Improved Linear Algebra Methods for Redshift Computation from Limited Spectrum Data

Miranda Braselton, Kelley Cartwright, Michael Hurley, Maheen Khan, Miguel Angel Rodriguez, David Shao, Jason Smith, Jimmy Ying, Genti Zaimi

December 8, 2006

Drs. Way and Srivastava (2006) compared different statistical models to approximate the redshifts of galaxies.

Computational limitations prevented them from using one particular model, known as Gaussian process regression, for large data sets.

We will present several efficient methods that allow us to overcome these limitations.

Outline

I. The Problem:
   A. Photometric Redshift - Miranda Braselton
   B. Gaussian Process - Jason Smith
   C. Gaussian Elimination - Miguel Angel Rodriguez
   D. Polynomial Kernel - Kelley Cartwright

II. The Solution:
   A. Reduced Rank - Genti Zaimi
   B. Cholesky Update - Maheen Khan
   C. Conjugate Gradient - Jimmy Ying
   D. Gibbs Sampler - Michael Hurley
   E. Testing Results - David Shao

What is a redshift?

- A redshift is the change in wavelength divided by the initial wavelength
- Indicates that an object is moving away from you

The Doppler Effect:

the light waves emitted

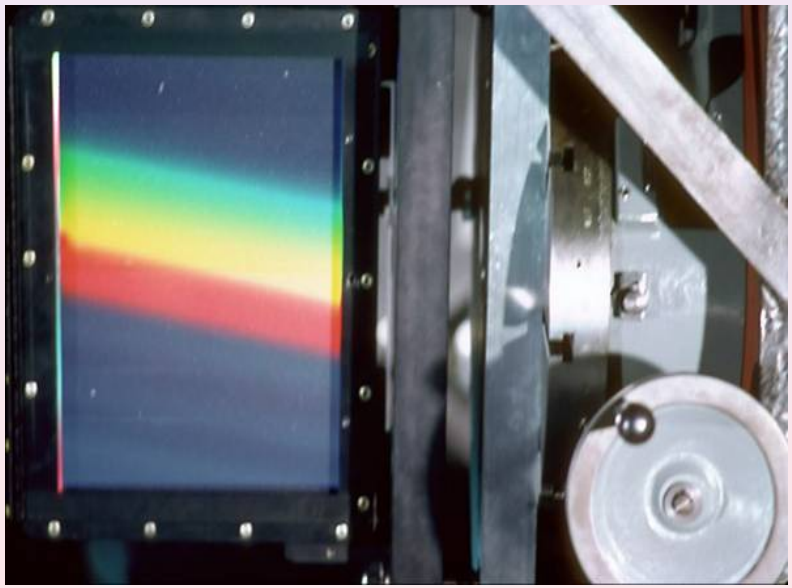
by the moving object shift

Why are they important?

- Scientists can determine many characteristics of galaxies and our universe.
- This information is useful for understanding the structure of the universe.

