Accelerating Chemical Similarity Search with GPUs and Metric Embeddings

Imran Haque
Department of Computer Science
Stanford University

http://cs.stanford.edu/people/ihaque
http://folding.stanford.edu

Thesis Defense, 11 Apr 2011
E. coli protein ???
E. coli penicillin binding protein 5
What do these compounds do?

- inhibit penicillin binding proteins?
- kill bacteria?
- kill viruses?
What do these compounds do?

- inhibit penicillin binding proteins?
- kill bacteria?
- kill viruses?
bisphenol A
estrogen mimic

clavulanic acid
beta-lactamase inhibitor

levofloxacin
DNA gyrase inhibitor

methicillin
beta-lactam antibiotic

zidovudine
HIV RT inhibitor

penicillin G
beta-lactam antibiotic
Chemical Biology - Questions

• Given a protein, what is its function? To what compounds does it bind?

• Given a small molecule, with which proteins will it interact?
Chemical Biology - Methods

- Experimental assays: expensive, labor-intensive

- Physical simulation: expensive, slow, accurate?

- Is there an alternative to giant molecular dynamics simulations?
Chemical Databases

• A modern trend – giant public databases of chemical assay data
  – NCBI PubChem: 34,340 assays; 965,730 compounds
  – EBI ChEMBLdb: 8,054 targets; 600,625 compounds

• Companies releasing their internal databases

• Let’s learn from this data and make predictions – chemical informatics/data mining!
The Cheminformatics Gap

Computational analysis has not kept up with growth in chemical databases: the cheminformatics gap.
Challenges in time and space

• Many methods $\sim O(N^2)$; FLOPs are not enough:

<table>
<thead>
<tr>
<th>Problem size</th>
<th>CPU time</th>
<th>Storage needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 mols</td>
<td>1 ms</td>
<td>1 kB</td>
</tr>
<tr>
<td>10K mols</td>
<td>1 min</td>
<td>1 GB</td>
</tr>
<tr>
<td>100K mols</td>
<td>1 day</td>
<td>1 TB</td>
</tr>
<tr>
<td>10M mols</td>
<td>3 yr</td>
<td>1 PB</td>
</tr>
<tr>
<td>1B mols</td>
<td>30K yr</td>
<td>10K PB</td>
</tr>
</tbody>
</table>

• Computing on existing-scale datasets requires entire datacenters’ worth of storage.
3 Views of Chemical Similarity

- 2D substructure:
  - Alanine: \( \text{H}_2\text{N}-\text{CH}-\text{C}-\text{OH} \)
  - Methionine: \( \text{H}_2\text{N}-\text{CH}-\text{C}-\text{OH} \)

- 3D shape:
  - Alanine: ![Alanine 3D structure](image1)
  - Methionine: ![Methionine 3D structure](image2)

- 3D chemotype:
  - Alanine: ![Alanine 3D chemotype](image3)
  - Methionine: ![Methionine 3D chemotype](image4)
A Modest Proposal

• All pairwise similarities in PubChem3D (N = 17M) based on 3D shape/color and 2D LINGO similarity

• 3D: OpenEye ROCS: 150/sec/core = ~ 30K cpu-years
  2D: OpenEye LINGO: 1M/sec/core = 4.5 cpu-years
  – 1 PB per matrix!
What’s in a GPU?

NVIDIA GF100
(GeForce GTX 480)

AMD Cypress
(Radeon HD 5870)
Why GPUs?

- GPUs have excellent peak throughput and efficiency

- BUT
  - Hard to program
  - Require inherent data parallelism
  - Often require complete rewrite
  - Questionable reliability?
But...specialized hardware is bad!

YO DAWG I HEARD U LIKE MASSIVELY PARALLEL COMPUTING

SO I PUT A GPU INSIDE YOUR CPU SO YOU CAN COMPUTE WHILE YOU COMPUTE

quickmeme.com
CPUs and GPUs are converging

**AMD Llano**

**Intel Sandy Bridge**

Integrated GPUs!

