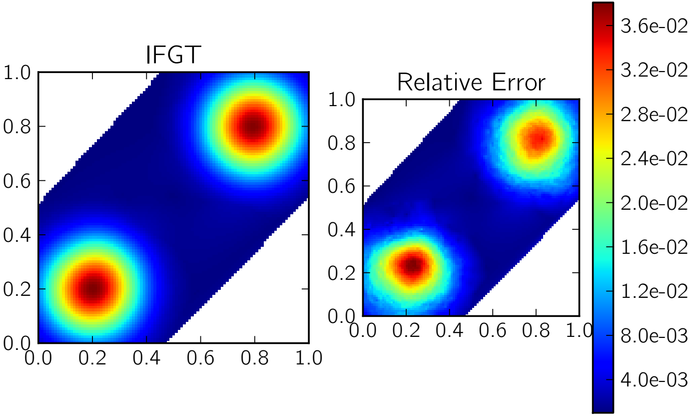
$$\approx \sum_{|y_j - c_k| \le h\rho_y} \sum_{|\alpha| \le p} C_\alpha^k e^{-|y_j - c_k|^2/h^2} \left(\frac{y_j - c_k}{h} \right)^\alpha$$

Coefficients in *graded lexicographical order*, Size of α is $\binom{p-1+d}{d}$

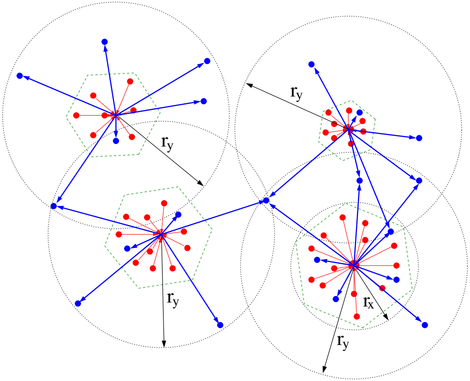
$$C_\alpha^k = \frac{2^{|\alpha|}}{\alpha!} \sum_{x_i \in S_k} q_i e^{-|x_i - c_k|^2/h^2} \left(\frac{x_i - c_k}{h} \right)^\alpha$$

The Improved Fast Gauss Transform

Speedup: 83x vs. direct evaluation for 20k points



The Improved Fast Gauss Transform: Error Bound



Truncation term:

$$|E_T| \leq \frac{2^p}{p!}\left(\frac{r_x r_y}{h}\right)^p$$

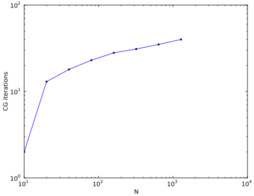
Cutoff term:

$$|E_C| \leq e^{-r_y^2/h^2}$$

$$\text{Total: } |E(y)| \leq Q\left(\frac{2^p}{p!}\left(\frac{r_x r_y}{h}\right)^p + e^{-r_y^2/h^2}\right)$$

Applying IFGT to GPR

- Rewrite $K^{-1}\mathbf{y}$ as CG minimization problem
  - Numerical experiments show that CG minimization without IFGT use 15-30 weak scaling with N



- Re-frame matrix-vector multiplication $K|p\rangle$ as a discrete Gauss transform
  - i.e., $N \times N$ matrix $\rightarrow$ N source points, evaluated at N target points with weights given by p_i, $i = 1, \ldots, N$.

Applying IFGT to GPR

$$K|p\rangle = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdots & k(x_1,x_N) \\ k(x_2,x_1) & k(x_2,x_2) & \cdots & k(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \cdots & k(x_N,x_N) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^N p_i k(x_1,x_i) \\ \sum_{i=1}^N p_i k(x_2,x_i) \\ \vdots \\ \sum_{i=1}^N p_i k(x_N,x_i) \end{bmatrix} = \begin{bmatrix} G(x_1) \\ G(x_2) \\ \vdots \\ G(x_N) \end{bmatrix}$$

$$\text{where } G(x_j) = \sum_{i=1}^{N} \underbrace{(p_i \sigma_f^2)}_{q_i} \exp\left[\frac{-(x_j - x_i)^2}{h^2}\right], \quad h = \sqrt{2}\ell$$

Can evaluate $K|p\rangle$ in $O(N+N)$ instead of $O(N^2)$ on $G(\mathbf{x})$.

Implementation

- To evaluate IFGT:
```
IFGT ifgt(sources, weights, 0.1, 15, 0.3, 2);
ifgt.evaluate(sources, result);
```
- To evaluate $K|x\rangle$:
```
Vector Kx;
Vector x(N, sigma_f * sigma_f);
IFGT KxIFGT(sources, x, sqrt(2) * length, degree, radius,
cutoff);
KxIFGT.evaluate(sources, Kx);
```
- C++, no special libraries.
- View project on github
`https://github.com/chi-feng/ifgt-gpr/tree/master/src`

Future Work

- Apply IFGT to *hyperparameter selection* (choose σ_f, σ_n, ℓ to maximize posterior likelihood), i.e. maximize:

$$\log p(\mathbf{y}|\mathbf{x}, \sigma_f, \sigma_n, \dots) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2} log|K|$$

- Apply IFGT to other covariance kernels.
    - Multiple length-scales (to capture oscillations)

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2\ell_1^2}\right] + \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2\ell_2^2}\right]$$

- Apply IFGT-accelerated Gaussian Process Regression to interesting problems, e.g. level sets and classification of high-dimensional data.

Bibliography

- Changjiang Yang and Duraiswami, R. and Gumerov, N.A. and Davis, L., *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference*, "Improved fast gauss transform and efficient kernel density estimation," (2003)
- M. Ebden, *Gaussian Processes for Regression: A Quick Introduction* (2008) `http://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf`
- Rasmussen, C. and C. Williams (2006). *Gaussian Processes for Machine Learning.* MIT Press.