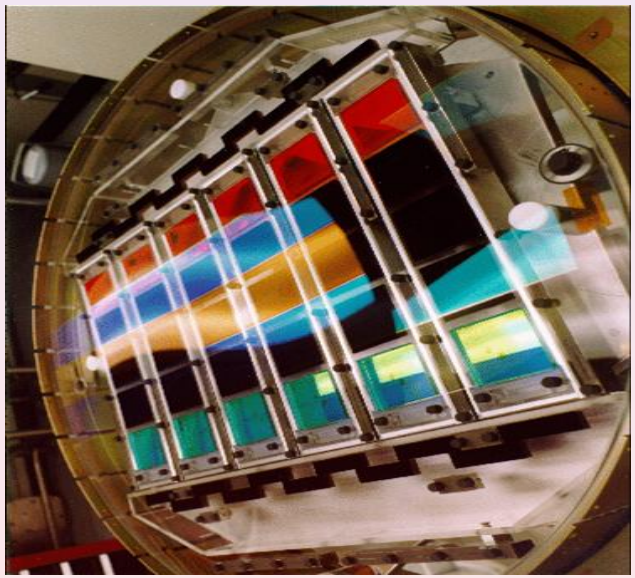
Spectroscopy vs. Photometry

- Uses diffraction grating to collect data from few objects



Diffraction grating

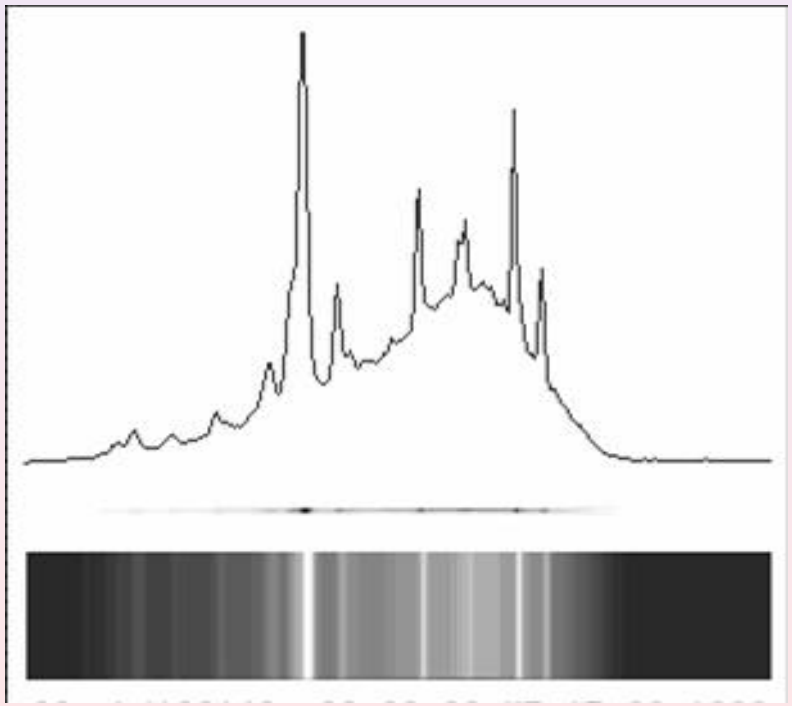
- Uses filters to collect data from multiple objects



