A Distributed Kernel Summation Framework for Machine Learning and Scientific Applications

Dongryeol Lee

Ph.D. Candidate in Computer Science
School of Computational Science and Engineering
College of Computing, Georgia Institute of Technology

May 4, 2012 10 AM - 12 PM, KACB 1212

Outline

Outline

Outline

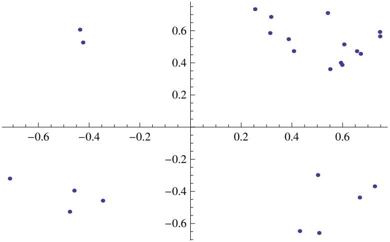
Outline

Outline

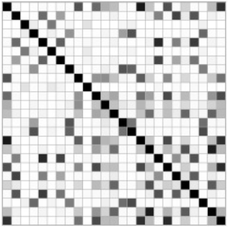
Outline

Kernel Methods

A kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$: Defines a similarity measure between a pair of objects.



$\Leftrightarrow$



Example: Gaussian $\{K_{i,j}\}_{1 \leq i,j \leq N} = e^{-||x_i - x_j||^2/(2h^2)}$

Kernel Summations

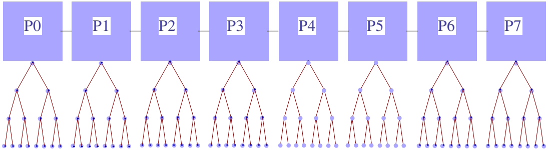
"Kernel summations is the computational bottleneck ubiquitous in kernel methods and scientific algorithms."

Kernel function which outputs a real number given a tuple of points.

Kernel summation computes $\approx$ average similarity.

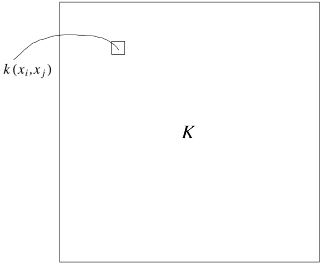
Distributed Data

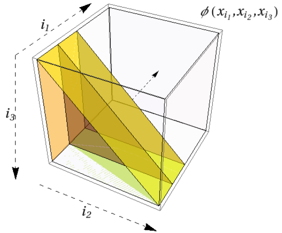
If the disk space is cheap, why can't we store everything on one machine?



- More cost-effective to distribute data on a network of less powerful nodes than storing everything on one powerful node.
- Allows distributed query processing for high scalability.
- In some cases, all of the data cannot be stored on one node due to privacy concerns.

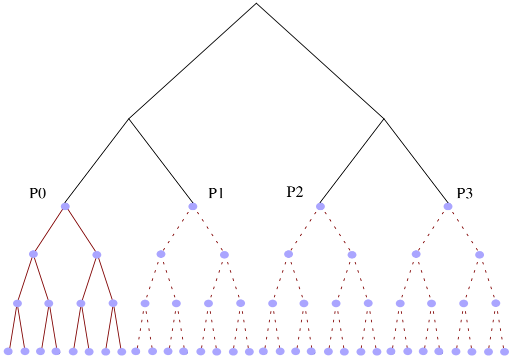
Why are Kernel Methods Hard to Scale?





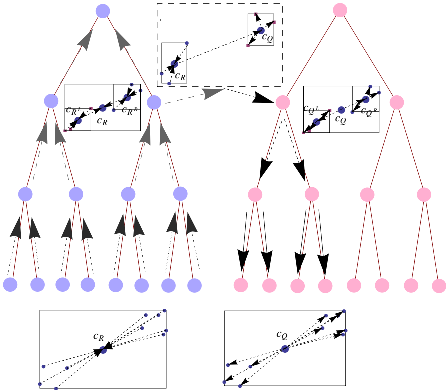
The problem is inherently super-quadratic in the number of data points.
How do we break up the pairwise/higher-order interaction?

Solution Preview 1: Distributed Tree

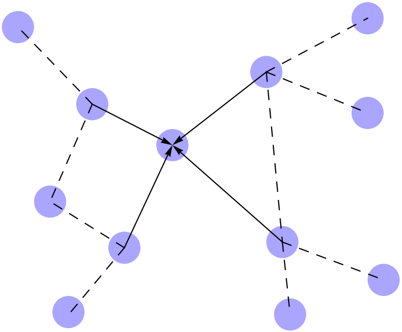


Solution Preview 2: Approximation Methods

Different approximation methods.



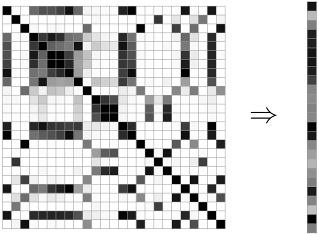
Solution Preview 3: Distributed Averaging

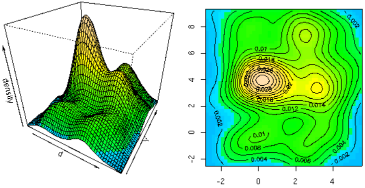


Outline

Kernel Methods for Density Estimation

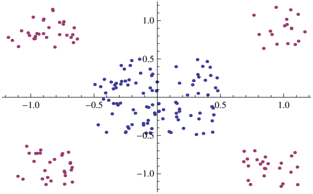
If the kernel is a pdf, then we can do density estimation. Kernel density estimation [Parzen 1962]: Weighted column average of the kernel matrix (map $\sum_{q \in Q} \sum_{r_i \in R} w_i k(q, r_i) = K \cdot w$).



