Scaling kernel machine learning algorithm via the use of GPUs

Balaji Vasan Srinivasan, Ramani Duraiswami

Kernel approaches to learning

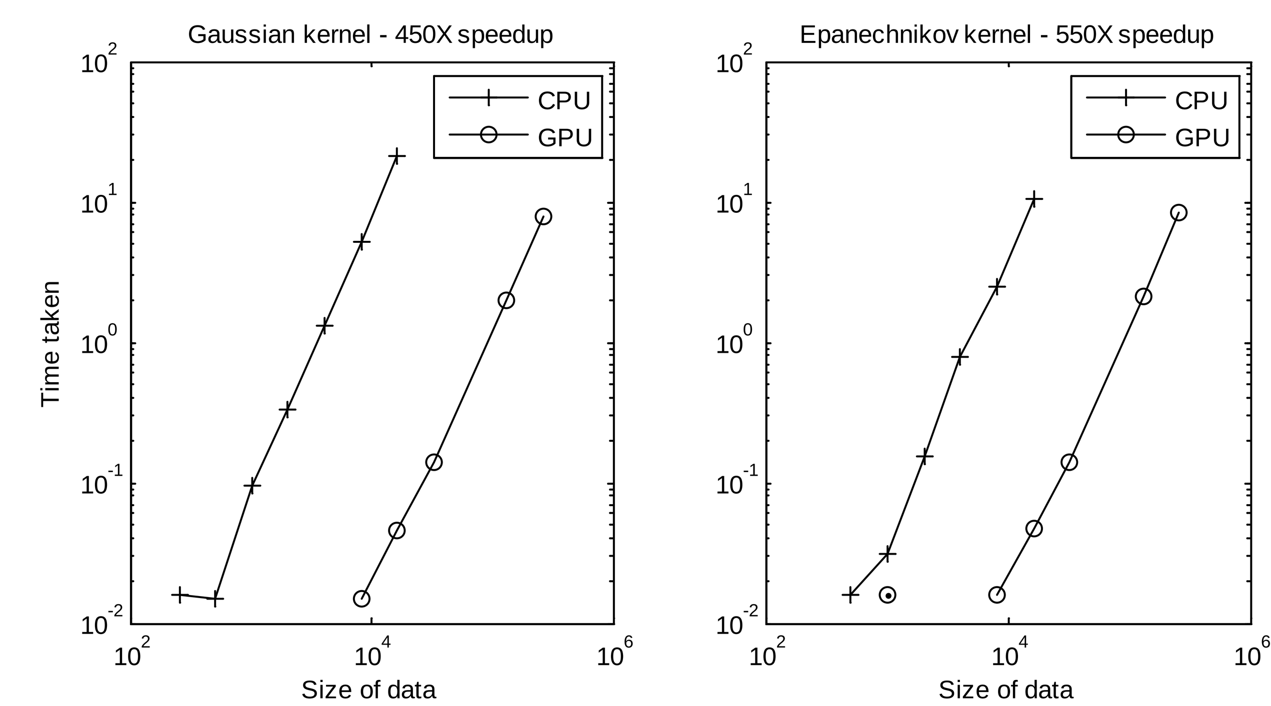
- A class of robust non-parametric learning methods
- Involves a definition for a "*kernel function*" [1]
  - Ex. Gaussian kernel: $k(x_1, x_2) = s \times \exp\left(-\frac{\|x_1 - x_2\|^2}{h^2}\right)$
- Learning methods based on kernels involves
  - A weighted summation of kernel functions ($K(x,x_i)$)
    - $\sum_{i=1}^{N} q_i k(x_i, x)$
  - Solution of linear system based on kernel matrices
- Scales $O(N^2)$ or $O(N^3)$ in time
- $O(N^2)$ in memory
- **Objective:** Use GPU to accelerate kernel machine learning approaches