GPU-style hardware is here to stay.
GPU-Accelerated 3D Similarity

- Shape overlay optimization: find compounds from a DB similar in shape to an active “query” molecule

- Complexity O(MN): double-loop over all atom pairs
- DB = ~10M mol.; CPU = 100/sec = ~2 days/query

- *Use GPU to exploit parallelism*, make a faster ROCS: PAPER Accelerates Parallel Evaluations of ROCS

http://simtk.org/home/paper

Haque IS and Pande VS. *GPU Computing Gems*, vol 1. (2011)
Each optimization is independent, and each SM (OpenCL work-group) executes independently, so run one DB molecule per GPU core.

Map atom pair calculations to vector lanes (OpenCL work-items) within cores, and iterate through atom-pair matrix.

Haque IS and Pande VS. *GPU Computing Gems*, vol 1. **(2011)**
PAPER or PLASTIC, sir?

- Added color (chemotype) matching to PAPER: PLASTIC aLigns Atoms with Shape Theory Incorporating Color

- GPU acceleration yields **100x speedup** (vs 1 core): 15K alignments/sec/GPU

Haque IS and Pande VS. *GPU Computing Gems*, vol 1. (2011)

http://simtk.org/home/paper
Introduction to LINGO

• “1-D” similarity method comparing canonical SMILES (depth-first traversal) of molecules by fragmentation

  Benzene -> c1ccccc1 -> [c1cc, ccc1, 1ccc, cccc, cccc]
  Pyridine -> n1ccccc1 -> [n1cc, ccc1, 1ccc, cccc, cccc]

\[
T_{A,B} = \frac{|A \cap B|}{|A \cup B|} \quad T_{A,B} = \sum_{i=1}^{\ell} \left( 1 - \frac{|N_{A,i} - N_{B,i}|}{N_{A,i} + N_{B,i}} \right)
\]

• Efficient implementation by Grant et al. (2006): build DFA from reference string (O(N)), run query strings through automaton to calculate Tanimoto (O(N))

http://simtk.org/home/siml

DFAs and GPUs

- DFA-based algorithm is poorly suited for GPUs:
DFAs and GPUs

- DFA-based algorithms are poorly suited for GPUs:

Coleman, *Introducing Speech and Language Processing*


...
DFAs and GPUs

• DFA-based algorithm is poorly suited for GPUs:
  
  - Efficient branching (*warp divergence*)
  - Large low-latency jump tables (*limited local memory*)

*Coleman, Introducing Speech and Language Processing*

LINGO: Kernelization

- LINGO intersection can be interpreted as a Mercer kernel (inner product) in an infinite-dimensional binary space

- Example: alphabet={A,B}, substring length=2 (AA, AB, BA, BB)
  S1 = AABABBBAAB  
  S2 = BBABAAABBA

- Sum counts for all repeats of a substring -> sparse finite dimension (2^32 for LINGO)
  - Replace * with \textit{min} -> query-vs-database search now looks like a \textit{sparse matrix-sparse vector product}
SIML: A New LINGO algorithm

- Pack substrings into 32-bit ints (sparse vector index)
- Molecules → sorted lists of integers (and counts)

\[ T_{AB} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \]

- Calculate intersection by algorithm like merging sorted lists: simple control logic, cache-friendly
- Run one similarity per vector lane since each is relatively cheap: Single-Instruction, Multiple-LINGO

http://simtk.org/home/siml

Haque IS and Pande VS. *GPU Computing Gems*, vol 1. (2011)
SIML: Memory Optimization

Row-major layout is fine for the (non-vectorized) CPU because we can rely on cache to bring in partial rows for each core...

... but *kills* GPU performance

http://simtk.org/home/siml

Haque IS and Pande VS. *GPU Computing Gems*, vol 1. (2011)
## SIML: Memory Optimization

Transposing molecule layout to column-major maximizes spatial locality among threads. Can barrier on each row to guarantee coalescing, or use a 2D texture (if available on hardware) for more speed.

