Computational Methods for Large-Scale Data Analysis

Alexander Gray
Georgia Institute of Technology
College of Computing

FASTlab: Fundamental Algorithmic and Statistical Tools
Is science in 2008 different from science in 1908?

Instruments

**Fig. 1.** Telescope area doubles every 25 years, whereas telescope CCD pixels double every 2 years. This rate seems to be accelerating. 

[Science, Szalay & J. Gray, 2001]
Is science in 2008 different from science in 1908?

**Instruments**

![Graph showing telescope area doubles every 25 years, whereas telescope CCD pixels double every 2 years. This rate seems to be accelerating.](image)

*Fig. 1. Telescope area doubles every 25 years, whereas telescope CCD pixels double every 2 years. This rate seems to be accelerating. [Science, Szalay & J. Gray, 2001]*

**Data: CMB Maps**

- 1990 COBE 1,000
- 2000 Boomerang 10,000
- 2002 CBI 50,000
- 2003 WMAP 1 Million
- 2008 Planck 10 Million

**Data: Local Redshift Surveys**

- 1986 CfA 3,500
- 1996 LCRS 23,000
- 2003 2dF 250,000
- 2005 SDSS 800,000

**Data: Angular Surveys**

- 1970 Lick 1M
- 1990 APM 2M
- 2005 SDSS 200M
- 2008 LSST 2B
Sloan Digital Sky Survey (SDSS)
1 billion objects
144 dimensions

(~250M galaxies in 5 colors,
~1M 2000-D spectra)

Size matters!  Now possible:
• low noise: subtle patterns
• global properties and patterns
• rare objects and patterns
• more info: 3d, deeper/earlier, bands
• in parallel: more accurate simulations
• 2008: LSST – time-varying phenomena
Happening everywhere!

- Molecular biology
- Drug discovery
- Earth sciences
- Physical simulation
- Neuroscience
- Internet

- Microarray chips
- Nuclear magnetic resonance
- Satellite topography
- Microprocessor
- Functional MRI
- Fiber optics

- Internet properties chart
1. How did galaxies evolve?
2. What was the early universe like?
3. Does dark energy exist?
4. Is our model (GR+inflation) right?
1. How did galaxies evolve?
2. What was the early universe like?
3. Does dark energy exist?
4. Is our model (GR+inflation) right?

- Kernel density estimator
- n-point spatial statistics
- Nonparametric Bayes classifier
- Support vector machine
- Nearest-neighbor statistics
- Gaussian process regression
- Hierarchical clustering

A. Connolly, U. Pitt Physics
C. Miller, NOAO
R. Brunner, NCSA
R. Scranton, U. Pitt Physics
M. Balogh, U. Waterloo Physics
G. Richards, Princeton Physics
A. Szalay, Johns Hopkins Physics

Machine learning/ statistics guy
1. How did galaxies evolve?
2. What was the early universe like?
3. Does dark energy exist?
4. Is our model (GR+inflation) right?

- Kernel density estimator $O(N^2)$
- n-point spatial statistics $O(N^n)$
- Nonparametric Bayes classifier $O(N^2)$
- Support vector machine $O(N^2)$
- Nearest-neighbor statistics $O(N^2)$
- Gaussian process regression $O(N^3)$
- Hierarchical clustering $O(N^3)$

Machine learning/statistics guy
1. How did galaxies evolve?
2. What was the early universe like?
3. Does dark energy exist?
4. Is our model (GR+inflation) right?

- Kernel density estimator $O(N^2)$
- n-point spatial statistics $O(N^n)$
- Nonparametric Bayes classifier $O(N^2)$
- Support vector machine $O(N^2)$
- Nearest-neighbor statistics $O(N^2)$
- Gaussian process regression $O(N^3)$
- Hierarchical clustering $O(N^3)$

But I have 1 million points
Statistics/learning challenges

**Statistical** (modeling, validation):
- *Best performance with fewest assumptions*

**Computational:**
- *Large* \( N \) (*#data*), \( D \) (*#features*)
Statistics/learning challenges

**Statistical** (modeling, validation):

- *Best performance with fewest assumptions*

**Computational:**

- *Large $N$ (#data), $D$ (#features), $M$ (#models)*
Statistics/learning challenges

**Statistical** (modeling, validation):
- Best performance with fewest assumptions

**Computational:**
- Large $N$ (#data), $D$ (#features), $M$ (#models)

Reduce? Simplify? Poor modeling
Statistics/learning challenges

**Statistical** (modeling, validation):
- *Best performance with fewest assumptions*

**Computational:**
- *Large $N$ (#data), $D$ (#features), $M$ (#models)*

Reduce? Simplify? **Poor modeling**
Avoid hard problems? **Poor funding**
My motivating datasets

• 1993-1999: POSS-II
• 1999-2008: SDSS
• Coming: Pan-STARRS, LSST
• Also:
  – Millennium simulation data
  – Large Hadron Collider data
  – network traffic (email) data
  – Inbio ecology data
What I like to think about...

• The statistical problems and methods needed for answering scientific questions

• The computational problems and methods involved in scaling all of them up to big datasets

• MLPACK: software for large-scale machine learning (later in 2008)
1. What are some of the **statistical problems and methods** to consider?
2. What are some of the **computational problems and methods** to consider?
3. What might the **software** which implements all this look like?
1. What are some of the statistical problems and methods to consider?

2. What are some of the computational problems and methods to consider?

3. What might the software which implements all this look like?
10 data analysis problems, and scalable tools we’d like for them

1. **Querying** *(e.g. characterizing a region of space, defining a trigger):* nearest-neighbor, spherical range-search, orthogonal range-search

2. **Density estimation** *(e.g. comparing galaxy types):* kernel density estimation, mixture of Gaussians

3. **Regression** *(e.g. optical redshifts):* linear regression, kernel regression, Gaussian process regression
10 data analysis problems, and scalable tools we’d like for them

4. **Classification** *(e.g. quasar detection, star-galaxy separation)*: nearest-neighbor classifier, nonparametric Bayes classifier, support vector machine

5. **Dimension reduction** *(e.g. galaxy characterization)*: principal component analysis, kernel PCA, maximum variance unfolding

6. **Outlier detection** *(e.g. new object types, data cleaning)*: by robust L₂ estimation, by density estimation, by dimension reduction
10 data analysis problems, and scalable tools we’d like for them

7. **Clustering** *(e.g. automatic Hubble sequence)*: k-means, hierarchical clustering (“friends-of-friends”), by dimension reduction

8. **Time series analysis** *(e.g. asteroid tracking, variable objects)*: Kalman filter, hidden Markov model, trajectory tracking

9. **2-sample testing** *(e.g. cosmological validation)*: n-point correlation

10. **Cross-match** *(e.g. multiple databases)*: bipartite matching
1. What are some of the statistical problems and methods to consider?