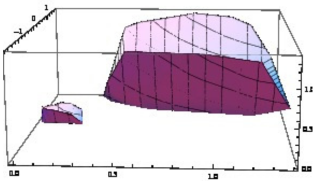


Kernel Methods

A kernel k is a similarity function.
If it in addition satisfies the Mercer's conditions ($K \succ 0$), then it corresponds to a dot-product: $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$.



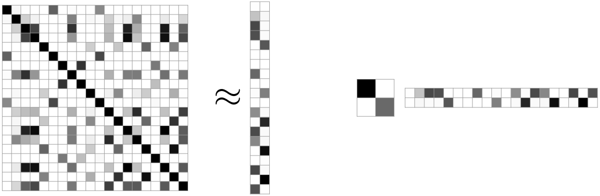
$\overset{\phi}{\longrightarrow}$

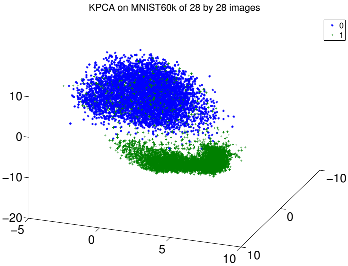


High-dimensional mapping ϕ can be automatically provided by k, i.e. *kernel trick*.

Kernel Methods for Nonlinear Feature Extraction

Kernel PCA [Scholkopf 1996]: eigendecompose ($K = U \Sigma U^T$)

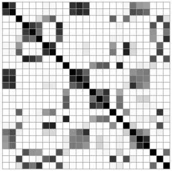




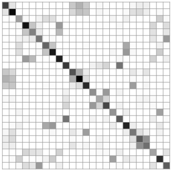
Kernel Methods for Regression

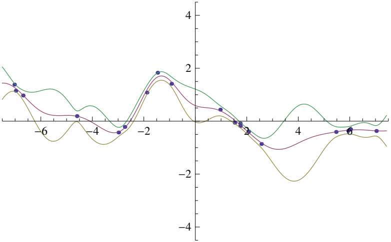
Gaussian process regression: solve a large linear system.

$K^{-1}y$



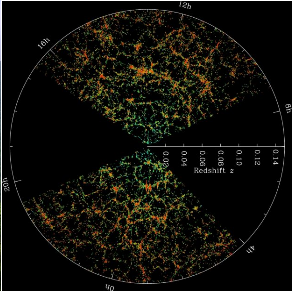
$\Rightarrow$





Scientific Motivation for Kernel Methods

Sloan Digital Sky Survey:



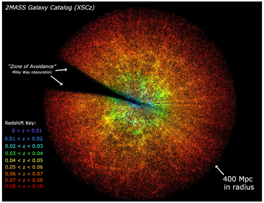
Collects around 200 GB of data/day. Has collected photometric data of around 500 M objects and spectra for more than 1 M objects.

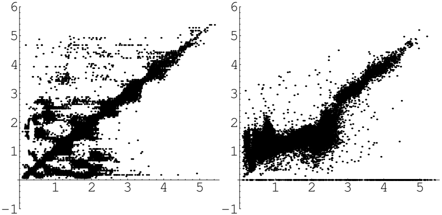
Scientific Motivation for Kernel Methods: Redshift Prediction

Kernel density estimator: $\operatorname{map}_{q \in Q} \sum_{r_j \in R} w_j k(q, r_j)$

Nadaraya-Watson: $\operatorname{map}_{q \in Q} \frac{\sum_{(r_j, y_j) \in R} w_j y_j k(q, r_j)}{\sum_{r_j \in R} w_j k(q, r_j)}$

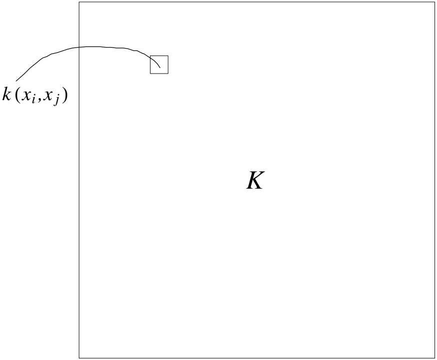
Gaussian process regression: $K^{-1} y$





Large-scale redshift prediction of galaxies and quasars.

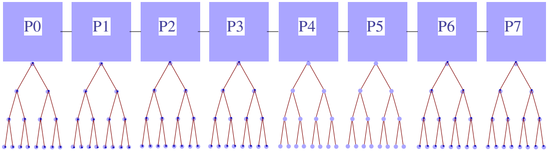
Problem: Scaling Kernel Methods



Parameter optimization is computationally intensive.

- Weighted column average (KDE): $O(N^2)$ / parameter.
- Eigendecomposition (KPCA): $O(N^3)$ / parameter.
- Solving linear systems (GPR): $O(N^3)$ / parameter.