<table>
<thead>
<tr>
<th>Mol 1</th>
<th>Mol 2</th>
<th>Mol 3</th>
<th>Mol 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>@H]()</td>
<td>]{C(</td>
<td>@H]()</td>
<td>(=O)</td>
</tr>
<tr>
<td>C(C)</td>
<td>@H]()</td>
<td>@H]()</td>
<td>(=O)</td>
</tr>
<tr>
<td>C=C</td>
<td>ccc()</td>
<td>ccc()</td>
<td>(O)c0</td>
</tr>
<tr>
<td>NH0+</td>
<td>cc0)</td>
<td>cc0)</td>
<td>ccn0</td>
</tr>
<tr>
<td>[NH0</td>
<td>(=O)</td>
<td>(=O)</td>
<td>CC(=</td>
</tr>
<tr>
<td>COc0</td>
<td>O-)]</td>
<td>](c0</td>
<td>NC(=</td>
</tr>
<tr>
<td>ccc0</td>
<td>)[O-]</td>
<td>(cc0</td>
<td>)Nnc</td>
</tr>
<tr>
<td>0CC=</td>
<td>(cc0</td>
<td>ccc0</td>
<td>=O)N</td>
</tr>
</tbody>
</table>

Haque IS and Pande VS. *GPU Computing Gems*, vol 1. (2011)
SIML: Performance

23x vs DFA (best CPU method)

3x speedup with new algorithm on CPU vs OpenEye

82x speedup on GPU vs OpenEye code

http://simtk.org/home/siml

A Modest Proposal

- All pairwise similarities in PubChem3D (N = 17M) based on 3D shape/color and 2D LINGO similarity

- 3D: PAPER: 15K/sec/gpu = ~ 300 gpu-years
  2D: SIML: 91M/sec/gpu = ~ 4 gpu-weeks
  - GPU 2D is faster than reading the solution from disk!

(Time passes differently in NVIDIA, so by putting the CPU and storage for my machine there, I was able to run through the Folding@Home and SETI@Home databases in about an hour. There are so many problems with that.)
Cinematic Interlude: What’s a Tanimoto, anyway?

Do you know what a Tanimoto is?

Yeah, it’s that guy whose mind you tried to hack, right?

... That’s “Saito”. Racist.

Lesson 1: Don’t ask the business guy.
Cinematic Interlude: What’s a Tanimoto, anyway?

Yusuf, you’re a chemist. Do you know what a Tanimoto is?

Sure, it’s a number between 0 and 1, where 1 is perfect similarity.

Yes, but ...

We need to go deeper.

TANIMOTO
On the Nature of the Tanimoto

• Many molecular similarity methods report similarity as a Tanimoto score
• How can we use the mathematical structure of Tanimotos to gain insight into the metrics and calculate them faster?

Classical vector Tanimoto returns value in \([0, 1]\) for a pair of vectors \(A, B \in \mathbb{R}^{+N}\) in terms of their inner products

\[
T_{AB} = \frac{\langle A, B \rangle}{\langle A, A \rangle + \langle B, B \rangle - \langle A, B \rangle}
\]

Tanimoto equation can be rearranged to get inner product in terms of Tanimoto and vector magnitudes

\[
\langle A, B \rangle = \frac{T_{AB}}{1 + T_{AB}} (\langle A, A \rangle + \langle B, B \rangle)
\]

Do You Know the Way to Vectorland?

- Assume molecules can be represented as vectors in $\mathbb{R}^N$
- Simple assumptions on $\langle A, A \rangle$ and $\langle B, B \rangle$ get us $\langle A, B \rangle$

$$\langle A, B \rangle = \frac{2T_{AB}}{1 + T_{AB}}$$

- Given a matrix $G$ of inner products, want matrix $M$ with molecule vectors along rows

$$MM^T = G$$

- $G$ is real-symmetric, so use eigenvalue decomposition

$$G = MM^T = VDV^T$$

$$M = VD^{\frac{1}{2}}$$

SCISSORS: The key

SCISSORS Computes Interpolated Shape Signatures Over ROCS Space

• Select $k << N$ molecules to act as a “basis set”
• Do all-pairs comparison on basis set and decompose to molecule matrix $M$
• For each new “library” molecule $x$, run slow method only against basis set. Place inner products in a vector and solve for vector rep of $x$ by least-squares:

$$M \vec{x} = T$$

• All-pairs: now only $O(kN)$ slow computations!