2. What are some of the computational problems and methods to consider?

3. What might the software which implements all this look like?
Core computational problems

What are the basic mathematical operations, or bottleneck subroutines, can we focus on developing fast algorithms for?
Core computational problems

• Aggregations
• Generalized N-body problems
• Graphical model inference
• Linear algebra
• Optimization
Core computational problems

Aggregations, GNPs, graphical models, linear algebra, optimization

- **Querying**: nearest-neighbor, sph range-search, ortho range-search
- **Density estimation**: kernel density estimation, mixture of Gaussians
- **Regression**: linear regression, kernel regression, Gaussian process regression
- **Classification**: nearest-neighbor classifier, nonparametric Bayes classifier, support vector machine
- **Dimension reduction**: principal component analysis, kernel PCA, maximum variance unfolding
- **Outlier detection**: by robust L₂ estimation, by density estimation, by dimension reduction
- **Clustering**: k-means, hierarchical clustering (“friends-of-friends”), by dimension reduction
- **Time series analysis**: Kalman filter, hidden Markov model, trajectory tracking
- **2-sample testing**: n-point correlation
- **Cross-match**: bipartite matching
Aggregations

• **How it appears:** nearest-neighbor, sph range-search, ortho range-search

• **Common methods:** locality sensitive hashing, kd-trees, metric trees, disk-based trees

• **Mathematical challenges:** high dimensions, provable runtime

• **Mathematical topics:** computational geometry, randomized algorithms
Aggregations

• **Interesting method:** *Cover-trees [Beygelzimer et al 2004]*
  – Provable runtime
  – Consistently good performance, even in higher dimensions

• **Interesting method:** *Learning trees [Cayton et al 2007]*
  – Learning data-optimal data structures
  – Improves performance over kd-trees

• **Interesting method:** *MapReduce [Google]*
  – Brute-force
  – But makes HPC automatic for a certain problem form
Generalized N-body Problems

• **How it appears:** kernel density estimation, mixture of Gaussians, kernel regression, Gaussian process regression, nearest-neighbor classifier, nonparametric Bayes classifier, support vector machine, kernel PCA, hierarchical clustering, trajectory tracking, n-point correlation

• **Common methods:** FFT, Fast Gauss Transform, Well-Separated Pair Decomposition

• **Mathematical challenges:** high dimensions, strong error guarantee

• **Mathematical topics:** approximation theory, computational physics
Generalized N-body Problems

• **Interesting method:** Generalized Fast Multipole Method, aka multi-tree methods [Gray et al. 2000-2008]
  - Fastest practical algorithms for most of the problems to which it has been applied
  - Hard relative error bounds
  - Automatic parallelization (*THOR: Tree-based Higher-Order Reduce*)
Graphical model inference

• **How it appears:** hidden Markov models, bipartite matching

• **Common methods:** belief propagation, expectation propagation

• **Mathematical challenges:** large cliques, upper and lower bounds, graphs with loops

• **Mathematical topics:** variational methods, statistical physics, turbo codes
Graphical model inference

• **Interesting method:** *Survey propagation* [Mezard et al 2002]
  – Good results for combinatorial optimization
  – Based on statistical physics ideas

• **Interesting method:** *Expectation propagation* [Minka 2001]
  – Variational method based on moment-matching idea
Linear algebra

• **How it appears:** linear regression, Gaussian process regression, PCA, kernel PCA, Kalman filter

• **Common methods:** QR, Krylov

• **Mathematical challenges:** numerical stability, sparsity preservation

• **Mathematical topics:** linear algebra
Linear algebra

• **Interesting method:** *Monte Carlo SVD* [Frieze, Drineas, et al. 1998-2008]
  – Sample either columns or rows, from squared length distribution
  – For rank-k matrix approx; must know k

• **Interesting method:** *QUIC-SVD* [Holmes, Gray, Isbell 2008]
  – Sample using cosine trees and stratification
  – Automatically sets rank based on desired error
Optimization

• **How it appears:** support vector machine, maximum variance unfolding, robust $L_2$ estimation

• **Common methods:** interior point, Newton’s method

• **Mathematical challenges:** large number of variables / constraints

• **Mathematical topics:** optimization theory, linear algebra, convex analysis
Optimization

• Interesting method: Sequential minimization optimization (SMO) [Platt 1999]
  – Much more efficient than interior-point, for SVM QPs

• Interesting method: Stochastic quasi-Newton [Schraudolph 2007]
  – Does not require scan of entire data
Interaction between statistics and computation

• Explicitly trade off between statistical accuracy and runtime

• Monte Carlo: a statistical idea for computational purposes

• Active learning, aka design of experiments: choose the important points
1. What are some of the statistical problems and methods to consider?

2. What are some of the computational problems and methods to consider?

3. What might the software which implements all this look like?
Keep in mind the machine

- **Memory hierarchy:** cache, RAM, out-of-core
- Dataset bigger than one machine: parallel/distributed
- Everything is becoming **multicore**
Keep in mind the overall system

- **Databases** can be more useful than ASCII files
- **Workflows** can be more useful than brittle perl scripts
- **Visual analytics** connects visualization/HCI with data analysis
Keep in mind the software complexity

- Automatic **code generation** (e.g. MapReduce)
- Automatic **tuning** (e.g. OSKI)
- Automatic **algorithm derivation** (e.g. AutoBayes, SPIRAL)
Our upcoming products

• **MLPACK**: “the LAPACK of machine learning” – Dec. 2008

• **THOR**: “the MapReduce of Generalized N-body Problems” – Apr. 2009

• **Algorithmica**: Automatic derivation of the above – 2010
The end

Always looking for collaborators, challenging applications, and generous funding!

Alexander Gray
agray@cc.gatech.edu
Goal of this talk:
Make our best methods fast!

- kernel density estimator
- n-point statistics
- nonparametric Bayes classifier
- support vector machine
- nearest neighbor statistics
- Gaussian process regression
- Bayesian inference
- …
“What’s the distribution?”

1. warm-up: generalized histogram
2. n-point statistics
3. kernel density estimator
4. general strategy: multi-tree
   - nonparametric Bayes classifier
   - support vector machine
   - nearest neighbor statistics
   - Gaussian process regression
5. science!
Comparing: “Same distribution?”