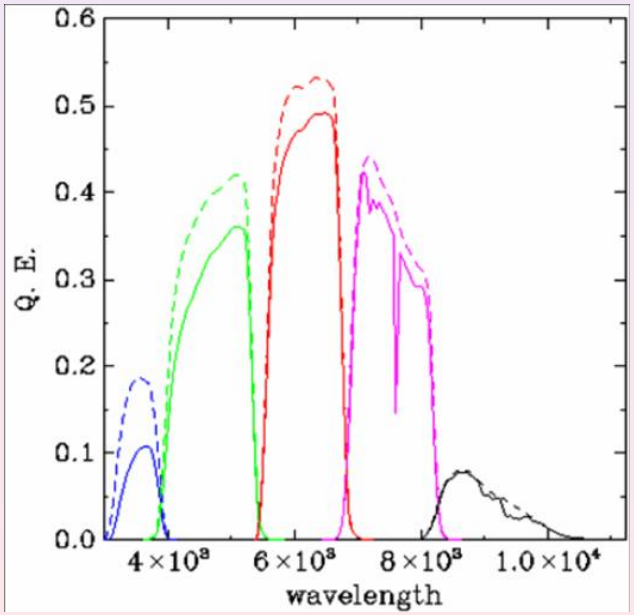
Photometric Filters

Spectroscopy vs. Photometry

- More accurate

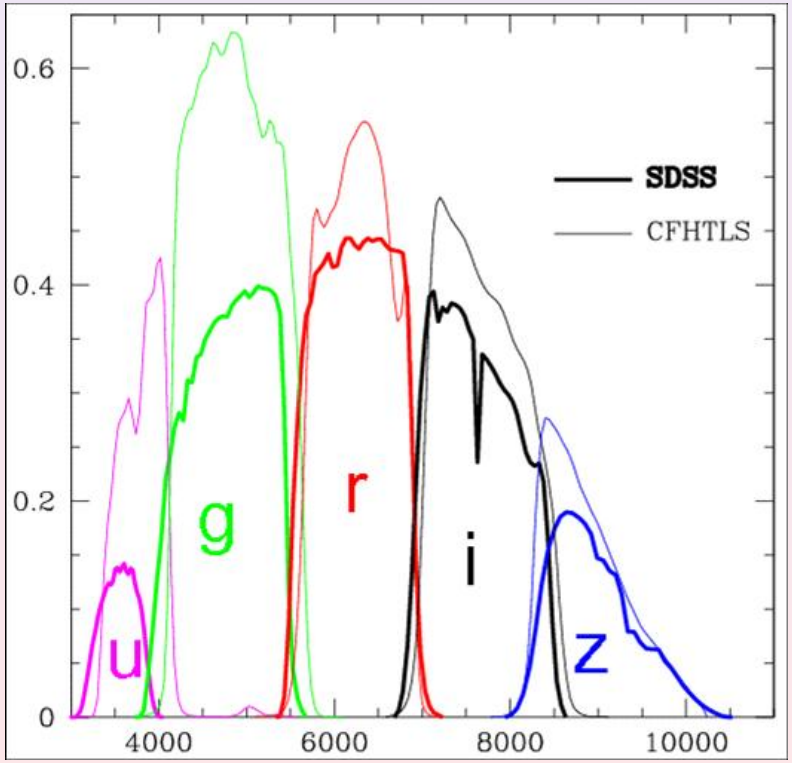


- Faster and cheaper



Photometric Data

Photometric data collected from a galaxy: area under each filter curve gives us 5 numbers (u, g, r, i, z) that are used in our calculations.



Training Set Method

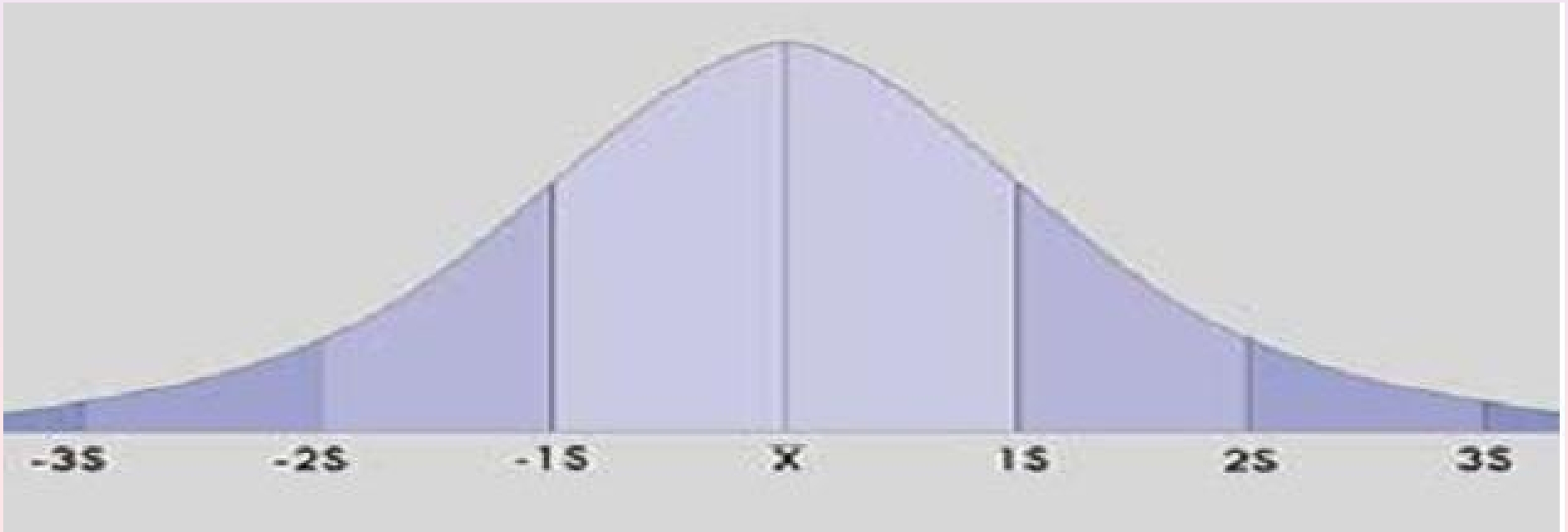
- Scientists need to find a better mathematical model for the photometric data that will estimate the redshift.
- Linear and quadratic regression, Artificial Neural Network Approach (ANNz), the ensemble model, ***Gaussian process***.