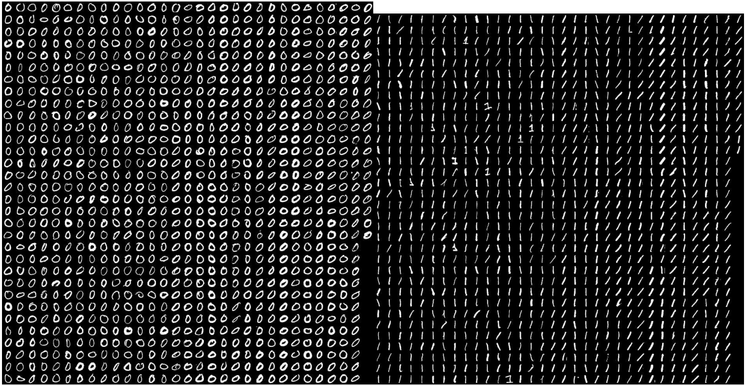
Problem: Scaling Kernel Methods



Depending on the kernel (long-range), the communication could be a bottleneck.

- Weighted column average (KDE): $O(N^2)$ / parameter.
- Eigendecomposition (KPCA): $O(N^3)$ / parameter.
- Solving linear systems (GPR): $O(N^3)$ / parameter.

Problem: Scaling Kernel Methods



MNIST Handwritten digit recognition data of 28×28 images.
Training set: 60K points $\Rightarrow$ requires 28 GB to store the kernel matrix.

Problem: Scaling Kernel Methods

For the moment, assume that the kernel hyperparameters are fixed. We are given the set of query points $\mathbf{Q} = \{\mathbf{q}\}$ and the set of data points $\mathbf{R}$.
For KDE: Given $\epsilon > 0$, approximate $\Phi(\mathbf{q};\mathbf{R}) = \sum_{\mathbf{r}\in\mathbf{R}} k(\mathbf{q},\mathbf{r})$ with $\widetilde{\Phi}(\mathbf{q};\mathbf{R})$ such that $\left|\widetilde{\Phi}(\mathbf{q};\mathbf{R}) - \Phi(\mathbf{q};\mathbf{R})\right| \leq \epsilon\Phi(\mathbf{q};\mathbf{R})$ as fast as possible.

Can also put probabilistic error bounds (Chapter 5 of the thesis).

Thesis Statement

"Utilizing the best general-dimension algorithms, approximation methods with error bounds, the distributed and shared memory parallelism can help scale kernel methods."

Thesis Contributions

A parallel kernel summation framework that utilizes:

- A recursive general-dimension divide-and-conquer algorithm using various types of deterministic and probabilistic approximations (Chapter 3, 4, 5).
- Indexing the data using any multi-dimensional binary tree with both distributed memory (MPI) and shared memory (OpenMP/Intel TBB) parallelism (Chapter 8).
- A dynamic load balancing scheme to adjust work imbalances during the computation (Chapter 8).
- A combination of distributed averaging and random feature extraction for scaling GPR and KPCA (Chapter 9).

Methods That Can be Easily Parallelized Within the Framework

- Kernel density estimation.
- Kernel regression/local polynomial regression.
- Gaussian process regression.
- Kernel PCA.
- Kernel SVM (test phase).
- Kernel k-means.
- Kernel conditional density estimation.
- Kernel orthogonal centroid.
- Kernel (...).

Earlier Related Work

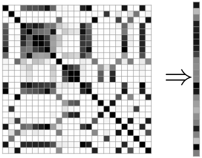
- "A framework for parallel tree-based scientific simulations," in Proceedings of 26 th International Conference on Parallel Processing, pp. 137-144, 1997. [Liu and Wu 1997]
- THOR: A Parallel N-body Data Mining Framework [Boyer, Riegel, Gray 2007]
- The PEGASUS peta-scale graph mining framework [Kang, Tsourakakis, Faloutsos 2009]

Our work: more comprehensive arsenal of approximation methods + data structures; uses MPI/OpenMP/Intel TBB from the ground-up.
Caveat: still a work in progress.

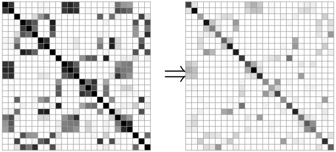
The First Part of the Talk

Density estimation
KDE: $O\left(N^2\right)$ / parameter.

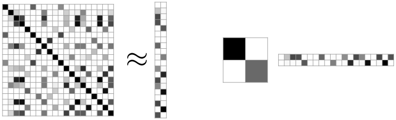
Clustering
Mean shift
$O\left(N^2\right)$ / iter / parameter.



Regression: GPR.
$O(N^3)$ / parameter.



Nonlinear feature extraction: KPCA.
$O(N^3)$ / parameter.

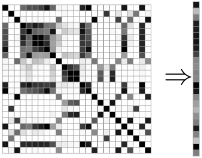


The Other Two

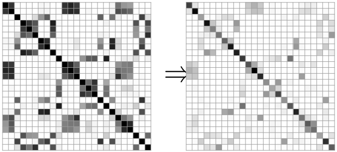
Claim: GPR/KPCA computations $\approx$ kernel summations

Density estimation
KDE: $O\left(N^2\right)$ / parameter.