**SCISSORS All-Pairs Dataflow**

- **Database**
  - N molecule database, K mol basis, d-dimensional vectors
  - **Similarity evaluations** (~1e3/s):
    - Basis all-pairs: $K(K-1)/2$
    - Library-vs-basis: $K(N-K)$
  - **Linear algebra** (~1e10/s):
    - Eigendecomposition: $O(K^3)$
    - Least-squares: $O(NKd)$
    - Vector all-pairs: $O(N^2d)$

- **Split into basis+library**
  - Basis all-pairs sim
  - Eigenvectors
  - Library-vs-basis sim
  - Least-squares
  - Vector all-pairs
SCISSORS is Kernel Approximation

- SCISSORS treats similarity measures as *kernel functions* mapping molecules to some high-dim. feature space in which an inner product is performed

- Embeds molecules into the maximum-variance PSD subspace of the kernel feature space
  - Uncentered *kernel principal components analysis*

- Computes a low-rank approx. to the kernel operator by sampling kernel values at a few *landmark* points.
  - *Rank-k Nyström approximation*

Haque IS and Pande VS. *In preparation*
Schölkopf, Smola, Müller. ICANN 1997.
Williams CKI, Seeger M. NIPS 2000
SCISSORS: Theoretical Bounds

- Interpreting SCISSORS as a kernel method allows us to use previous results to probabilistically bound the error induced by sampling a small basis set.

- From kernel PCA: expected kernel error over pairs of independently chosen molecules is bounded.

- From Nyström: error 2-norm, F-norm, and RMS error over an approximated matrix are all bounded.

**SCISSORS: Bounded RMS Error**

**Theorem 6.5** (Bounded RMS error)

With probability at least $1 - \delta$, the elementwise root-mean-square (RMS) error in the SCISSORS kernel matrix is worse than the lowest possible RMS error from a rank $k$-approximated kernel matrix by only a bounded amount:

$$\text{RMS}\{\|K - \hat{K}\|\} \leq \text{RMS}\{\|K - K_k\|\}$$

$$+ \left[ \frac{64k}{\ell} \right]^{1/4} R^2 \left[ 1 + 2 \sqrt{\frac{(n - \ell)^2}{(n - 1/2)(n - \ell - 1/2)} \log \frac{1}{\delta}} \right]^{1/2}$$

- Theoretical result turns out to be fairly loose: for data in original SCISSORS paper, error bound $\sim 3K_{\text{max}}$

- Suggests that chemical data are well-structured, *not* worst-case

Haque IS and Pande VS. *In preparation*
Limitations on Theory

• All bounds assume a PSD kernel
  – Easy to estimate RMS error from negative eigenvalues

• Bounds are taken after the Tanimoto->IP step
  – Distortion is related to structure of feature-space distribution

• Bounds assume exact kernel computations
High-D SCISSORS over LINGO

RMS Kernel Error of LINGO with no added noise; mean kernel value=7.86
High-D SCISSORS over Shape

RMS Kernel Error of shape overlap; mean kernel value=658.39

ERROR INCREASES?

CAN'T EXPLAIN THAT
High-D SCISSORS over noisy LINGO

RMS Kernel Error of LINGO with added $N(0,1.00^2)$ noise; mean kernel value=7.86
Shape overlay is noisy

- **Numerical local opt:** estimated Tanimoto (given true objective) is only a lower bound on true Tanimoto

- **Truncated objective:** ignores higher-order overlaps; estimates an upper bound on true Tanimoto at global optimum.

Histogram of true shape overlap (by quadrature) computed using truncated objective and local optimizer with 4 or 12 starting points; 12 are a strict superset of the 4.

