Virtual Sensors and Large-scale Gaussian Processes

Ashok N. Srivastava, Ph.D.
Principal Investigator, IVHM Project
Group Lead, Intelligent Data Understanding
ashok.n.srivastava@nasa.gov

NASA Data Systems

- Earth and Space Science
  - Earth Observing System generates ~21 TB of data per week.
  - Ames simulations generating 1-5 TB per day
- Aeronautical Systems
  - Distributed archive growing at 100K flights per month with 2M flights already.
- Exploration Systems
  - Space Shuttle and International Space station downlinks about 1.5GB per day.

Developing *Virtual* Sensors

- Virtual Sensors predict the value of one sensor measurement by exploiting the nonlinear correlations between its values and other sensor readings.
- Useful for emulating sensors back in time or estimating the value of one sensor based on other sensor measurements

Z: Sensors measurements
λ: Wavelength or Frequency
u: Position

$$Z(\mathbf{u},\lambda,t) = [Z_{\mathbf{u}}(\lambda,t)]$$
$$= [Z_{u_1}(\lambda,t), Z_{u_2}(\lambda,t), \ldots, Z_{u_n}(\lambda,t)]^T$$

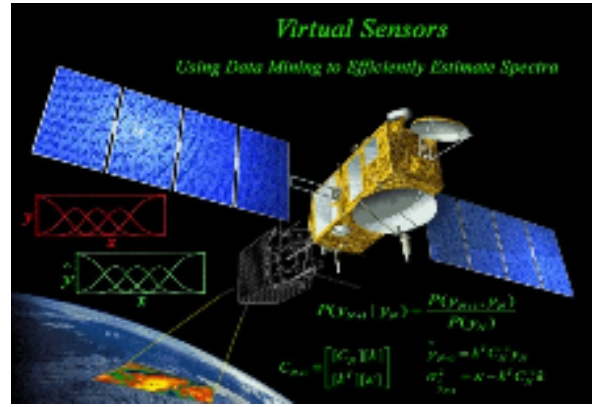
$$\mu(Z(\mathbf{B})) = \int \Gamma(Z(\mathbf{B}))Z(\mathbf{B})d\mathbf{B}$$

Predicted Sensor Measurement

$$\sigma^2(Z(\mathbf{B})) = \int [\Gamma(Z(\mathbf{B})) - \mu(Z(\mathbf{B}))]^2 Z(\mathbf{B})d\mathbf{B}$$

Estimated Uncertainty

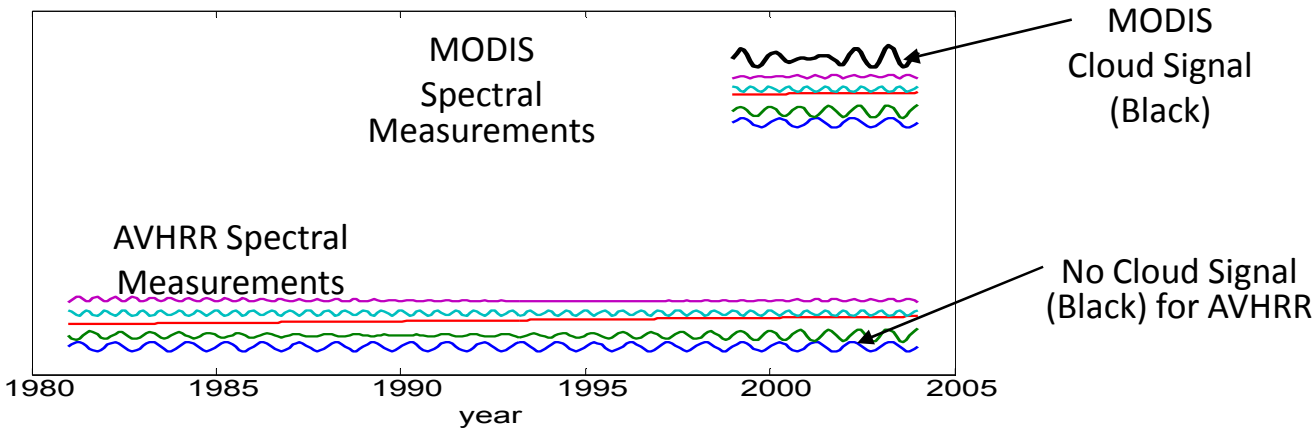
Earth and Space Sciences



Aeronautics and Space Systems

Virtual Sensors in the Earth Sciences

- Detecting change in cloud cover
  - New sensors on the MODIS system can detect clouds over snow and ice in the 1.6μm band (circa 1999).
  - Difficult over snow and ice-covered surfaces because of low contrast in visible and thermal infrared wavelengths.
  - Older sensors from the AVHRR system do not detect cloud cover over snow and ice because of poor contrast.
  - Predict 1.6μm channel using a Virtual Sensor