1. **warm-up:** generalized histogram

2. **n-point statistics**

3. **kernel density estimator**

   strategy:
   - nonparametric Bayes classifier
   - support vector machine
   - nearest neighbor statistics
   - Gaussian process regression
   - Bayesian inference
These are all “Generalized N-body problems”

2. n-point statistics
3. kernel density estimator
4. **general strategy: multi-tree**
   1. nonparametric Bayes classifier
   2. support vector machine
   3. nearest neighbor statistics
   4. Gaussian process regression
   5. Bayesian inference

5. science!
1. warm-up: generalized histogram
2. n-point statistics
3. kernel density estimator
4. general strategy:
   1. nonparametric Bayes classifier
   2. support vector machine
   3. nearest neighbor statistics
   4. Gaussian process regression
   5. Bayesian inference
5. science!
Special case of 2 and 3

1. **warm-up**: generalized histogram
2. n-point statistics
3. kernel density estimator
4. **general strategy**: multi-tree
   1. nonparametric Bayes classifier
   2. support vector machine
   3. nearest neighbor statistics
   4. Gaussian process regression
   5. Bayesian inference
5. science!
Histogram (1-D)
Generalized histogram (1-D)

\[ \hat{f}(q) \propto \sum_j I(|q - x_j| < h) \]
Generalized histogram

\[ \hat{f}(q) \propto \sum_j I \left( \| q - x_j \| < h \right) \]
How can we compute this efficiently?

**kd-trees:**

most widely-used space-partitioning tree

[Bentley 1975], [Friedman, Bentley & Finkel 1977], [Moore & Lee 1995]
A $kd$-tree: level 1
A *kd*-tree: level 2
A \textit{kd}-tree: level 3
A *kd*-tree: level 4
A *kd*-tree: level 5
A \textit{kd}-tree: level 6
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm

Pruned!
(inclusion)
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm

Pruned!
(exclusion)
Range-count recursive algorithm
Range-count recursive algorithm
Range-count recursive algorithm

fastest practical algorithm
[Bentley 1975]

our algorithms can use any tree
OUTLINE

1. warm-up: generalized histogram

2. n-point statistics

3. kernel density estimator

4. general strategy: multi-tree

   - nonparametric Bayes classifier
   - support vector machine
   - nearest neighbor statistics
   - Gaussian process regression
   - Bayesian inference

5. science!
Characterization of an entire distribution?

2-point correlation

“How many pairs have distance < r ?”

\[
\sum_{i}^{N} \sum_{j \neq i}^{N} I(\|x_i - x_j\| < r)
\]
The \( n \)-point correlation functions

- **Spatial inferences**: filaments, clusters, voids, homogeneity, isotropy, 2-sample testing, …

- **Foundation** for theory of point processes [Daley, Vere-Jones 1972], unifies spatial statistics [Ripley 1976]

- **Used heavily** in biostatistics, cosmology, particle physics, statistical physics

**2pcf definition:**

\[
dP = \lambda^2 dV_1 dV_2 [1 + \xi(r)]
\]

**3pcf definition:**

\[
dP = \lambda^3 dV_1 dV_2 dV_3 \cdot [1 + \xi(r_{12}) + \xi(r_{23}) + \xi(r_{13}) + \zeta(r_{12}, r_{23}, r_{13})]
\]
3-point correlation

“How many triples have pairwise distances < r?”

\[
\sum_{i}^{N} \sum_{j \neq i}^{N} \sum_{k \neq j \neq i}^{N} I(\delta_{ij} < r_1) I(\delta_{jk} < r_2) I(\delta_{ki} < r_3)
\]
How can we count $n$-tuples efficiently?

“How many triples have pairwise distances < $r$ ?”
Use $n$ trees!

[Gray & Moore, NIPS 2000]
“How many valid triangles a-b-c (where \(a \in A, \ b \in B, \ c \in C\)) could there be?

\[
\text{count}\{A,B,C\} = \ ?
\]
“How many valid triangles a-b-c (where \( a \in A, \ b \in B, \ c \in C \) could there be?

\[
\text{count}\{A,B,C\} = \text{count}\{A,B,C\.left\} + \text{count}\{A,B,C\.right\}
\]
“How many valid triangles a-b-c (where \( a \in A, \ b \in B, \ c \in C \)) could there be?

\[
\text{count}\{A,B,C\} = \text{count}\{A,B,C.left\} + \text{count}\{A,B,C.right\}
\]
“How many valid triangles a-b-c (where \( a \in A, \ b \in B, \ c \in C \)) could there be?

count\{A,B,C\} = ?
“How many valid triangles $a$-$b$-$c$ (where $a \in A$, $b \in B$, $c \in C$) could there be?

\[
\text{count}\{A,B,C\} = 0!
\]
“How many valid triangles $a$-$b$-$c$ (where $a \in A$, $b \in B$, $c \in C$) could there be?

\[
\text{count}\{A,B,C\} = ?
\]
“How many valid triangles a-b-c (where \( a \in A, \ b \in B, \ c \in C \)) could there be?

\[
\text{count}\{A,B,C\} = |A| \times |B| \times |C|
\]
Key idea
(combinatorial proximity problems):

for $n$-tuples:

$n$-tree recursion
Exclusion and inclusion on **multiple radii** simultaneously

Find the largest radius which gives exclusion: binary search
Exclusion and inclusion on **multiple radii** simultaneously

Find the largest radius which gives exclusion: binary search
Exclusion and inclusion on \textbf{multiple radii} simultaneously

Recurse on the remaining radii
Key idea
(combinatorial proximity problems):

**multi-radius recursion**
(two layers of recursion)
n-point correlations: problem status

• 50-year-old problem [Peebles, 1956]

• main proposals:
  – FFT [Peebles and Groth 76] (approximate)
    • must interpolate to equi-spaced grid points
    • \( n=2: O(W^D \log W^D), \, n=3: O(W^D (W^D \log W^D)) \)
    • Case 1: no error bounds
    • Fourier ringing at edges
  – counts-in-cells (grid) [Szapudi 97] \( O(W^n) \) (approximate)
    • Case 1: no error bounds
3-point runtime

(biggest previous: 20K)

VIRGO simulation data, N = 75,000,000

naïve: 5x10^9 sec. (~150 years)

multi-tree: 55 sec. (exact)

Scaling behavior with data size, by tuple order

- $n=2$: $O(N)$
- $n=3$: $O(N^{\log_3})$
- $n=4$: $O(N^2)$
But...

Depends on $r^{D-1}$.
Slow for large radii.

VIRGO simulation data,
$N = 75,000,000$

naïve: ~150 years
multi-tree:
large $h$: 24 hrs

Let’s develop a method for large radii.
$$c = p^T$$

known.