Proposed acceleration approach

- Linear systems of kernel matrices can be solved using iterative methods like Conjugate Gradient
  - Each iteration will now involve a weighted kernel summation
- Approach to accelerate kernel sums of the form, $G(y_j) = \sum_{i=1}^{N} q_i \exp\left(-\frac{\|x_i - y_j\|^2}{h^2}\right), j = 1, \ldots, M$
  - Assign each thread to evaluates the sum corresponding to one y_j
- **Steps:**
  1. Load yj corresponding to the current thread in to a local register.
  2. Load the first block of x_i to the shared memory.
  3. Evaluate part of kernel sum corresponding to x_i's in the shared memory.
  4. Store the temporary result in a local register.
  5. If all the x_i's have not been processed yet, load the next block of x_i's, go to Step 3.
  6. Write the sum in the local register to the global memory.

Kernel Density Estimation

- Non-parametric way of estimating probability density function of a random variable [1]
- Density,
  - Two popular kernels: *Gaussian* and *Epanechnikov*
- Obtained speed up of ~450X on the data from [2]

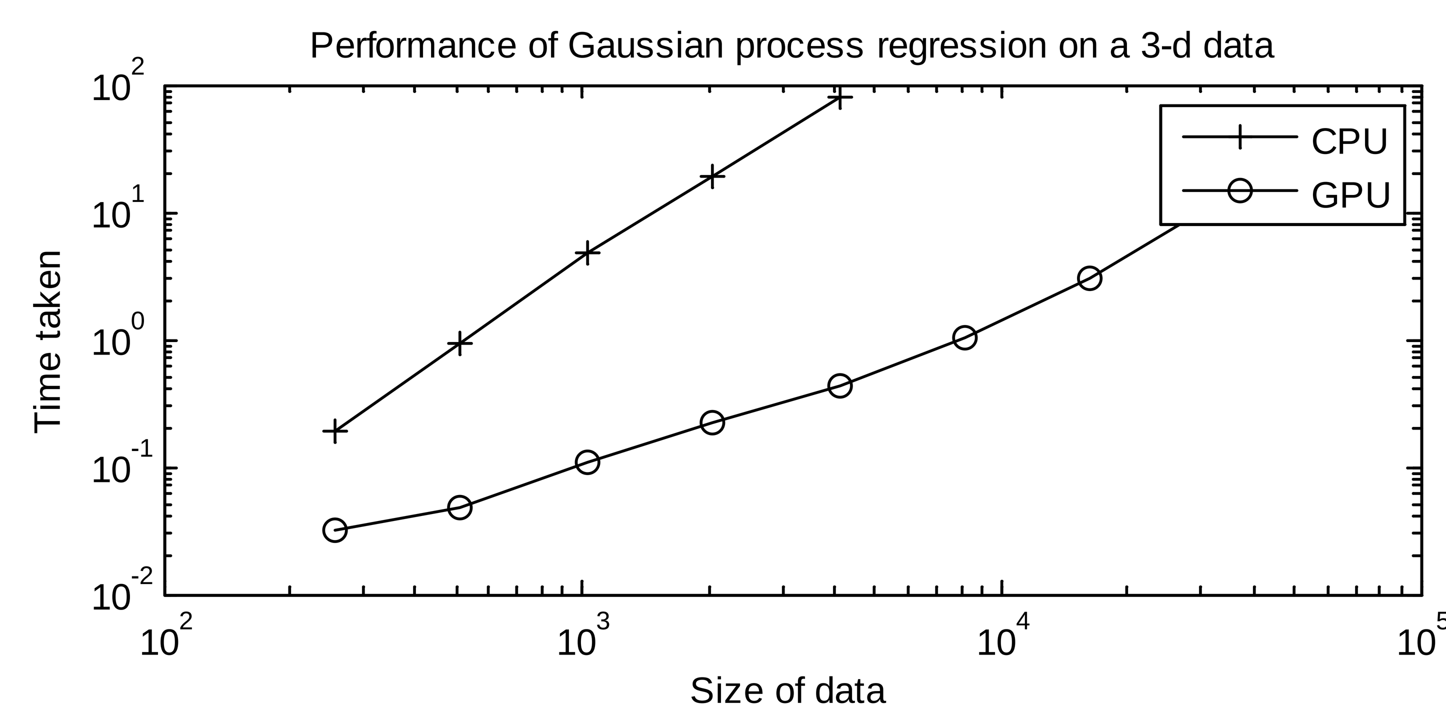


Gaussian Process Regression

- Non-parametric robust regression [3]
- Given training data $\{x_i, y_i\}$, $i=1,..,N$, test data $\{x^*_j, y^*_j\}$, $j=1,..,M$, predictions at x^*'s is given by

  $$y^* = K(x^*, x) \times K(x,x)^{-1} \times y$$

- Time complexity: $O(N^3)$, Space complexity $O(N^2)$
- Time complexity reduced to $O(kN^2)$ using Conjugate Gradient, further accelerated using the proposed approach

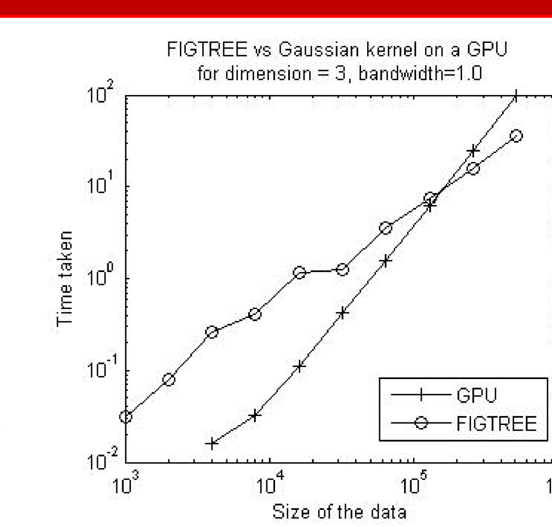


Ranking

- In information retrieval, a ranking function f maps a pair of data-points to a score value, which can be sorted for ranking (according to their relevance), ex. *search engines*.