- Detecting land cover change using surface reflectance measurement
  - Predict missing surface reflectance data in one sensor channel using observations from a combination of other channels.
  - Create a high quality complete data record for use in new Earth science analysis and explorations.
  - Study the residual pattern of the prediction algorithm across years in order to make significant conclusions regarding change in land cover across the globe.

A. N. Srivastava, N. C. Oza, and J. Stroeve, "Virtual Sensors: Using Data Mining Techniques to Efficiently Estimate Remote Sensing Spectra," Special Issue on Advanced Data Analysis, IEEE Transactions on Geoscience and Remote Sensing, March 2005.

Prediction Method

- Build a prediction model that offers
  - Interpretability
  - Confidence in the prediction
  - Scalability
- Choices
  - Linear regression
  - Quadratic & cubic regression
  - Polynomial regression
  - Neural networks
  - KD trees & Bayesian classification approaches
  - Support vector machines & Gaussian Process regression

Gaussian Process Regression

- Gaussian Process regression uses Bayesian inferencing under additive Gaussian noise assumption to learn a function on a given data set with some confidence

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} \quad , \quad y = f(\mathbf{x}) + \epsilon \quad , \text{ where } \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

- Likelihood function: $P(\mathbf{y}|X, \mathbf{w}) \sim \mathcal{N}(X^T\mathbf{w}, \sigma^2 I)$
- Gaussian prior over parameters: $P(\mathbf{w}) \sim \mathcal{N}(0, \Sigma_p)$
- Inference is the posterior distribution over the weights $\mathbf{w}$ given by

$$P(\mathbf{y}|X) = \int (P(\mathbf{y}|X, \mathbf{w})P(\mathbf{w})d\mathbf{w}$$

- Predictive distribution is:

$$P(f^*|\mathbf{x}^*\mathbf{y}, X) = \mathcal{N}(\frac{1}{\sigma^2}\mathbf{x}^{*T}A^{-1}X\mathbf{y}), \mathbf{x}^{*T}A^{-1}\mathbf{x}^* \text{ , where } A = \Sigma_p^{-1} + \frac{1}{\sigma^2}XX^T$$

Gaussian Process Regression

Training data

- X data matrix of observations – n x d
- y vector of target data – n x 1

Test data

- X* matrix of new observations – n* x d

Covariance function

$$K_{ij} = k(x_i, x_j), K^*_{ij} = k(x^*_i, x_j)$$

Goal

- Predict y* corresponding to X*

Model building

- Train hyperparameters on a sample of X
- Compute covariance matrix K (n x n)

Prediction

- Compute cross covariance matrix K* (n* x n)
- Compute mean prediction on y* using

$$\widehat{y}^* = K^*(\lambda^2 I + K)^{-1} y$$

- Compute variance of prediction using

$$C = K^{**} - K^*(\lambda^2 I + K)^{-1} K^{*T}$$

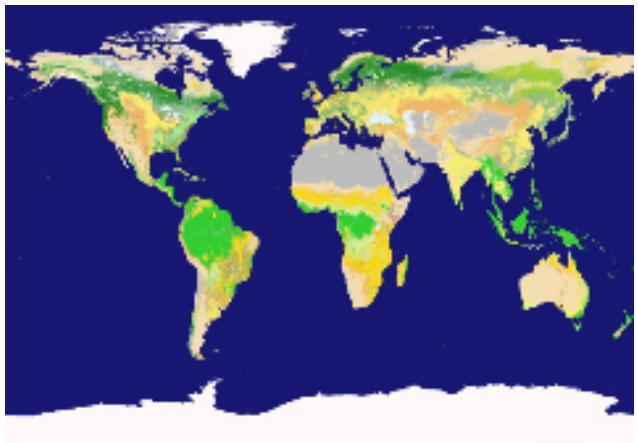
Algorithm Analysis

- Storage Complexity: Storing covariance matrix $O(n^2)$
- Time Complexity: Computing matrix inversion $O(n^3)$