Haque IS, Pande VS. *In preparation*
Modeling noise effects on SCISSORS

• Model: iid zero-mean Gaussian noise added to each kernel value:

\[ K'(x,y) = K(x,y) + N(0,\sigma^2) \]

  – Normal distribution is not critical here; for large enough kernel matrices, final effect will be normal thanks to CLT

• Errors from approximate kernel calculations are likely neither nondeterministic nor independent, but this makes a tractable and explanatory model
First-Order Eigenvalue Perturbation

• Eigenvalue perturbation theory (from QM): compute first-order approximations to the eigenvalues and eigenvectors of a slightly perturbed matrix

Given solutions $x_{0i}$ and $\lambda_{0i}$, $i = 1 \cdots N$ to the problem:

$$[K_0] x_{0i} = \lambda_{0i} [M_0] x_{0i}$$

We perturb the system as follows, where each $\delta$ term is assumed to be much smaller than its corresponding starting element:

$$[K] = [K_0] + [\delta K]$$
$$[M] = [M_0] + [\delta M]$$

To first order, the eigenvalues/eigenvectors $x_i$, $\lambda_i$ of the perturbed system are:

$$\lambda_i = \lambda_{0i} + x_{0i}^T ([\delta K] - \lambda_{0i} [\delta M]) x_{0i}$$
$$x_i = x_{0i} \left( 1 - \frac{1}{2} x_{0i}^T [\delta M] x_{0i} \right) + \sum_{j \neq i}^N \frac{x_{0j}^T ([\delta K] - \lambda_{0i} [\delta M]) x_{0i}}{\lambda_{0i} - \lambda_{0j}} x_{0j}$$

Haque IS, Pande VS. In preparation
Noise impact on SCISSORS: Eigenvalues

- We can reliably estimate large eigenvalues (those with value >> $\sigma$)

- Eigenvalues grow in magnitude with number of molecules in basis -> can estimate more dimensions with a larger basis

$$\lambda_i = \lambda_{0i} + N \left( 0, \sigma^2 \right)$$
Noise impact on SCISSORS: Eigenvectors

- We can reliably estimate eigenvectors whose corresponding eigenvalues have wide separation from other eigenvalues.

- Closely spaced eigenvalues should produce essentially random eigenvectors.

- *Spectral gap criterion* finds error minimum for shape.

\[ x_i = x_{0i} + \sum_{j \neq i}^N N \left( 0, \frac{\sigma^2}{(\lambda_{0i} - \lambda_{0j})^2} \right) x_j \]

Haque IS, Pande VS. *In preparation*
Noise impact on SCISSORS: Eigenvectors

- We can reliably estimate eigenvectors whose corresponding eigenvalues have wide separation from other eigenvalues.
- Closely spaced eigenvalues should produce essentially random eigenvectors.
- Spectral gap criterion finds error minimum for shape.

Haque IS, Pande VS. In preparation
SCISSORS: Shape Performance

On $>10^9$ molecule pairs from PubChem3D (159 x 4k * 4k):
shape RMS error < 0.05 at 256D, 2612 basis molecules;

**SCISSORS is within inherent noise of shape overlay**

Haque IS, Lucent D, Pande VS. *In preparation*
SCIORS: Color Performance

- Using SCISSORS discovered a defect in the color Tanimotos calculated by OE tools.

Haque IS, Lucent D, Pande VS. In preparation
SCISSORS: Combo Performance

- Clean metric properties from PLASTIC versus ROCS allow excellent combo-Tanimoto approximation

Haque IS, Lucent D, Pande VS. In preparation
Hardly Even a Request...

- 3D: Using PAPER+SCISSORS (basis size=2700)
  \[
  17M \times 2700 / 15000 = 35.4 \text{ gpu-day} + \\
  17M \times 17M / 600M = 5.6 \text{ gpu-day}
  \]
  \[274,000x \text{ speedup} \text{ (vs 30 000 cpu-yr)}\]

- 2D: Using SIML
  \[
  17M \times 17M / 91M = 36 \text{ gpu-day}
  \]
  \[40x \text{ speedup} \text{ (vs 4.5 cpu-yr)}\]

- Storage: fast methods cut space from $O(N^2)$ to $O(N)$:
  - 200M for SIML, 17GB for SCISSORS
  - 33,000 x reduction (3D); 2.8M x reduction (2D)
Why is it important that we have trapped antimatter?  Oh! It has future applications in propulsion, energy creation, data transmission, you name it!
Escherichia coli cluster

SCIENCE ADVOCATES

WHY IS IT IMPORTANT THAT WE HAVE TRAPPED ANTIMATTER?