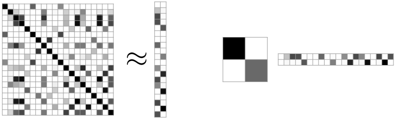
Clustering
Mean shift
$O\left(N^2\right)$ / iter / parameter.



Regression: GPR.
$O(N^3)$ / parameter.



Nonlinear feature extraction: KPCA.
$O(N^3)$ / parameter.



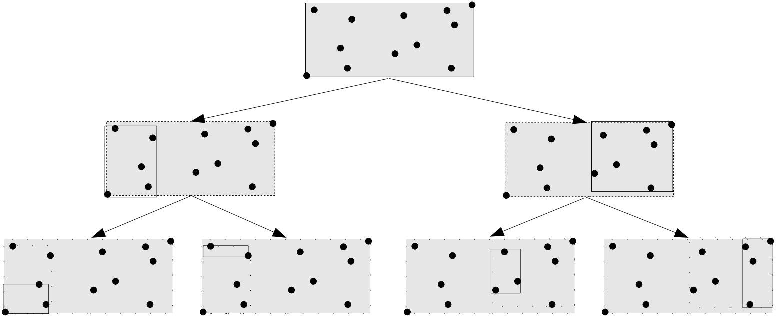
Outline

Contribution 1: Unified Framework for Kernel Summation

- Use multidimensional trees for full generality.
- Divide-and-conquer using trees via approximations.

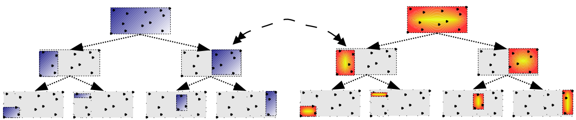
Multidimensional Trees

kd-tree [Bentley 1975]: recursively split the data points using an axis-aligned split.



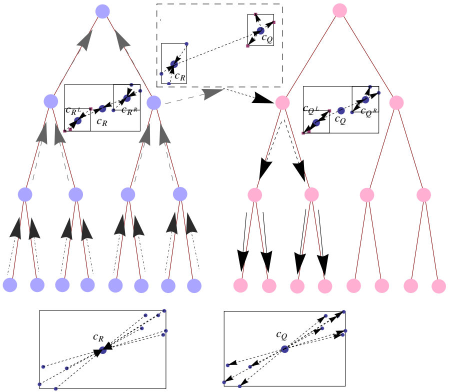
Generalized N-body Framework

[Gray/Moore 2000] For problems of the form $\bigoplus_{q \in Q} \bigotimes_{r \in R} k(q, r)$ where $\oplus$ and $\otimes$ are associative, commutative binary operators:



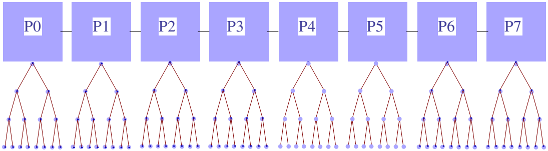
DualTree(Q, R)
if CanSummarize(Q, R) **then**
  Summarize(Q, R)
else
  if Q and R are leaf nodes **then**
    DualTreeBase(Q, R)
  else
    DualTree(Q^L, R^L),DualTree(Q^L, R^R)
    DualTree(Q^R, R^L),DualTree(Q^R, R^R)
  end if
end if

Contribution 2: Deterministic/Probabilistic Approximations



My research prior to year 2008 has focused on this aspect (Chapter 3, 4, 5 of the thesis); 16K lines of open-sourced C++ series expansion code.

Parallelizing Kernel Summations

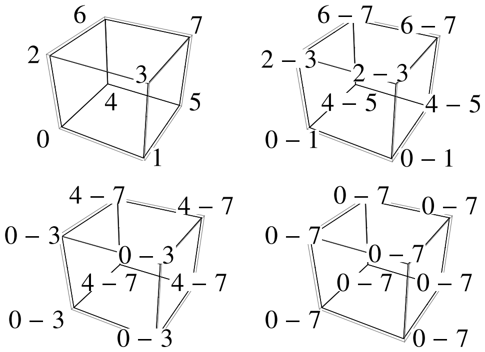


Based on the work (Chapter 8 of the thesis):

> D. Lee, R. Vuduc, and A. G. Gray. A Distributed Kernel Summation Framework for General-Dimension Machine Learning. In Proceedings of SIAM International Conference on Data Mining, 2012. Best Paper Award.

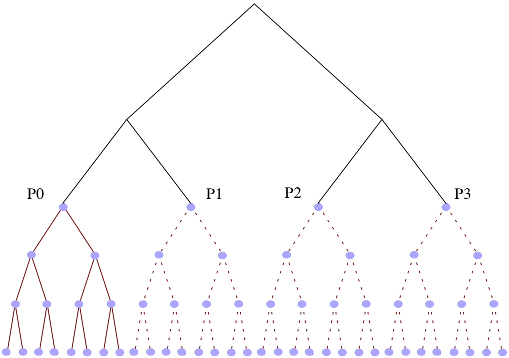
Extended version invited for a SIAM journal.