Redshift Calculations

- Goal: Predict the redshift of a galaxy from a photometric measurement (u, g, r, i, z) based upon the known redshifts of other galaxies.
- The statistical model developed for this project is based on Gaussian process.

Gaussian Process

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian (i.e. **bell shaped**) distribution.



The Model

Predicted photometric redshift is given by

$$\hat{\mathbf{y}} = \kappa^{\mathbf{T}}(\lambda^2 I + K)^{-1}\mathbf{y}$$

- K and y come from a training set.
- κ comes from a new photometric measurement.

OUR JOB

Major calculation is reduced to solving the following system:

$$(\lambda^2 I + K)\mathbf{w} = \mathbf{y}$$

up to 180,000 equations and 180,000 variables.

GAUSSIAN ELIMINATION

	general n	$n = 1000$	$n \approx 180,000$
Operations count	$\frac{2}{3}n^3$	7×10^8	4×10^{15}
Storage	n^2	10^6	3×10^{10}

The Bad News

Out of memory!

at $n \approx 11,000$

THE CHALLENGE

Find efficient ways to solve linear systems for ***large*** n.

The covariance matrix, K

- K models the interdependencies between the data.
- Its properties depend on a kernel function of the training data

$$K = \Sigma(u, g, r, i, z)$$

- Choices of Σ give different models.

The polynomial kernel

- K is low rank:

$$K = QQ^T$$

  where Q is n by 21.
- Q can be obtained quickly from the training data.
- Storage requirement for Q is less than K.
- This special structure of K leads to fast linear solvers.

And now

Our challenge is to solve

$$(\lambda^2 I + QQ^T)w = y$$

quickly for large n.

Big-O Notation

A procedure is $O(n)$ if the number of operations required is proportional to the input size n.

Let $n = 10^6$, then

	operations
$O(n)$	10^6
$O(n^3)$	10^{18}

SHERMAN-MORRISON-WOODBURY FORMULA

$$\left(\lambda^2 I + QQ^T\right)^{-1} y = \frac{1}{\lambda^2}\left[y - Q\left(\left(\lambda^2 I + Q^T Q\right)^{-1}\left(Q^T y\right)\right)\right]$$

$\uparrow$ $\Uparrow$ $\uparrow$

$n \times n$ 21×21 21×1

REDUCED RANK METHOD

ADVANTAGES

- This method is $O(n)$, VERY FAST.
- Storage requirement is $O(n)$.

DISADVANTAGE

- Numerical instability

Iterative Refinement

Iterative Refinement is a method for improving the accuracy of a solution produced by solving a linear equation.

Size	Before	After
50	1.7×10^{-6}	9.1×10^{-9}
100	1.2×10^{-5}	2.2×10^{-8}
500	1.1×10^{-4}	2.8×10^{-7}
1000	3.5×10^{-4}	8.4×10^{-7}
5000	4.2×10^{-3}	1.3×10^{-5}

Table: Error before and after iterative refinement

SPEED COMPARISON

Method	n = 1000	n = 10,000	n = 180,000
Gaussian Elimination	0.1510	NA	NA
Reduced Rank w/ Iterative Refinement	0.0035	0.0047	0.1173

TABLE: Running time (in seconds) for finding **w**

Cholesky Rank-One Update

Find L_1, D_1 so that $LDL^T + qq^T = LL_1 D_1 L_1^T L^T$, where

$$L_1 = \begin{pmatrix} 1 & & & & \\ p_2\beta_1 & 1 & & & \\ p_3\beta_1 & p_3\beta_2 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ p_n\beta_1 & p_n\beta_2 & \cdots & p_n\beta_{n-1} & 1 \end{pmatrix}$$