OH! IT HAS FUTURE APPLICATIONS IN PROPULSION, ENERGY CREATION, DATA TRANSMISSION, YOU NAME IT!

SCIENTISTS

WHY IS IT IMPORTANT THAT WE HAVE TRAPPED ANTIMATTER?

BECAUSE IT'S FUCKING AWESOME.

Acknowledgments

• Graduate funding from NSF graduate fellowship and NIH/NIGMS training grant

• Funding for MOAR CORES provided by:
  – **Bio-X2 cluster**: NSF award CNS-0619926 for computer resources
  – **SimTK GPU cluster**: NSF grant for cyberinfrastructure CHE-0535616 and Roadmap GM072970
  – **NCSA Lincoln GPU cluster**: TeraGrid TG-MCB{090191,090183}
  – **Certainty cluster**: NSF award 0960306 MRI-R2: Acquisition of a Hybrid CPU/GPU and Visualization Cluster for Multidisciplinary Studies in Transport Physics with Uncertainty Quantification; this award is funded under the American Recovery and Reinvestment Act of 2009 (Public Law 111-5).

  *(lawyers like words)*
## Acknowledgments

### Stanford
- Vijay Pande (PI)
- Paul Novick
- Greg Bowman
- Kyle Beauchamp
- Jason Wagoner
- Vincent Yoelž
- Sergio Bacallado
- Rest of the Pande Lab
- Nick Tatonetti
- Mark Friedrichs
- Peter Eastman

### Collaborators
- John Chodera
- Del Lucent
- Pat Walters
- Kim Branson
- Henry Lin
- Bet Gregori
- Erik Lindahl
- Brian Cole
- Michael Houston
- Folding@home users
Conclusions

• Large-scale biochemical machine learning faces both computation and storage barriers

• Algorithmic transformation reveals coarse- and fine-grained parallelism enabling GPUs to handle chemical similarity

• The SCISSORS low-rank approximation recovers similarities down to their inherent noise floors, with >1000x less work

• First-order perturbation theory explains the performance of rank-reduction methods on a new domain of noisy or approximate kernels

• A problem that would have taken 2 months on all of Folding@home can now be done in a week on one machine!
Conclusions

- Large-scale biochemical machine learning faces both computation and storage barriers.
- Algorithmic transformation reveals coarse- and fine-grained parallelism enabling GPUs to handle chemical similarity.
- The SCISSORS low-rank approximation recovers similarities down to their inherent noise floors, with >1000x less work.
- First-order perturbation theory explains the performance of rank-reduction methods on a new domain of noisy or approximate kernels.
- A problem that would have taken 2 months on all of Folding@home can now be done in a week on one machine!
Conclusions

• Large-scale biochemical machine learning faces both computation and storage barriers.

• Algorithmic transformation reveals coarse- and fine-grained parallelism enabling GPUs to handle chemical similarity.

• The SCISSORS low-rank approximation recovers similarities down to their inherent noise floors, with >1000x less work.

• First-order perturbation theory explains the performance of rank-reduction methods on a new domain of noisy or approximate kernels.

• A problem that would have taken 2 months on all of Folding@home can now be done in a week on one machine!
Accelerating Chemical Similarity Search with GPUs and Metric Embeddings

Extra Slides

Imran Haque
Department of Computer Science
Stanford University

http://cs.stanford.edu/people/ihaque
http://folding.stanford.edu

Folding@home distributed computing

Thesis Defense, 11 Apr 2011
Theoretical Bounds from Kernel PCA

• Reduction allows the use of convergence results on kernel PCA.

• KPCA proj. error is bounded (Thm 1, Shawe-Taylor):

Define: $\hat{V}_k$ is the space spanned by the first $k$ eigenvectors of the sample correlation matrix. $\hat{V}_k^T$ is the orthogonal complement to space $\hat{V}_k$. Denote by $\lambda_k$ the kth process eigenvalue ("true" eigenvalues over the entire distribution generating our sample points) and $\hat{\lambda}_k$ the kth empirical eigenvalue (of the kernel matrix). Denote by $\lambda^{>k}$ the sum $\sum_{i>k} \lambda_k$, and similarly for $<k$ and for $\hat{\lambda}$.