Recursive Doubling



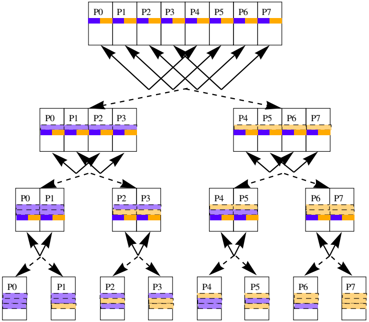
- Used for constructing trees in parallel.
- Passing data/messages among MPI processes.

Contribution 3: Distributed Tree



Black line: the global tree shared by all processes.
Red line: the local tree owned by process $P0$.
Each process owns the global tree and its local tree.

Distributed Memory Parallelism in Tree Building

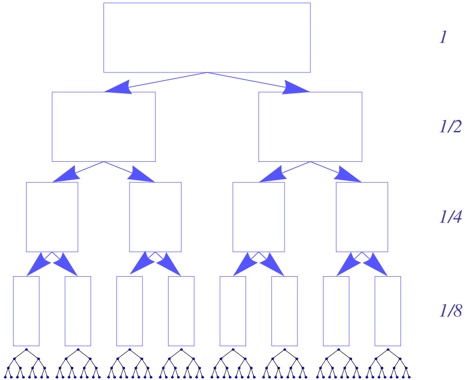


Build the first $\log p$ levels in parallel.

Overall complexity on the hypercube topology:

$$\mathcal{O}\left(\frac{DN}{p}\log\left(\frac{N}{p}\right)\right) + \mathcal{O}\left(Dt_w m_{bound}\left(2p - \log 4p\right)\right) + \mathcal{O}\left(\frac{2N(D+t_w)}{p}\log p\right) + \mathcal{O}\left(\frac{t_s}{2}\log p\left(\log p + 3\right)\right)$$

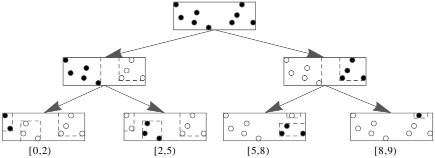
Shared Memory Parallelism in Tree Building

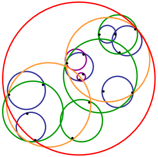


Assign available number of threads to reduction processes necessary in computing the bounding primitive.

What Can be Indexed in Parallel?

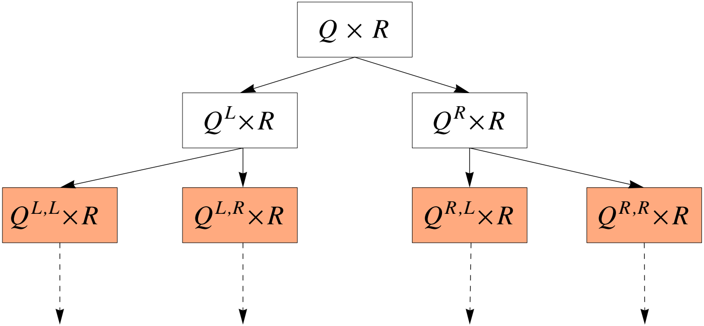
Any multi-dimensional binary tree.





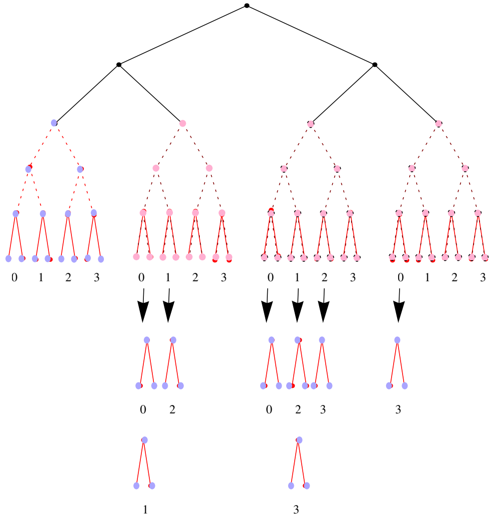
Tree type	Bound type	$\textsc{Rule}(x)$
kd-trees	hyper-rectangle $\{b_{d,\min}, b_{d,\max}\}_{d=1}^{D}$	$x_i \leq s_i$ for $1 \leq i \leq D$, $b_{d,\min} \leq s_i \leq b_{d,\max}$
metric trees	hyper-sphere $B(c,r)$, $c \in \mathbb{R}^D$, $r > 0$	$\|\|x - p_{left}\|\| < \|\|x - p_{right}\|\|$ for $p_{left}, p_{right} \in \mathbb{R}^D$

Contribution 4: Parallelization



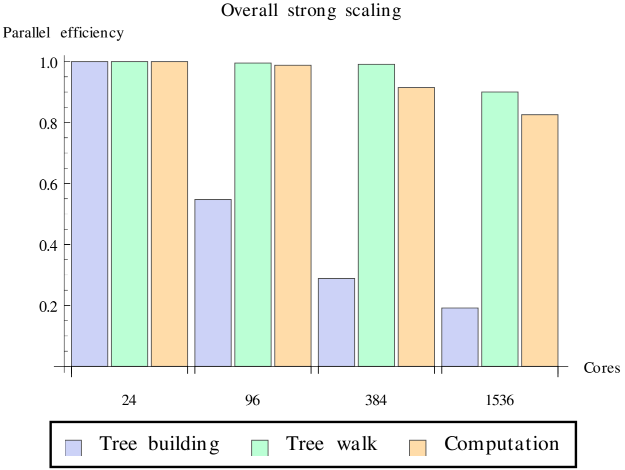
Pre-divide and spawn off independent computations.