- Storage: only p, β vectors and diagonal entries of D_1 are stored which is $O(n)$
- Operations: finding p, β and D_1 is $O(n)$

Cholesky Rank-One Update for $A = \lambda^2 I + QQ^T = \lambda^2 I + \sum_{i=1}^{21}(q_i q_i^T)$

$$\lambda^2 I + q_1 q_1^T =$$

$$L_1 D_1 L_1^T$$

Cholesky Rank-One Update for $A = \lambda^2 I + QQ^T = \lambda^2 I + \sum_{i=1}^{21}(q_i q_i^T)$

$$\lambda^2 I + q_1 q_1^T + q_2 q_2^T =$$

$$L_1 L_2 D_2 L_2^T L_1^T$$

Cholesky Rank-One Update for $A = \lambda^2 I + QQ^T = \lambda^2 I + \sum_{i=1}^{21}(q_i q_i^T)$

$$\lambda^2 I + q_1 q_1^T + q_2 q_2^T + q_3 q_3^T =$$

How many?

$$L_1 L_2 L_3 D_3 L_3^T L_2^T L_1^T$$

Cholesky Rank-One Update for $A = \lambda^2 I + QQ^T = \lambda^2 I + \sum_{i=1}^{21}(q_i q_i^T)$

$$\lambda^2 I + q_1 q_1^T + q_2 q_2^T + \ldots + q_{21} q_{21}^T =$$

Just 21!

$$L_1 L_2 \ldots L_{21} D_{21} L_{21}^T \ldots L_2^T L_1^T$$

Forward and Backward solvers

Solving for $(L_1 L_2 \ldots L_{21} D_{21} L_{21}^T \ldots L_2^T L_1^T) w = y$

$$\begin{pmatrix} 1 & & & & \\ p_2\beta_1 & 1 & & & \\ p_3\beta_1 & p_3\beta_2 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ p_n\beta_1 & p_n\beta_2 & \ldots & p_n\beta_{n-1} & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

$$w_j = y_j - p_j(\beta_1 w_1 + \beta_2 w_2 + \ldots + \beta_{j-1} w_{j-1})$$

Cholesky Update Method

Advantages

- This method is $O(n)$, FAST.
- Storage requirement is $O(n)$.
- Numerically more stable.

Speed Comparison

Method	n = 1000	n = 10,000	n = 180,000
Gaussian Elimination	0.1510	NA	NA
Reduced Rank w/ Iterative Refinement	0.0035	0.0047	0.1173
Cholesky Update	0.2066	0.4437	6.2529

Table: Running time (in seconds) for finding **w**

Conjugate Gradient Method

- A semi-iterative method that works well with special (i.e. positive definite) linear system.
- In practice, it takes a small number of iterations to get an acceptable solution.
- Each iteration involves a matrix-vector multiplication.
- In general, matrix-vector multiplication requires $O(n^2)$ operations.

Fast Multiplication

The special form of A gives fast matrix-vector multiplication which requires $O(n)$ operations:

$$A\mathbf{u} = \lambda^2\mathbf{u} + Q(Q^T\mathbf{u})$$

Conjugate Gradient Method

Advantages

- It is $O(n)$, FAST.
- Storage requirement is $O(n)$.
- It converges quickly to the actual solution.

Disadvantage

- Susceptible to rounding error for badly behaved matrices.

Speed Comparison

Method	n = 1000	n = 10,000	n = 180,000
Gaussian Elimination	0.1510	NA	NA
Reduced Rank w/ Iterative Refinement	0.0035	0.0047	0.1173
Cholesky Update	0.2066	0.4437	6.2529
Conjugate Gradient	0.2487	0.3058	10.4278

Table: Running time (in seconds) for finding **w**

The Monte Carlo Idea

- Predict the mean of a population using the average of a sample.
- Random (unbiased) sample must be used.
- The sample is generated from a simulated distribution of the population.