Real-life Data Sets

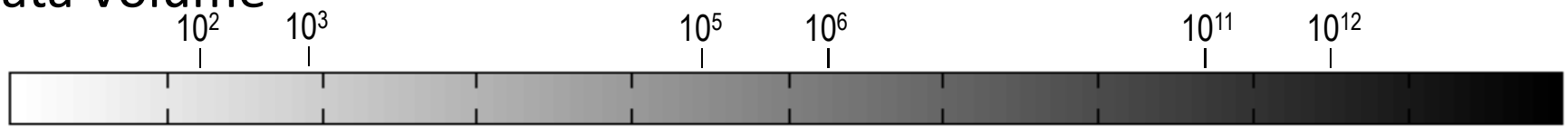
Napa

California

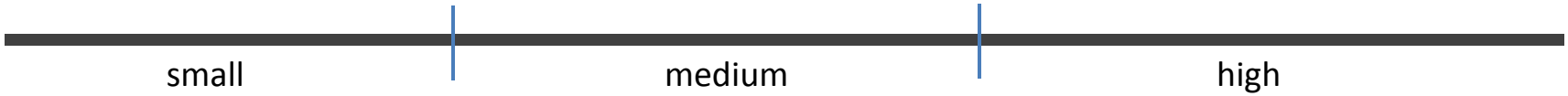


World

Data Volume



Number of modes

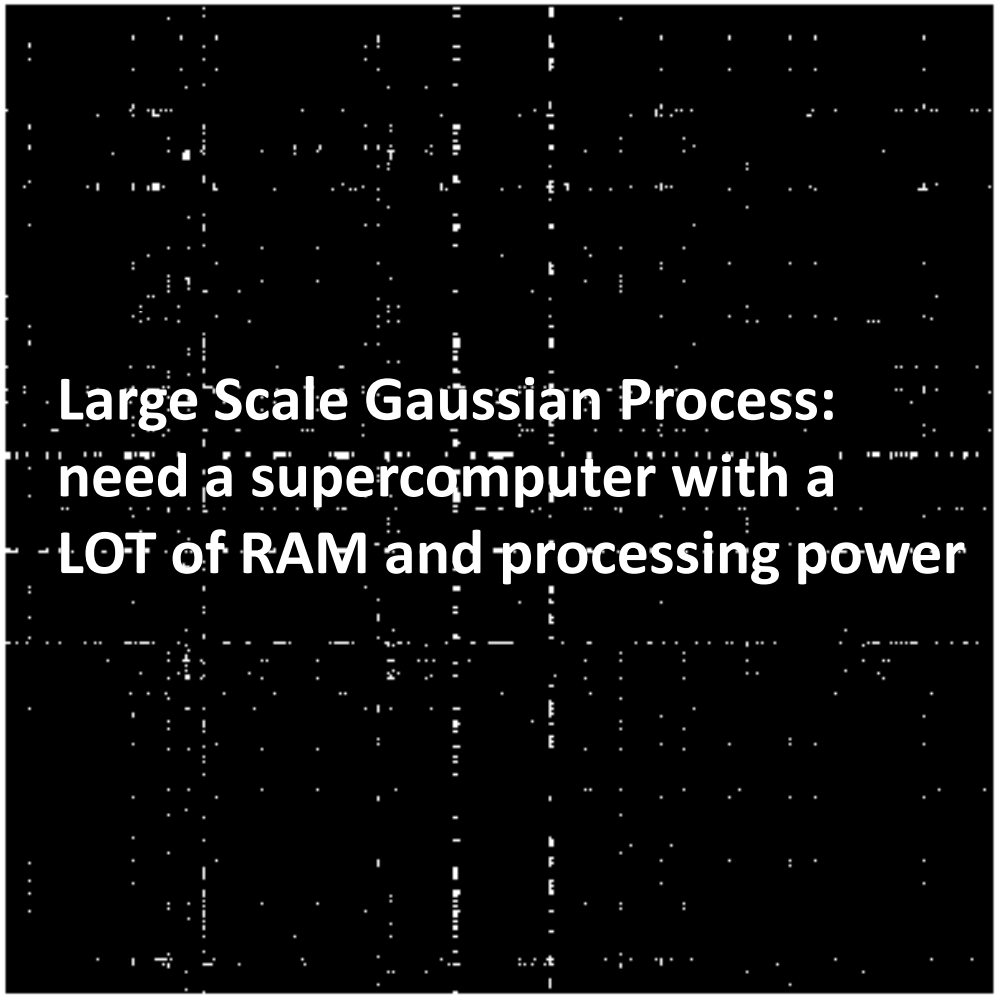


Gaussian Process Regression - Illustration

- Size of training data : 10^2

Gaussian Process Regression - Illustration

- Size of training data: 10^6



Low-rank Approximations

- Numerical approximation techniques exist such as Subset of Regressors, Q-R decomposition, V method
  - Numerical instability can be a problem
- ***Solution:*** ***<u>Stable GP</u>*** (V formulation using Cholesky decomposition with pivoting)
- The V-Formulation provides an extremely scalable and numerically stable method to compute Gaussian Process Regression for arbitrary kernels.