Queueing up Independent Tasks



Overall Strong Scaling

10 million subset of SDSS Data Release 6.



Raw numbers: (13.52, 339.36, 2371), (7.41, 24.38, 244), (2.93, 2.78, 98.78), (1.10, 0.27, 39.51)

Analysis of Strong Scaling

Building a distributed multidimensional tree is different from building a geometrically constrained data structure such as octrees (communication costs in green).

$$\mathcal{O}\left(\frac{DN}{p}\log\left(\frac{N}{p}\right)\right) + \mathcal{O}\left(D t_w m_{bound}\left(2p - \log 4p\right)\right)$$
$$+\mathcal{O}\left(\frac{2N(D+t_w)}{p}\log p\right) + \mathcal{O}\left(\frac{t_s}{2}\log p\left(\log p + 3\right)\right)$$

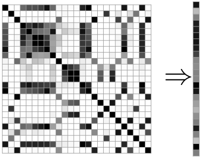
- Requires multiple all-reduce operations which hurt scalability.
- It is possible to trade the quality of the constructed tree for scalablity.

Outline

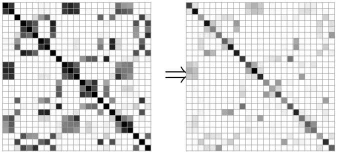
GPR and KPCA

Density estimation
KDE: $O\left(N^2\right)$ / parameter.

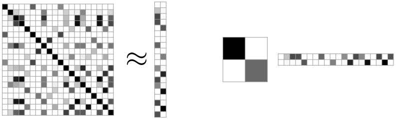
Clustering
Mean shift
$O\left(N^2\right)$ / iter / parameter.



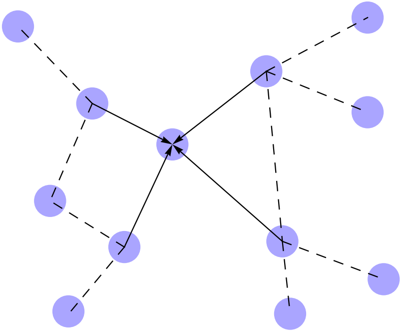
Regression: GPR.
$O(N^3)$ / parameter.



Nonlinear feature extraction: KPCA.
$O(N^3)$ / parameter.



Distributed Averaging

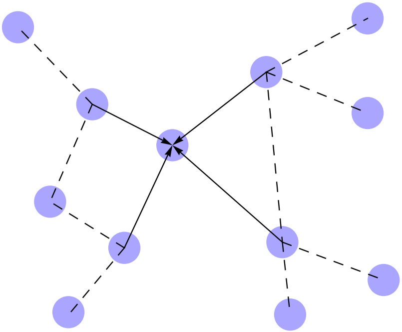


Each process maintains an estimate of the global average of the numbers in the network and iterates the following difference equation.

$$\mu_i(t+1) = \mu_i(t) + \epsilon \sum_{j \in \mathcal{N}_i} (\mu_j(t) - \mu_i(t))$$

Each state on each process converges to the average of the initial values on all nodes.

Distributed Averaging

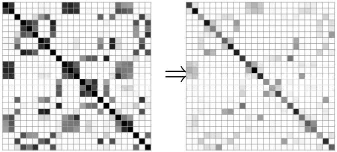


Popular technique used in literature on algorithms running on wireless sensor networks. A type of gossip-based algorithm.

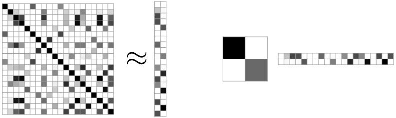
Contribution 5: Distributed Averaging + Random Features

Use random features to linearize the problem and apply distributed averaging on the localized averages (Chapter 9 of the thesis).

Regression: GPR.
$O(N^3)$ / parameter.

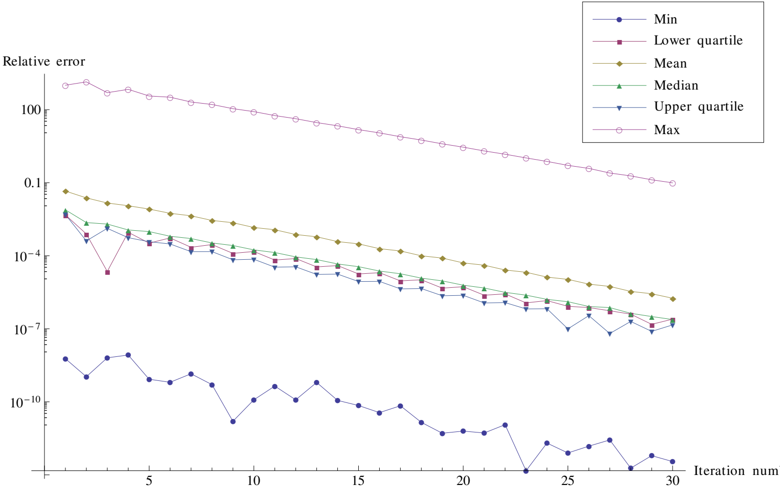


Nonlinear feature extraction: KPCA.
$O(N^3)$ / parameter.