- In [5], the authors maximize the generalized Wilcoxon-Mann-Whitney (WMW) statistic non-linear conjugate gradient approach to learn the ranking function.
  - The formulation simplifies to summation of *erfc* function, applied proposed approach here.
- Result with standard datasets in [4]
  - California Housing, d=9, N=20640
    - GPU Time: 1.84s (Time taken by [5]: 45.2s)
  - MachineCpu, d=22, N=8192
    - GPU Time: 0.53s (Time taken by [5]: 4.08s)
- Note: The algorithm in [5] is a linear algorithm, ours is a quadratic time complexity, still our approach outperforms [5] for large datasets

Comparison with FIGTREE [6]

- FIGTREE is a linear algorithm to accelerate Gaussian kernel summation
- For a 3-dimensional data, the linear approach outperforms our quadratic approach only beyond a datasize of 100,000



Available as an open source: www.umiacs.umd.edu/~balajiv/GPUML.htm

References:

1. R. Duda, P Hart, and D Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.
2. J Marron and MWand. **Exact mean integrated squared error**, *The Annals of Statistics*, pp. 712–736, 1992.
3. C Rasmussen and C Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.
4. www.liaad.up.pt/~ltorgo/Regression/DataSets.html
5. V Raykar, R Duraiswami, and B Krishnapuram. **A fast algorithm for learning a ranking function from large-scale data sets**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1158–1170, 2008.
6. V Morariu, B Srinivasan, V Raykar, R Duraiswami, and L Davis. **Automatic online tuning for fast Gaussian summation.** In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*. MIT Press, 2008.

UMIACS University of Maryland Institute for Advanced Computer Studies

GPU Technology Conference 2009

PIRL Perceptual Interfaces and Reality Laboratory