Computational Challenges

- Subset of Regressors (Wahba, 1990)

$$\widehat{y}_N^* = K_1^*(\lambda^2 K_{11} + K_1^T K_1)^{-1} K_1^T y$$

where,

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} = (K_1 \quad K_2), K^* = (K_1^* \quad K_2^*)$$

- Memory: Storing covariance matrix – *O(nm)*
- Time: Solving linear systems – *O(nm²)*
- Numerical stability:???

Cures for Numerical Instability

Approach

1. Select columns to make K_1 well conditioned
2. Use stable technique for least squares problem such as
    - QR factorization
    - V method
3. Requirement: maintain $O(nm)$ memory use and $O(nm^2)$ efficiency.

Column Selection

1. Use Cholesky factorization with pivoting to partially factor K
2. selects appropriate columns for K_1
3. K_1 will be well conditioned if $cond(K_1)$ is O(condition of optimal low rank approximation).

Stable GP

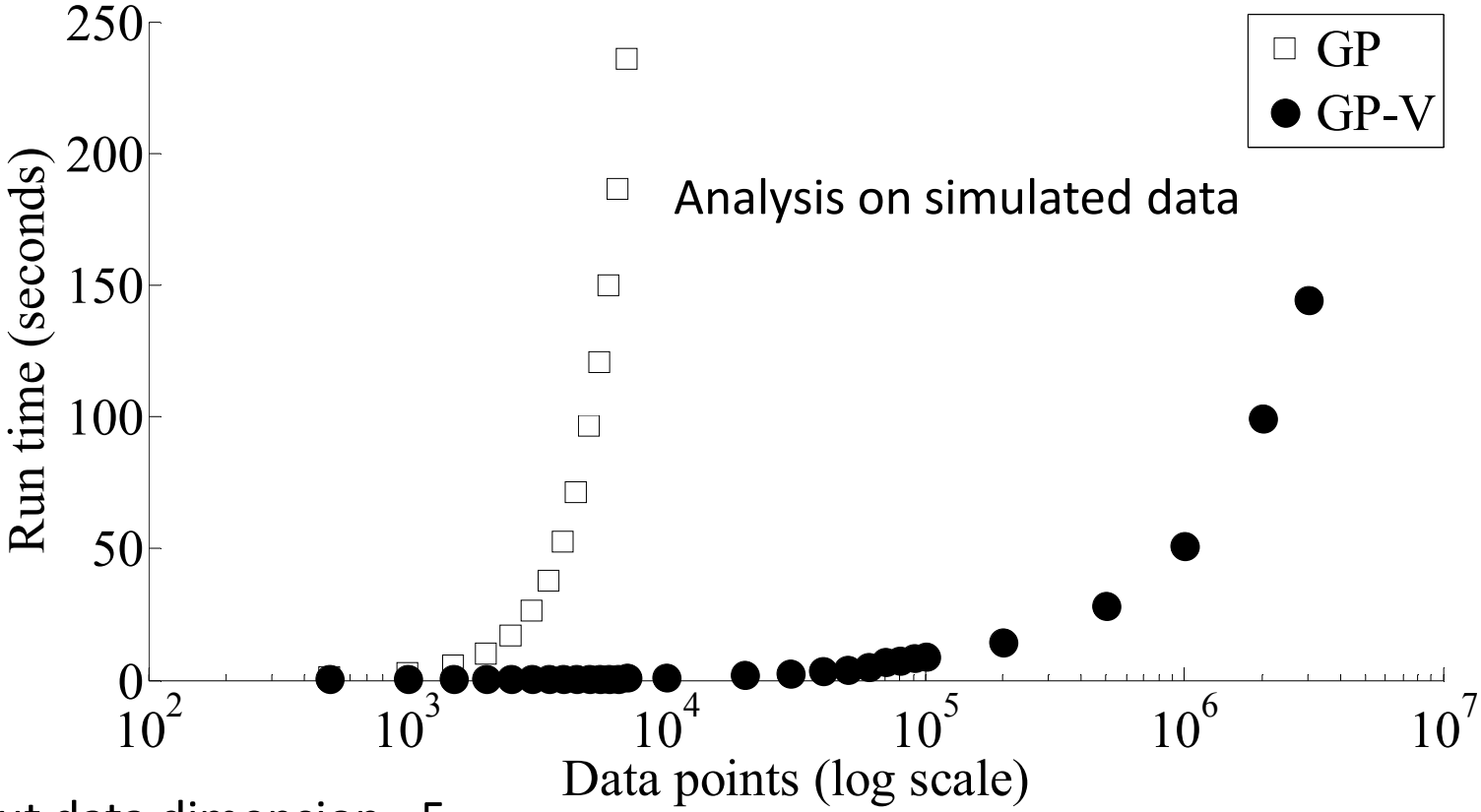
- Approximate $K_1 \approx V V_{11}^T$ by Cholesky factorization where V is is $n \times m$ and V_{11} is $m \times m$
- Predicted mean can be rewritten as