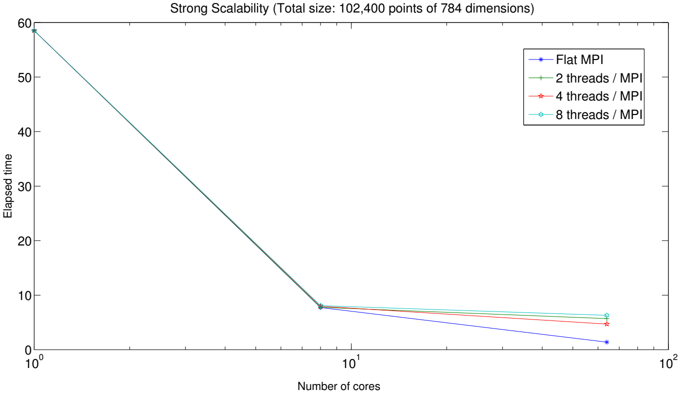


Distributed Averaging GPR

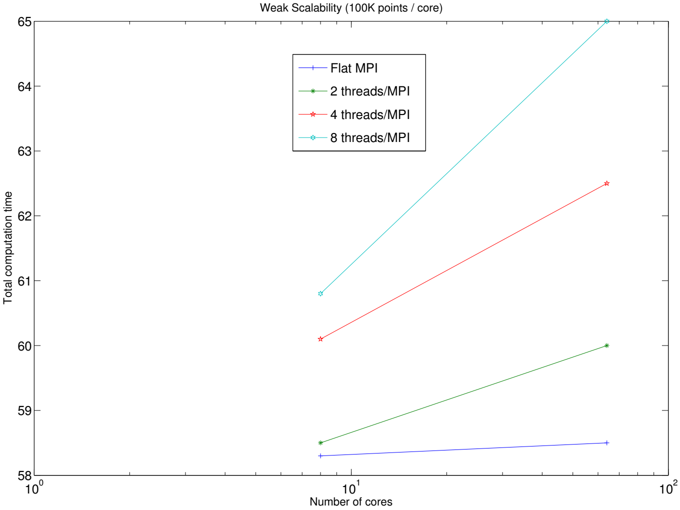
Plot of the relative error distribution between the centralized estimates and the decentralized estimates.



Distributed KPCA: Strong Scaling



Distributed KPCA: Weak Scaling



Outline

Conclusion

Summary of contributions:

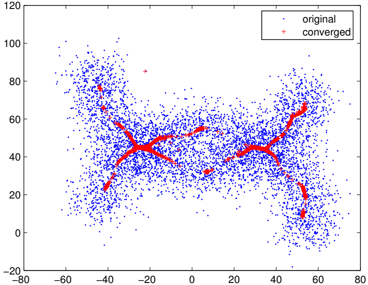
- Parallel multidimensional trees.
- Adopted algorithmic strategies from wide range of fields: distributed averaging (networking), random features (machine learning), series expansion (computational physics).
- Open-source contribution: MLPACK - more than 45K+ lines of contributed code. Paving the way for the next generation.

"Utilizing the best general-dimension algorithms, approximation methods with error bounds, the distributed and shared memory parallelism can help scale kernel methods."

List of Other Research Work

Included in the thesis, but omitted in this presentation:

Fast nonparametric clustering (Chapter 6): with Ping Wang, James M. Rehg, Alexander G. Gray. AISTATS 2007.

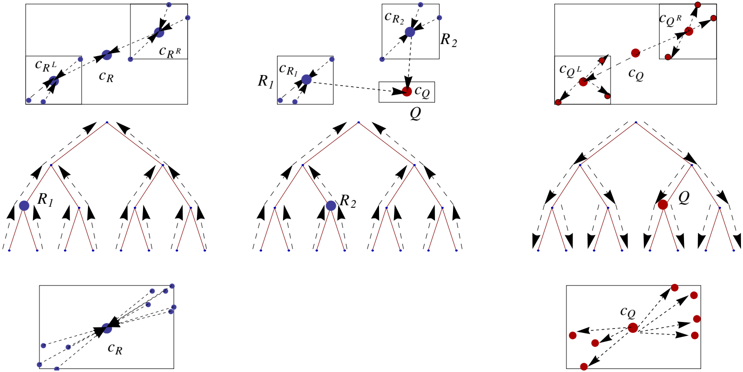


$$\operatorname*{map}_{q \in Q} \frac{\sum_{r_j \in R} w_j r_j k(q, r_j)}{\sum_{r_j \in R} w_j k(q, r_j)}$$

List of Other Research Work

Included in the thesis, but omitted in this presentation:
Higher-order extension of the kernel summation (Chapter 7):

with Arkadas Ozakin, Alexander G. Gray. Submitted to Journal of Computational Physics.