EASIER?

hard.

$$\hat{p} \pm z_{\varepsilon/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{S}}$$

no dependence on N! but it does depend on p
\[ c = p \mathbf{T} \]

\[ \hat{p} \pm z_{\varepsilon/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{S}} \]

no dependence on \( N \) but it does depend on \( p \)
\[ c = p^T \]

\[ \hat{p} \pm z_{\varepsilon/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{S}} \]

no dependence on \( N! \) but it does depend on \( p \)
\[ c = p \, T \]

\[ \hat{p} \pm z_{\frac{\varepsilon}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{S}} \]

no dependence on N! but it does depend on \( p \)
\[ \hat{p} \pm z_{\varepsilon/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{S}} \]

no dependence on $N!$ but it does depend on $p$
This is junk:
don’t bother

c = p \ T
This is promising
Basic idea:

1. Remove some junk
   (Run exact algorithm for a while)
   \[\rightarrow \text{make } p \text{ larger}\]

2. Sample from the rest
Get disjoint sets from the recursion tree

\[ \binom{N}{n} \text{ all possible } n\text{-tuples} \]

... [prune]
Now do stratified sampling

\[
T_1 + T_2 + T_3 = T
\]

\[
\frac{T_1}{T} \hat{p}_1 + \frac{T_2}{T} \hat{p}_2 + \frac{T_3}{T} \hat{p}_3 = \hat{p}
\]

\[
\left(\frac{T_1}{T}\right)^2 \hat{\sigma}_1^2 + \left(\frac{T_2}{T}\right)^2 \hat{\sigma}_2^2 + \left(\frac{T_3}{T}\right)^2 \hat{\sigma}_3^2 = \hat{\sigma}^2
\]
**Speedup Results**

VIRGO simulation data
N = **75,000,000**

naïve: ~150 years
multi-tree:
  large h: **24 hrs**

**multi-tree monte carlo:**
  99% confidence:
  **96 sec**
Key idea
(combinatorial proximity problems):

Multi-tree Monte Carlo
n-point correlation wrapup

• Properties:
  – fastest practical exact algorithm for general $D$
  – polychromatic, general $n$
  – extends to: weighted, projected, general constraints
  – conjecture: $O(N\log N)+O(N^{\log n})$ under some conditions
  – Monte Carlo: complements exact algorithm, error bounds

• Insights: natural generalization of range-counting to $n$-tuples

• Has been used in practice [Scranton et al. 03, Kayo et al. 03, Nichol et al. 04]

• See [Gray & Moore NIPS 00], [Moore et al 00], [Gray & Moore 04].
OUTLINE

1. warm-up: generalized histogram

2. n-point statistics

3. kernel density estimator

4. general strategy: multi-tree
   - nonparametric Bayes classifier
   - support vector machine
   - neighbor statistics
   - Gaussian process regression

5. science!
Kernel density estimation

\[ \forall x_q, \quad \hat{f}(x_q) = \frac{1}{N} \sum_{r \neq q}^{N} K_h(\|x_q - x_r\|) \]
Kernel density estimation

\[ \hat{f}(x) \rightarrow f(x) \quad N \rightarrow \infty \]

- Guaranteed to converge to the true underlying density (consistency)
- Nonparametric – distribution need only meet some weak smoothness conditions
- Achieves optimal rate
- These are true given the optimal bandwidth
- Most mathematically studied and widely used general (nonparametric) density estimator
How to use a tree…

1. **How** to approximate?
2. **When** to approximate?

[Barnes and Hut, Science, 1987]

\[ \sum_{i} K(q, x_i) \approx N_R K(q, \mu_R) \]

if \( s > \frac{r}{\theta} \)
How to use a tree...

3. How to know potential error?

Let’s maintain bounds on the true kernel sum

$$
\Phi(q) \equiv \sum_i K(q, x_i)
$$

At the beginning:

$$
\begin{align*}
\Phi^{lo}(q) & \leftarrow NK^{lo} \\
\Phi^{hi}(q) & \leftarrow NK^{hi}
\end{align*}
$$

$$
\begin{align*}
\Phi^{lo}(q) & \leftarrow \Phi^{lo}(q) + N_R K(q, \delta^{lo}_{qR}) - N_R K^{lo} \\
\Phi^{hi}(q) & \leftarrow \Phi^{hi}(q) + N_R K(q, \delta^{hi}_{qR}) - N_R K^{hi}
\end{align*}
$$
How to use a tree...

4. How to do ‘all’ problem?

\[ \forall x_q, \quad \hat{f}(x_q) = \frac{1}{N} \sum_{r \neq q}^{N} K_h(\|x_q - x_r\|) \]

Single-tree:

Dual-tree (symmetric): [Gray & Moore 2000]
How to use a tree…

4. How to do ‘all’ problem?

\[ \forall q \in Q, \sum_{i} K(q, x_i) \approx N_R K(q, \mu_R) \]

if \[ s > \frac{\max(r_Q, r_R)}{\theta} \]

Generalizes Barnes-Hut to dual-tree
Key idea
(kernel summation problems):

Treat kernel summation as an extension of the basic proximity problems:

* dual-tree + simple approximation

+ bounds
BUT:
We have a tweak parameter: $\theta$

Case 1 – alg. gives no error bounds
Case 2 – alg. gives error bounds, but must be rerun
Case 3 – alg. automatically achieves error tolerance

So far we have case 2; let’s try for case 3

Let’s try to make an automatic stopping rule
Finite-difference function approximation.

Taylor expansion:

\[ f(x) \approx f(a) + f'(a)(x - a) \]

Gregory-Newton finite form:

\[ f(x) \approx f(x_i) + \frac{1}{2} \left( \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \right)(x - x_i) \]

\[ K(\delta) \approx K(\delta^{lo}) + \frac{1}{2} \left( \frac{K(\delta^{hi}) - K(\delta^{lo})}{\delta^{hi} - \delta^{lo}} \right)(\delta - \delta^{lo}) \]
Finite-difference function approximation.

assumes monotonic decreasing kernel

\[
K = \frac{1}{2} \left[ K(\delta_{QR}^{lo}) + K(\delta_{QR}^{hi}) \right]
\]

\[
\text{err}_q = \sum_{r} |K(\delta_{qr}) - \bar{K}| \leq \frac{N_R}{2} \left[ K(\delta_{QR}^{lo}) - K(\delta_{QR}^{hi}) \right]
\]

\[
\forall q, R: \frac{\text{err}_{QR}}{\phi(x_q)} \leq \frac{N_R}{N} \varepsilon \Rightarrow \forall q: \frac{\text{err}_q}{\phi(x_q)} \leq \varepsilon
\]

approximate \( \{Q,R\} \) if

\[
K(\delta_{lo}) - K(\delta_{hi}) \leq \frac{2\varepsilon}{N} \Phi_{lo}(Q)
\]
Key idea
(kernel summation problems):