If we perform PCA in the feature space defined by $\kappa$, then over random samples of points $S$ st $|S| = \ell$ ($\ell$-samples), for all $1 \leq k \leq \ell$, if we project new data onto the space $\hat{V}_k$, the expected squared residual is bounded by the following, with probability greater than $1 - \delta$:

$$\lambda^{>k} \leq \mathbb{E} \left[ \left| P_{\hat{V}_k}^T (\Phi(x)) \right|^2 \right]$$

$$\leq \min_{1 \leq d \leq k} \left[ \frac{1}{\ell} \hat{\lambda}^{>d} (S) + \frac{1 + \sqrt{d}}{\sqrt{\ell}} \sqrt{\frac{2}{\ell} \sum_{i=1}^{\ell} \kappa (x_i, x_i)^2} \right] + R^2 \sqrt{\frac{18 \ln \left( \frac{2\ell}{\delta} \right)}{\ell}}$$

Where the support of the distribution is in a ball of radius $R$ in feature space.

Theoretical Bounds from Kernel PCA

- Have extended Shawe-Taylor bound to one on the IP error:

Given a kernel $\kappa$ and a distribution of vectors $D$ such that $\kappa(x, x) \leq R^2 \ \forall x \in D$. Infer a SCISSORS basis on $\kappa$ on a random sample $S$ of $\ell$ points from $D$. Draw two vectors $x, y$ from $D$ and compute their projections $x_\parallel, y_\parallel$ onto the SCISSORS eigenspace. With probability $\geq (1-\delta)^2$ over samples $S$, for any $x, y$ chosen independently from $D$, the expected error in inner products is bounded:

$$0 \leq E[\kappa(x, y) - x_\parallel \cdot y_\parallel] \leq \min_{1 \leq d \leq \ell} \left[ \frac{1}{\ell} \hat{\lambda}^d(S) + \frac{1 + \sqrt{d}}{\sqrt{\ell}} \sqrt{\frac{2}{\ell} \sum_{i=1}^{\ell} \kappa(x_i, x_i)^2} + R^2 \left( \frac{1}{4} + \sqrt{\frac{18}{\ell} \ln \left( \frac{2\ell}{\delta} \right)} \right) \right]$$

Where $\hat{\lambda}^d(S)$ is the sum of all but the largest $d$ eigenvalues of the kernel matrix on $S$. 

Theoretical Bounds from Nyström

- Added error from sampling, w.r.t. error from rank-k approximation, is bounded in matrix 2-norm (from Theorem 5.2, Talwalkar):

Given a kernel $\kappa$ and a finite set of input vectors drawn from some probability distribution such that the feature space representation of the vectors is contained within a ball of radius $R$ around the origin. Let the true kernel matrix be denoted $K$ and the best possible rank-$k$ approximation to $K$ be denoted $K_k$. Compute a SCISSORS-approximated kernel matrix $\tilde{K}$ based on a size $\ell$ uniform sample of these vectors and a $k$-dimensional vector expansion. By substitution in Theorem 5.2 from Talwalkar, with probability $1 - \delta$, the 2-norm error in the SCISSORS kernel matrix is worse than the lowest possible error from a rank $k$-approximated kernel matrix by a bounded amount:

$$\|K - \tilde{K}\|_2 \leq \|K - K_k\|_2 + \frac{2n}{\sqrt{\ell}} K_{\max} \left[ 1 + \sqrt{\frac{n - \ell}{n - 1/2} \frac{1}{\beta(\ell, n)} \frac{1}{\delta} \log \frac{1}{\delta} q_{\text{max}}^{1/2} / K_{\max}^{1/2}} \right]$$

$$\leq \|K - K_k\|_2 + \frac{2n}{\sqrt{\ell}} R^2 \left[ 1 + 2 \sqrt{\frac{(n - \ell)^2}{(n - 1/2)(n - \ell - 1/2)} \frac{1}{\delta}} \right]$$