The Gibbs Sampler

- Solve special (i.e. symmetric positive definite) linear system using the covariance matrix of a sample of vectors.
- Vectors are generated from simulated normal distributions with means and standard deviations derived from the given linear system.

Gibbs Sampler Method

Works properly if

$$1 < \frac{\text{max eigenvalue}}{\text{min eigenvalue}} < 100$$

Advantage

- Works well for the exponential kernel, the above ratio ≈ 50.

Disadvantage

- Works badly for the polynomial kernel, the above ratio $\approx 10^{10}$.

How to measure a method's performance

Given separate training and testing datasets, each with *ugriz* and redshift values.

1. Fit model using training dataset.
2. Use model to predict redshift values for *ugriz* of testing dataset.
3. Average the square of the errors, then take square root to obtain root mean square error (RMS).
4. Resample using bootstrap 100 times.

Quadratic Regression vs Gaussian Process

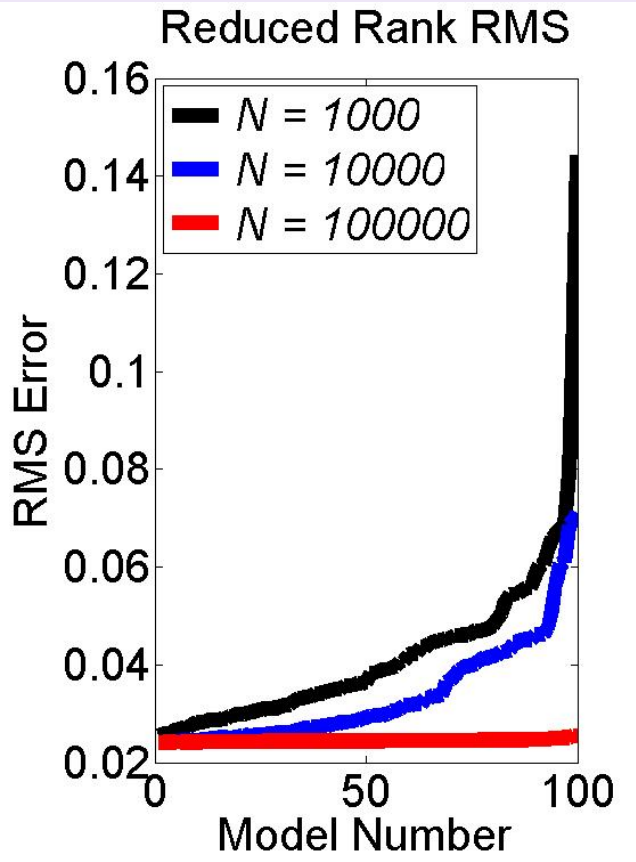
- Quadratic regression
  - Simplest model with *ugriz* and interactions.
  - Baseline method for comparison.
- Gaussian process
  - Polynomial kernel
  - Three improved linear algebra methods: Reduced Rank, Cholesky Update, and Conjugate Gradient.

Time for 100 Bootstrap RMS Calculations

Training Set Size	QR (baseline)	Reduced Rank	Cholesky Update	Conjugate Gradient
1000	10	10	12	14
5000	11	12	23	23
10000	14	15	38	40
50000	37	44	165	202
100000	68	83	387	578
180045	116	142	690	1055