Automatic error control
Kernel density estimation: problem status

• **50-year-old problem** [Rosenblatt 1953]

• main proposals:
  – FFT [Silverman 1982, 1-D], [Fan & Marron 1994, multi-D]: designed for signal processing

\[
K\left(\|q - x_j\|\right) = \frac{1}{\|q - x_j\|^a}
\]
Fast Multipole Method
[Greengard & Rokhlin 1987]

\[ O(p^D) \] and grid-based: not intended for high dimensions

FGT: no tree; IFGT \( O(D^p) \) and clusters [Yang & Duraiswami 03]

dual-tree: like high-D FMM
## colors (N=50k, D=2)

<table>
<thead>
<tr>
<th>Method</th>
<th>50% (rel. error)</th>
<th>10% (rel. error)</th>
<th>1% (rel. error)</th>
<th>0% (rel. error)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive</td>
<td>329.7</td>
<td>329.7</td>
<td>329.7 sec.</td>
<td>329.7</td>
</tr>
<tr>
<td>FFT</td>
<td>0.1</td>
<td>2.9</td>
<td>&gt; 660</td>
<td>-</td>
</tr>
<tr>
<td>IFGT</td>
<td>1.7</td>
<td>&gt; 660</td>
<td>&gt; 660</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Gaussian)</td>
<td>12.2 (65.1*)</td>
<td>18.7 (89.8*)</td>
<td>24.8 (117.2*)</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Epanechn.)</td>
<td>6.2 (6.7*)</td>
<td>6.5 (6.7*)</td>
<td>6.7 (6.7*)</td>
<td>58.2 [111.0]</td>
</tr>
</tbody>
</table>
### sj2 (N=50k, D=2)

<table>
<thead>
<tr>
<th>Method</th>
<th>50% (rel. error)</th>
<th>10% (rel. error)</th>
<th>1% (rel. error)</th>
<th>0% (rel. error)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive</td>
<td>301.7</td>
<td>301.7</td>
<td>301.7 sec.</td>
<td>301.7</td>
</tr>
<tr>
<td>FFT</td>
<td>3.1</td>
<td>&gt; 600</td>
<td>&gt; 600</td>
<td>-</td>
</tr>
<tr>
<td>IFGT</td>
<td>12.2</td>
<td>&gt; 600</td>
<td>&gt; 600</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Gaussian)</td>
<td>2.7 (3.1*)</td>
<td>3.4 (4.8*)</td>
<td>3.8 (5.5*)</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Epanechn.)</td>
<td>0.8 (0.8*)</td>
<td>0.8 (0.8*)</td>
<td>0.8 (0.8*)</td>
<td>6.5 [109.2]</td>
</tr>
</tbody>
</table>
## bio5 (N=100k, D=5)

<table>
<thead>
<tr>
<th></th>
<th>50% (rel. error)</th>
<th>10%</th>
<th>1%</th>
<th>0%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive</td>
<td>1966.3</td>
<td>1966.3</td>
<td>1966.3 sec.</td>
<td>1966.3</td>
</tr>
<tr>
<td>FFT</td>
<td>&gt; RAM</td>
<td>&gt; RAM</td>
<td>&gt; RAM</td>
<td>-</td>
</tr>
<tr>
<td>IFGT</td>
<td>&gt; 4000</td>
<td>&gt; 4000</td>
<td>&gt; 4000</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Gaussian)</td>
<td>72.2 (98.8*)</td>
<td>79.6 (111.8*)</td>
<td>87.5 (128.7*)</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Epanechn.)</td>
<td>27.0 (28.2*)</td>
<td>28.4 (28.4*)</td>
<td>28.4 (28.4*)</td>
<td>408.9 [1074.9]</td>
</tr>
<tr>
<td>Method</td>
<td>50% (rel. error)</td>
<td>10%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>-----</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Exhaustive</td>
<td>710.2</td>
<td>710.2</td>
<td>710.2 sec.</td>
<td>710.2</td>
</tr>
<tr>
<td>FFT</td>
<td>&gt; RAM</td>
<td>&gt; RAM</td>
<td>&gt; RAM</td>
<td>-</td>
</tr>
<tr>
<td>IFGT</td>
<td>&gt; 1400</td>
<td>&gt; 1400</td>
<td>&gt; 1400</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Gaussian)</td>
<td>155.9 (159.7*)</td>
<td>159.9 (163*)</td>
<td>162.2 (167.6*)</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Epanechn.)</td>
<td>10.0 (10.0*)</td>
<td>10.1 (10.1*)</td>
<td>10.1 (10.1*)</td>
<td>261.6 [558.7]</td>
</tr>
</tbody>
</table>
### covtype (N=150k, D=38)

<table>
<thead>
<tr>
<th>Method</th>
<th>50% (rel. error)</th>
<th>10%</th>
<th>1%</th>
<th>0%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhaustive</td>
<td>13157.1</td>
<td>13157.1</td>
<td>13157.1 sec.</td>
<td>13157.1</td>
</tr>
<tr>
<td>FFT</td>
<td>&gt; RAM</td>
<td>&gt; RAM</td>
<td>&gt; RAM</td>
<td>-</td>
</tr>
<tr>
<td>IFGT</td>
<td>&gt; 26000</td>
<td>&gt; 26000</td>
<td>&gt; 26000</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Gaussian)</td>
<td>139.9 (143.6*)</td>
<td>140.4 (145.7*)</td>
<td>142.7 (148.6*)</td>
<td>-</td>
</tr>
<tr>
<td>Dualtree (Epanechn.)</td>
<td>54.3 (54.3*)</td>
<td>56.3 (56.3*)</td>
<td>56.4 (56.4*)</td>
<td>1572.0 [11486.0]</td>
</tr>
</tbody>
</table>
### Speedup Results: Large dataset

<table>
<thead>
<tr>
<th>N</th>
<th>naïve</th>
<th>tree</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.5K</td>
<td>7</td>
<td>.12</td>
<td></td>
</tr>
<tr>
<td>25K</td>
<td>31</td>
<td>.31</td>
<td></td>
</tr>
<tr>
<td>50K</td>
<td>123</td>
<td>.46</td>
<td></td>
</tr>
<tr>
<td>100K</td>
<td>494</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>200K</td>
<td>1976*</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>400K</td>
<td>7904*</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>800K</td>
<td>31616*</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.6M</td>
<td>35 hrs</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