(1)

Theoretical Bounds from Nyström

- Added error from sampling, w.r.t. error from rank-\(k\) approximation, is bounded in Frobenius norm (from Theorem 5.2, Talwalkar):

\[
\|K - \tilde{K}\|_F \leq \|K - K_k\|_F + \left[ \frac{64k}{\ell} \right]^{1/4} nK_{\text{max}} \left[ 1 + \sqrt{\frac{n - \ell}{n - 1/2} \beta(\ell, n) \log \frac{1}{\delta} \frac{q^k}{K_{\text{max}}^{1/2}} \right]^{1/2}
\]

\[
\leq \|K - K_k\|_F + \left[ \frac{64k}{\ell} \right]^{1/4} nR^2 \left[ 1 + 2 \sqrt{\frac{(n - \ell)^2}{(n - 1/2)(n - \ell - 1/2)} \log \frac{1}{\delta} \right]^{1/2}
\]

(1)

MemtestG80: Motivation

- GPUs originate in *error-insensitive* consumer graphics
- Neither ECC nor parity on most graphics memory

- How suitable is the installed base of consumer GPUs (and consumer GPU-derived professional hardware!) for *error-sensitive* general purpose computing?

http://simtk.org/home/memtest

MemtestG80 – Folding@home

- Expect a low error rate and environment sensitivity, so must sample *many* cards in diverse environments

- Ran for ~7 months over 50,000+ NVIDIA GPUs on Folding@home (>840 TB-hr of testing)

- >97% of data tested 64 MiB RAM, k=512 logic LCG

---

http://simtk.org/home/memtest

GT200 has typical $P_f = 2.2 \times 10^{-6}$
(one-tenth of G80!)

*Both archs. show monotonic decline in zero-error populations.*

Tesla traces are rougher from poorer sampling, but appear to represent same error distribution as GeForce.

http://simtk.org/home/memtest

What about Fermi/AMD/CPU?

- NVIDIA’s Fermi (GF100) architecture adds SECDED ECC (disabled in consumer GeForce line), GDDR5 memory bus ECC, and L1/L2 caches

- FAH test does not run (yet) on Fermi/AMD; used standalone MemtestG80/MemtestCL
  - In-house:
    - GF100: GeForce GTX 480; Tesla C2050
    - RV770/Evergreen: Radeon 4870 (RV770); Radeon 5870 (Cypress)
  - Public:
    - GF100: 44 GTX 470, 43 GTX 480
    - RV700: 2 RV710, 15 RV730, 88 RV770
    - Evergreen: 1 Cedar, 6 Redwood, 50 Juniper, 103 Cypress
    - CPUs: 16 Core i7, 11 Core 2, 17 Phenom/Athlon II
Results – Fermi/AMD/CPU

- **CPU**: no errors seen

- **Tesla**: no app-level errors seen, at least one double-bit error reported by ECC

- **GeForce**: most cards exhibited memory errors on walking 8-bit zeros – observed in-house $P_f = 1.6 \times 10^{-5}$

- **RV770**: typ. fail random blocks/mod-20 – around $P_f = 7 \times 10^{-4}$
  - **Cypress**: almost all fail random blocks around $P_f = 4 \times 10^{-4}$
    - Tend to fail close to a kilobit simultaneously; unofficial word from AMD: random patterns might interfere with internal memory logic
SCISSORS: Shape Performance

On $>10^9$ molecule pairs from PubChem3D (159 x 4k * 4k):
shape RMS error < 0.05 at 256D, 2612 basis molecules;

*SCISSORS is within inherent noise of shape overlay*

Haque IS, Lucent D, Pande VS. *In preparation*
SCISSORS: Color Performance

- Using SCISSORS discovered a defect in the color Tanimotos calculated by OE tools

Haque IS, Lucent D, Pande VS. In preparation
SCISSORS: Combo Performance

- Clean metric properties from PLASTIC versus ROCS allow excellent combo-Tanimoto approximation

Haque IS, Lucent D, Pande VS. In preparation