Collaborators

- Professor Edmond Chow (Georgia Tech)
- Professor Alexander G. Gray (Georgia Tech)
- Kihwan Kim (NVIDIA Research)
- Professor Richard Vuduc (Georgia Tech)
- William March (Georgia Tech)
- Nishant Metha (Georgia Tech)
- Hua Ouyang (Georgia Tech)
- Parikshit Ram (Georgia Tech)
- Ryan Riegel (Georgia Tech)
- Nikolaos Vasiloglou (Georgia Tech)

List of Other Collaborations

Published but not included in the thesis chapters:

- Run-time analysis of N-body problems: with Parikshit Ram, William B. March, Alexander G. Gray. NIPS 2009.
- Rank-approximate NN: With Parikshit Ram, Hua Ouyang, Alexander G. Gray. NIPS 2009.
- GPR for motion trajectory analysis: With Kihwan Kim, Irfan Essa. ICCV 2011. (post-proposal).
- Time-constrained NN: With Parikshit Ram, Alexander G. Gray. SDM 2012. (post-proposal).
- GPR for camera motion automation: With Kihwan Kim, Irfan Essa. CVPR 2012. (post-proposal).

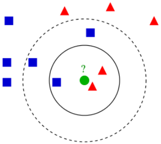
On-going collaborations involving significant code transfer:

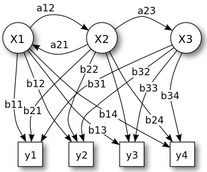
- Parallel n-point correlation: With William B. March et al.

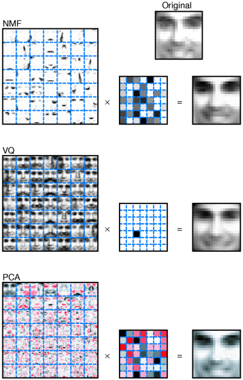
Other collaborations omitted...

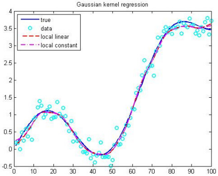
MLPACK: Open-source ML

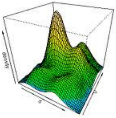
NIPS 2008 Demonstration, NIPS 2011 Big Learning Workshop

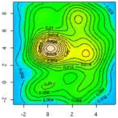












Pre-Proposal Publication List

Dongryeol Lee, Alexander G. Gray, and Andrew W. Moore. Dual-Tree Fast Gauss Transforms. In: Advances in Neural Information Processing Systems, 2005.

Dongryeol Lee and Alexander G. Gray. Faster Gaussian Summation: Theory and Experiment. In: Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence, 2006.

Ping Wang, Dongryeol Lee, Alexander G. Gray, and James M. Rehg. Fast Mean Shift with Accurate and Stable Convergence. In: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, 2007.

Dongryeol Lee and Alexander G. Gray. Fast High-dimensional Kernel Summations Using the Monte Carlo Multipole Method, In: Advances in Neural Information Processing Systems, 2008.

Dongryeol Lee, Alexander G. Gray, and Andrew W. Moore. Dual-Tree Fast Gauss Transforms (arXiv).

Parikshit Ram, Dongryeol Lee, Hua Ouyang, and Alexander G. Gray. Rank-Approximate Nearest Neighbor Search: Retaining Meaning and Speed in High Dimensions. In: Advances in Neural Information Processing Systems, 2009.

Parikshit Ram, Dongryeol Lee, William B. March, and Alexander G. Gray. Linear-time Algorithms for Pairwise Statistical Problems. In: Advances in Neural Information Processing Systems, 2009. Spotlight Presentation.

Post-Proposal Publication List

Dongryeol Lee, Arkadas Ozakin, and Alexander G. Gray. Multibody Multipole Methods. In Journal of Computational Physics, 2011.

Kihwan Kim, Dongryeol Lee, and Irfan Essa. Gaussian Process Regression Flow for Analysis of Motion Trajectories. In: Proceedings of IEEE International Conference on Computer Vision, 2011.

William B. March, Arkadas Ozakin, Dongryeol Lee, Ryan Riegel, and Alexander G. Gray. Multi-Tree Algorithms for Large-Scale Astrostatistics. In Advances in Machine Learning and Data Mining for Astronomy, Chapman and Hall/CRC Press, 2012.

Dongryeol Lee, Richard Vuduc, and Alexander G. Gray. A Distributed Kernel Summation Framework for General-Dimension Machine Learning. To appear in SIAM International Conference on Data Mining, 2012. Best Paper Award.

Parikshit Ram, Dongryeol Lee, and Alexander G. Gray. Nearest-Neighbor Search on a Time Budget via Max-Margin Trees. To appear in SIAM International Conference on Data Mining, 2012.

Kihwan Kim, Dongryeol Lee, and Irfan Essa. Detecting Regions of Interest in Dynamic Scenes for Camera Motion. To appear in IEEE Conference on Computer Vision and Pattern Recognition, 2012.

William B. March, Kent Czechowski, Marat Dukhan, Thomas Benson, Dongryeol Lee, Andy J. Connolly, Richard Vuduc, Edmond Chow, and Alexander G. Gray. Optimizing the Computation of N-Point Cor- relations on Large-Scale Astronomical Data. To appear in Proc. ACM/IEEE Conf. Supercomputing (SC), 2012

Nishant A. Mehta, Dongryeol Lee, and Alexander G. Gray. Minimax Multi-Task Learning and a Generalized Loss-Compositional Paradigm for MTL. To appear in Advances in Neural Information Processing Systems, 2012.