One order-of-magnitude speedup over single-tree at ~2M points

5500x

![Scaling behavior with number of data](chart.png)
Kernel density estimation wrapup

• Properties:
  – fastest practical algorithm for general $D$
  – all kernels, weighted, variable-kernel
  – hard bounds, automatic error control
  – simple, easy to program
  – conjecture: $O(N\log N)+O(N)$

• Insights: like FMM with adaptive geometry
  + automatic error control

• Has been used in practice [Balogh et al. 02, Miller et al. 03]

• See [Gray & Moore NIPS 00], [Gray & Moore 03]
1. warm-up: generalized histogram
2. n-point statistics
3. kernel density estimator
4. general strategy: multi-tree
   1. nonparametric Bayes classifier
   2. support vector machine
   3. nearest neighbor statistics
   4. Gaussian process regression
   5. Bayesian inference
5. science!
\[ q : \sum_i I_r(\delta_{qi}) \]
\[ q : \bigcup_i iI_r(\delta_{qi}) \]
\[ q : \arg\min_i \delta_{qi} \]
\[ \forall q : \arg\min_i \delta_{qi} \]
\[ \sum_i \sum_j \sum_k I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \]
\[ \forall q : \sum_i K_r(\delta_{qi}) \]
\[ \forall q : \sum_i w_i K_r(\delta_{qi}) \]
\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]
\[ q : \sum I_r(\delta_{qi}) \]  \hspace{1cm} \text{(radial) range count}

\[ q : \bigcup_i i I_r(\delta_{qi}) \]  \hspace{1cm} \text{(radial) range search}

\[ q : \arg \min_i \delta_{qi} \]  \hspace{1cm} \text{nearest-neighbor}

\[ \forall q : \arg \min_i \delta_{qi} \]

\[ \sum \sum \sum \sum I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \]

\[ \forall q : \sum_i K_r(\delta_{qi}) \]  \hspace{1cm} \text{different operators, same alg.}

\[ \forall q : \sum w_i K_r(\delta_{qi}) \]  \hspace{1cm} \text{fastest practical algorithms}

\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]
\[ \forall q : \arg \min_i \delta_{qi} \]

\[ \forall q : \arg \min_i \delta_{qi} \quad \text{all-nearest-neighbors} \]

\[ \sum_i \sum_j \sum_k I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \quad \text{common, e.g. in LLE} \]

\[ \forall q : \sum_i K_r(\delta_{qi}) \]

\[ \forall q : \sum_i w_i K_r(\delta_{qi}) \]

\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]
All-nearest-neighbors

<table>
<thead>
<tr>
<th>Dataset</th>
<th>D</th>
<th>N</th>
<th>Naive</th>
<th>Vaidya</th>
<th>WSPD</th>
<th>Single/ball-tree</th>
<th>Dual/ball-tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>sj2-50k-2</td>
<td>2</td>
<td>50000</td>
<td>171.56</td>
<td>3828.30</td>
<td>4594.07</td>
<td>1.060714</td>
<td>0.453877</td>
</tr>
<tr>
<td>colors50k</td>
<td>2</td>
<td>50000</td>
<td>171.72</td>
<td>5876.74</td>
<td>8977.11</td>
<td>1.819025</td>
<td>0.595279</td>
</tr>
<tr>
<td>bio5</td>
<td>5</td>
<td>103010</td>
<td>1180.0</td>
<td>28707.39</td>
<td>74342.05</td>
<td>4.182194</td>
<td>1.644841</td>
</tr>
<tr>
<td>corel</td>
<td>32</td>
<td>37749</td>
<td>465.75</td>
<td>–</td>
<td>–</td>
<td>107.820000</td>
<td>71.6488</td>
</tr>
<tr>
<td>covtype38d</td>
<td>38</td>
<td>150000</td>
<td>13057.41</td>
<td>–</td>
<td>–</td>
<td>32.152322</td>
<td>18.7831</td>
</tr>
<tr>
<td>covtype</td>
<td>55</td>
<td>581013</td>
<td>191552.03</td>
<td>–</td>
<td>–</td>
<td>402.865</td>
<td>268.1345</td>
</tr>
<tr>
<td>biotrain</td>
<td>75</td>
<td>285409</td>
<td>67127.7</td>
<td>–</td>
<td>–</td>
<td>2729.5</td>
<td>3969.271</td>
</tr>
<tr>
<td>phytrain</td>
<td>79</td>
<td>150000</td>
<td>20247.9</td>
<td>–</td>
<td>–</td>
<td>1415.81</td>
<td>218.190</td>
</tr>
<tr>
<td>mnist10k</td>
<td>784</td>
<td>10000</td>
<td>661.28</td>
<td>–</td>
<td>–</td>
<td>650.195400</td>
<td>656.9486</td>
</tr>
<tr>
<td>disk</td>
<td>1025</td>
<td>40000</td>
<td>13561.64</td>
<td>–</td>
<td>–</td>
<td>10392.435874</td>
<td>8307.970</td>
</tr>
<tr>
<td>galaxy</td>
<td>3840</td>
<td>40000</td>
<td>51069.59</td>
<td>–</td>
<td>–</td>
<td>9105.325601</td>
<td>11278.814</td>
</tr>
</tbody>
</table>

- natural generalization of nn alg.
- fastest practical algorithm
\[ q : \sum_i I_r(\delta_{qi}) \]
\[ q : \bigcup_i i I_r(\delta_{qi}) \]
\[ q : \arg \min_i \delta_{qi} \]

\[ \forall q : \arg \min_i \delta_{qi} \]

\[ \sum_i \sum_j \sum_k I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \]

\[ \forall q : \sum_i K_r(\delta_{qi}) \]

\[ \forall q : \sum_i w_i K_r(\delta_{qi}) \]

\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]

- extension to n-tuples
- *fastest practical algorithm*

n-point correlations
\[ q : \sum I_r(\delta_{qi}) \]
\[ q : \bigcup_i iI_r(\delta_{qi}) \]
\[ q : \arg \min_i \delta_{qi} \]
\[ \forall q : \arg \min_i \delta_{qi} \]
\[ \sum \sum \sum I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \]
\[ \forall q : \sum_i K_r(\delta_{qi}) \]
\[ \forall q : \sum_i w_i K_r(\delta_{qi}) \]
\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]
\[ q : \sum I_r(\delta_{qi}) \]
\[ q : \bigcup_i iI_r(\delta_{qi}) \]
\[ q : \arg \min_i \delta_{qi} \]
\[ \forall q : \arg \min_i \delta_{qi} \]
\[ \sum \sum \sum I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \]
\[ \forall q : \sum K_r(\delta_{qi}) \]
\[ \forall q : \sum_i w_i K_r(\delta_{qi}) \]
\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]