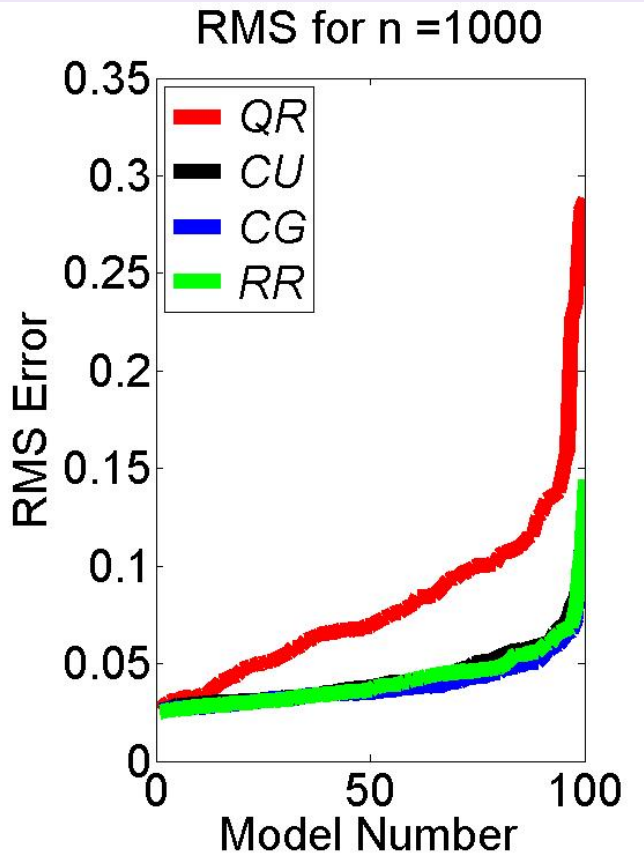
Table: Time (in seconds) for 100 bootstrap RMS calculations

Reduced Rank RMS As Training Size Increases



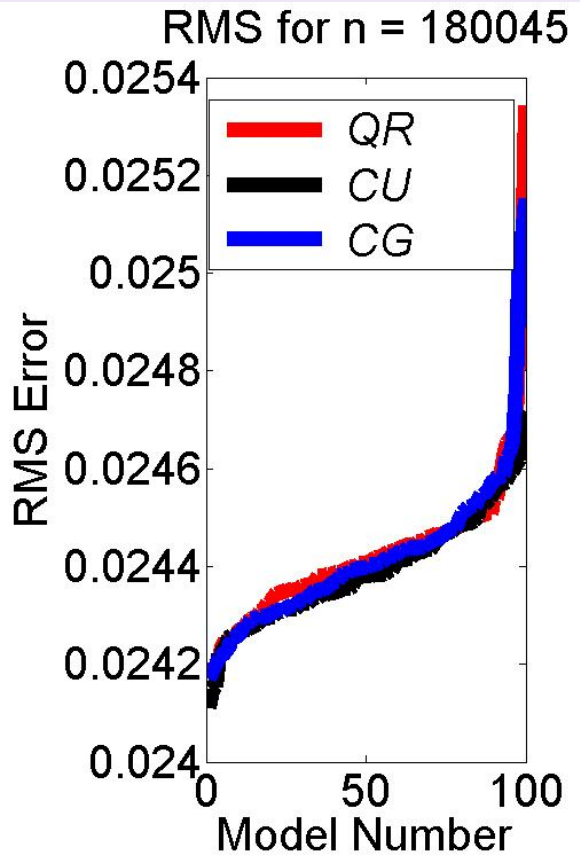
- Behavior for Reduced Rank as training set size increases is the same as for Cholesky Update and Conjugate Gradient.
- As training set size increases, range of RMS decreases.

RMS for Small Training Set Size 1000



- Compare Quadratic Regression, Reduced Rank, Cholesky Update, and Conjugate Gradient for small training set size.
- RMS has wide range from 0.02 to 0.3.

RMS for Large Training Set Size 180045



- Compare Quadratic Regression, Cholesky Update, and Conjugate Gradient for large training set size.
- RMS has narrow range from 0.0240 to 0.0256
- Reduced Rank not shown because of larger RMS range.

Conclusions

- We are able to solve large linear systems fast in a limited memory environment.
- The performance of Gaussian process improves as the size of training sample increases.
- Gaussian process performs better than quadratic regression when small training set is used, BUT it does not when large training set is used.

Future directions

- Explain the data.
- Use optimal value of λ.
- Try exponential kernel function.

Acknowledgement

We would like to thank the **Woodward Fund** for the financial support and the following people for their guidance.

- Drs. Michael Way, Ashok Srivastava, Tim Lee, Paul Gazis (NASA scientists)
- Dr. Tim Hsu (CAMCOS director)
- Drs. Bem Cayco, Wasin So (Faculty advisors)
- Drs. Leslie Foster, Steve Crunk (SJSU faculty)

THANK YOU!

Any Questions?