- arbitrary scalars
- fastest practical algorithm

\[ \text{Nadaraya-Watson regression} \]
Bayesian inference

\[ I = \int \frac{g(x)f(x)}{f(x)} \, dx \]

Adaptive importance sampling

\[ I = \int \frac{f(x)}{q(x)}q(x) \, dx \]

Sample from \( q() \)

Re-estimate \( q() \) from samples

\[ \int [f(x) - q(x)]^2 \, dx \]

\[ V(\hat{I}_q) = E[(\hat{I}_q - E[\hat{I}_q])^2] \]

\[ \min \int \frac{[f(x) - Iq(x)]^2}{q(x)} \, dx \]

New computational capabilities inspire new methods
\[ q : \sum I_r(\delta_{qi}) \]
\[ q : \bigcup_i iI_r(\delta_{qi}) \]
\[ q : \arg\min_i \delta_{qi} \]
\[ \forall q : \arg\min_i \delta_{qi} \]
\[ \sum \sum \sum I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \]
\[ \forall q : \sum K_r(\delta_{qi}) \]
\[ \forall q : \sum w_i K_r(\delta_{qi}) \]
\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]

\[ K^{-1}y \rightarrow Kw \]

- N x N matrix inverse \rightarrow kernel matrix-vector multiply
- problems sometimes hidden
- awaiting further testing

\[ \text{Gaussian process regression} \]
\[ q : \sum_i I_r(\delta_{qi}) \]
\[ q : \bigcup_i iI_r(\delta_{qi}) \]
\[ q : \arg \min_i \delta_{qi} \]
\[ \forall q : \arg \min_i \delta_{qi} \]
\[ \sum_i \sum_j \sum_k I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \]
\[ \forall q : \sum_i K_r(\delta_{qi}) \]
\[ \forall q : \sum_i w_i K_r(\delta_{qi}) \]
\[ \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \]

- decision problem \(\rightarrow\) exact alg. using priority queues
- fastest practical algorithm

\[ \text{nonparametric Bayes classifier} \]
\( q : \sum I_r(\delta_{qi}) \)

\( q : \bigcup_i iI_r(\delta_{qi}) \)

\( q : \arg \min_i \delta_{qi} \)

\( \forall q : \arg \min_i \delta_{qi} \)

\( \sum \sum \sum I_{rst}(\delta_{ij}, \delta_{jk}, \delta_{ki}) \)

\( \forall q : \sum K_r(\delta_{qi}) \)

\( \forall q : \sum w_i K_r(\delta_{qi}) \)

\( \forall q : \max \left\{ \sum_i K_r(\delta_{qi}), \sum_j K_r(\delta_{qj}) \right\} \quad \text{support vector machine} \)

\( \forall q : \text{sgn} \sum_i \alpha_i K(\delta_{qj}) \)

• only 2-3x speedup over naive

• failure for this problem
These were examples of...

**Generalized N-body problems**

[Gray thesis 2003]

**All-NN:** \( \{ \forall, \text{arg min}, \delta, \cdot \} \)

**2-point:** \( \{ \Sigma, \Sigma, I_r(\delta), w \} \)

**3-point:** \( \{ \Sigma, \Sigma, \Sigma, I_R(\delta), w \} \)

**KDE:** \( \{ \forall, \Sigma, K_r(\delta); \{ r \} \} \)

Gaussian process regression
nonparametric Bayes classif.
radial basis functions
particle filters
nonparam. belief propagation

mean shift
local poly. regression
Coulombic simulation
SPH fluid dynamics
kernel PCA
Isomap
projection pursuit
minimum spanning tree
k-means
Hausdorff distance
mixture of Gaussians

...
These were examples of Multi-tree methods
[Gray thesis 2003]

quite general
simple, recursive
error bounds
automatic error control

general dimension
data structure-agnostic
general tuple order
polychromatic
multiple kernels
subset-decomposable
operators
symmetric monotonic
kernel functions
metric space

Unifies/extends: FMM, Barnes-Hut, Appel’s algorithm, WSPD, nearest-neighbor search, spatial join, graphics collision detection
OUTLINE

1. warm-up: generalized histogram

2. n-point statistics

3. kernel density estimator

4. general strategy:
   1. nonparametric Bayes classifier
   2. support vector machine
   3. nearest neighbor statistics
   4. Gaussian process regression
   5. Bayesian inference

5. science!
Science: Map of the quasars, i.e. mass?

NBC on 500,000 training data, 800,000 test data

Largest quasar catalog to date, deepest mass map of universe.


Coming: 1,000,000 quasars
Science: Does the model fit the data?

Same?

3-point on 130,000 galaxies, 1.3M random

Ongoing: 3-point on VIRGO

Most comprehensive third-order statistics on universe to date.

Science: Does dark energy exist?

Do we see the ISW Effect?

2-point on 2,000,000 galaxies and WMAP pixels

Physical evidence of dark energy.

[Scranton et al., PRL 2005 submitted]
Bob Nichol on David Letterman show
July 2003
Science

#1 Breakthrough of 2003
Summary

• **Fastest practical algorithms:** n-point, KDE, all-NN, NBC, more coming…

• **Major science results:** directly due to faster algorithms; *much* more coming…

• **General principles:** generalized N-body problems → multi-tree methods
END
Machine learning in general

data
  +
model/task
  +
objective function
  ↓
learning algorithm
  ↓
scalable learning algorithm
  ↓
code
Future steps...

data + model/task +

objective function

learning algorithm

scalable learning algorithm

code

non-vector objects!
e.g. proteins, spatio-temporal, relations

learning deduction, action!
e.g. reinforcement learning, ILP

generalize maximum likelihood!

generalize EM!

new N-body & Monte Carlo methods!
Machine learning in general

- data
+ model/task
  + objective function
    ↓
    learning algorithm
      ↓
      scalable learning algorithm
        ↓
        code
AutoBayes (Prolog system)
[Buntine 95], [Gray, Fischer, Schumann, Buntine NIPS 02]

- data
  - +
  - model/task
  - +
  - objective function
  - ↓
  - learning algorithm
  - ↓
  - scalable learning algorithm
  - ↓
  - code

- face-data.txt
  - +
  - mixture of Gaussians / clustering
  - +
  - maximum likelihood
  - ↓
  - EM-mog { ....
  - }
  - ↓
  - nbody-EM-mog { ....
  - }
  - ↓
  - nbody_EM_mog.c
Future steps...

- data
- +
- model/task
- +
- objective function
- ↓
- learning algorithm
- ↓
- scalable learning algorithm
- ↓
- code

- non-vector objects!
  e.g. proteins, spatio-temporal, relations

- learning deduction, action!
  e.g. ILP, reinforcement learning

- generalize maximum likelihood!

- generalize EM!

- new N-body & Monte Carlo methods!

- deductive code optimization!
Summary

• Fastest practical algorithms: n-point, KDE, all-NN, NBC, more coming...

• Major science results: directly due to faster algorithms; NVO, parallel; much more coming...

• General principles: generalized N-body problems → multi-tree methods

• Next:
  – computational principles: formalize/extend framework; distribution-sensitive analysis
  – statistical principles: robust learning theory, active learning

• My dream: automated application of principles → automatic data analysis (AI) [Gray, Fischer, Schumann, Buntine 02]
### Speedup Results: Dimensionality

<table>
<thead>
<tr>
<th>N</th>
<th>Epan.</th>
<th>Gauss</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.5K</td>
<td>.12</td>
<td>.32</td>
</tr>
<tr>
<td>25K</td>
<td>.31</td>
<td>.70</td>
</tr>
<tr>
<td>50K</td>
<td>.46</td>
<td>1.1</td>
</tr>
<tr>
<td>100K</td>
<td>1.0</td>
<td>2</td>
</tr>
<tr>
<td>200K</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>400K</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>800K</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>1.6M</td>
<td>23</td>
<td>51</td>
</tr>
</tbody>
</table>

![Scaling behavior with number of dimensions](image)
Observation: there’s a pattern

[Gray and Moore 00]

- kernel density estimator
- n-point statistics
- nonparametric Bayes classifier
- support vector machine
- nearest neighbor statistics
- Gaussian process regression
- Bayesian inference

\[ \forall q, \sum K(\|q - x_j\|) \]
\[ \sum_{i}^{N} \sum_{j}^{N} \sum_{k}^{N} I(\delta_{ij} < r_1)I(\delta_{jk} < r_2)I(\delta_{ki} < r_3) \]
\[ \forall q, \max \left\{ \sum_{i}^{N} K(\|q - x_i\|), \sum_{j}^{N} K(\|q - x_j\|) \right\} \]
\[ \forall q, \arg \min_{j} \sum_{i}^{N} K(\|q - x_i\|, \sum_{j}^{N} K(\|q - x_j\|) \}
\[ K^{-1} x \]
\[ \int f(x) p(x) dx \]

generalized N-body problems \( \rightarrow \) multi-tree methods

\[ \min \sum A^{-1} \int \]
**Science:** Spiral/elliptical galaxies - WHY?

KDE on 100,000 galaxies

First large-scale evidence explaining elliptical galaxies.

[Balogh et al., MNRAS 2004]
Can we ‘see’ general relativity?

13.5M galaxies, 195,000 quasars

First observation of general relativity of this kind.

[Scranton, Nichol, Connolly, et al. in prep.]
Experiments

• Optimal bandwidth \( h^* \) found by LSCV
• Error relative to truth: \( \text{maxerr} = \max |\text{est} - \text{true}| / \text{true} \)
• Only require that 95% of points meet this tolerance
• Note that these are small datasets for manageability
• Tweak parameters
  – FFT tweak parameter \( M \): \( M = 16 \), double until error satisfied
  – IFGT tweak parameters \( K, r_y, p \): 1) \( r_y = 2.5, K = \sqrt{N} \) 2) \( K = 10 \sqrt{N}, r_y = 16 \) and doubled until error satisfied; hand-tune \( p \) for dataset: \{8,8,5,3,2\}
  – Dualtree tweak parameter \( \tau \): \( \tau = \text{maxerr} \), double until error satisfied
  – Dualtree auto: just give it \( \text{maxerr} \)
Observations

- FGT can’t use tree; FMM doesn’t apply here
- like FMM on adaptive trees (general D):
  - conjecture: $O(N\log N) + O(N)$
  - works for all density estimation kernels
  - case 3 error control
  - simple, easy to program
- we trade off continuous sophistication for discrete sophistication

let’s compare…
These were examples of...

**Generalized N-body problems**

**All-NN:** \( \{ \forall, \text{arg min}, \delta, \cdot \} \)

**2-point:** \( \{ \Sigma, \Sigma, I_r(\delta), w \} \)

**3-point:** \( \{ \Sigma, \Sigma, \Sigma, I_R(\delta), w \} \)

**KDE:** \( \{ \forall, \Sigma, K_r(\delta), ; \{ r \} \} \)

**SPH:** \( \{ \forall, \Sigma, K_r(\delta), w; t \} \)

**Multi-tree methods:**
General algorithmic framework and toolkit for such problems
Ball-trees

Our algorithms can use any of these data structures

- Auton ball-trees III [Omohundro 91], [Uhlmann 91], [Moore 99]
- Cover-trees [Alina B., Kakade, Langford 04]
- Crust-trees [Yianilos 95], [Gray, Lee, Rotella, Moore 2005]
Basic proximity problems

- nearest-neighbor search
  \[ \arg\min_j \|q - x_j\| \]

- (radial) range search
  \[ \bigcup_j x_j I(\|q - x_j\| < r) \]

- (radial) range count
  \[ \sum_j I(\|q - x_j\| < r) \]
Exclusion and inclusion, on multiple radii simultaneously.

\[ \min ||x - x_i|| < r_1 \Rightarrow \min ||x - x_i|| < r_2 \]

Use binary search to locate critical radius: \( O(\log B) \)
1. Nonparametric Bayes classifier

\[ P(C_1 \mid x_q) = \frac{P(C_1) \hat{f}(x_q \mid C_1)}{P(C_1) \hat{f}(x_q \mid C_1) + P(C_2) \hat{f}(x_q \mid C_2)} \]
1. Nonparametric Bayes classifier

kernel sum decision problem

\[ \Phi_1(q) = \sum_i K(\|q - x_i\|), \quad \Phi_2(q) = \sum_j K(\|q - x_j\|) \]

\[ \forall q, \max \{ \Phi_1(q), \Phi_2(q) \} \]
1. Nonparametric Bayes classifier

kernel sum decision problem

\[ \Phi_1(q) = \sum_i K(\|q - x_i\|), \quad \Phi_2(q) = \sum_j K(\|q - x_j\|) \]

\[ \forall q, \max\{\Phi_1(q), \Phi_2(q)\} \]

\[ \Phi_1^{hi}(q) \quad \Phi_2^{hi}(q) \]

Exact

\[ \Phi_1^{lo}(q) \quad \Phi_2^{lo}(q) \]