$$\widehat{y}^* = V^*(\lambda^2 I + V^T V)^{-1} V^T y$$

- Inverting $m \times m$ instead of $n \times n$ matrix
- Method is numerically stable
- Method can be faster and needs less memory

L. Foster, A. Waagen, N. Aijaz, M. Hurley, A. Luis, J. Rinsky, C. Satyavolu, M. J. Way, P. Gazis, and A. N. Srivastava, "Stable and Efficient Gaussian Process Calculations," Journal of Machine Learning Research, 10(Apr):857--882, 2009.

GP–V formulation Results



Input data dimension =5

No. of sample points = 3 million

Run time = time to build the model + time to evaluate 500 test points

Maximum rank = 25 (used for GP-V)

Hyper parameters are trained on 100 sample points

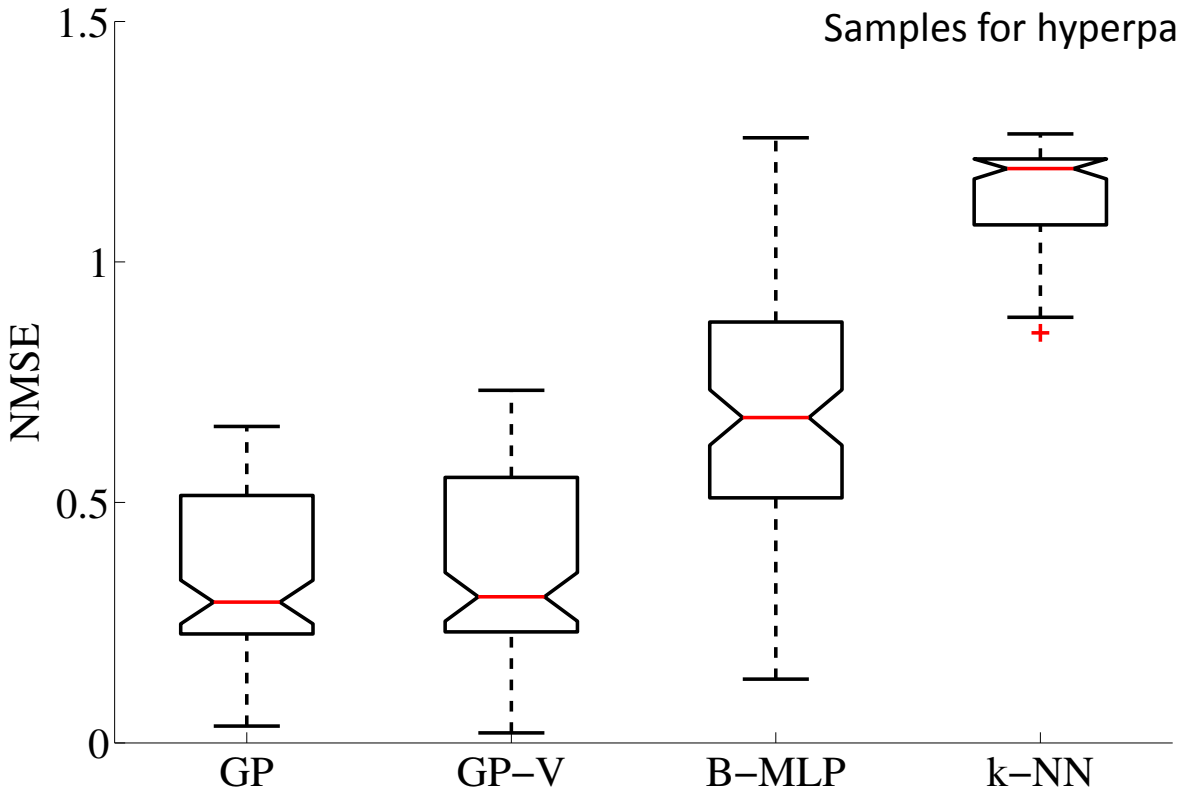
Stable GP Results

- Accuracy

Based on 100 runs on NH_3 laser data

d = 34, n = 1166, maximum rank = 340

Samples for hyperparameter training = 300



With *low-rank matrix inversion approximation using pivoting* Stable GP performed close to standard GP.

A. N. Srivastava, S. Das. (2009).“Detection and prognostics on low-dimensional systems,” Transactions of Systems, Man and Cybernetics Part C 39, 1, 2009.

Mixture of Gaussian Process Experts

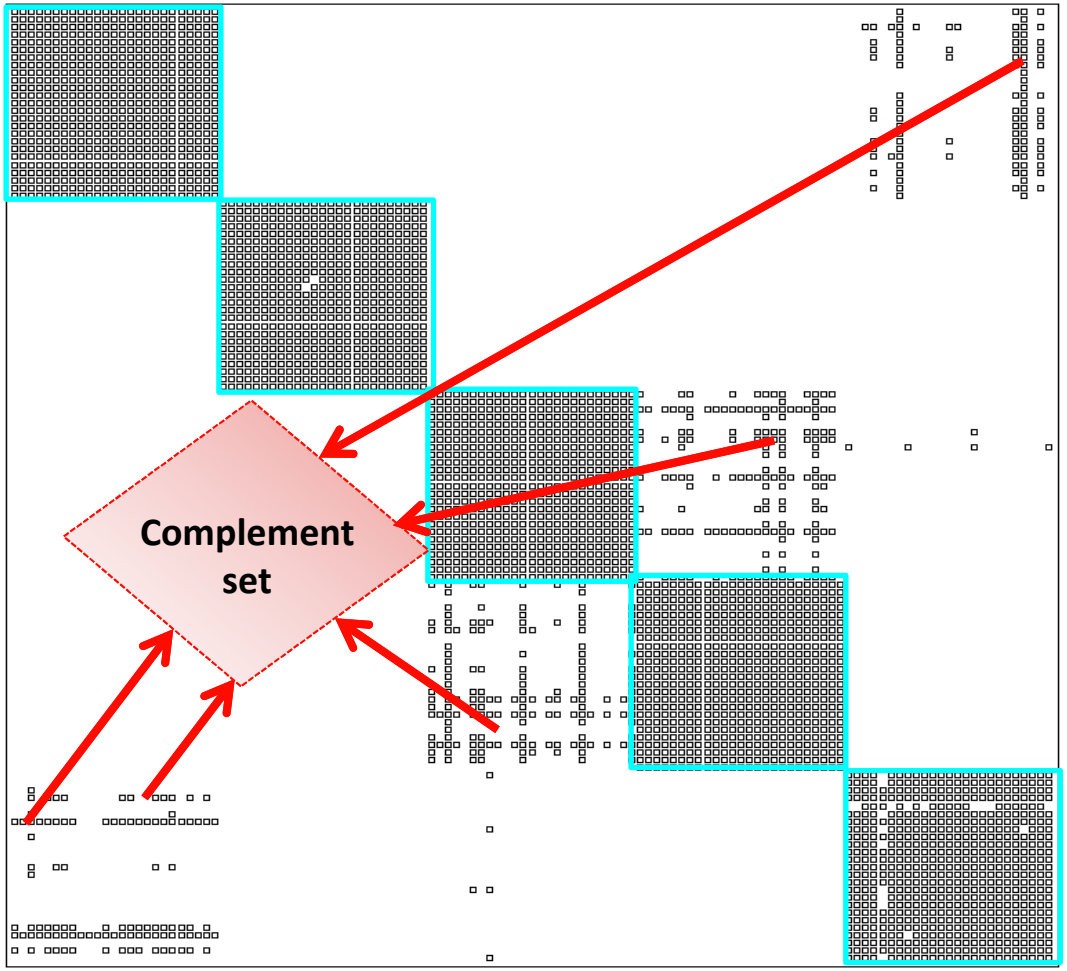
- Numerical approximations do not work for multimodal data
- Solution: Mixture of Gaussian Process Experts [1,2]
  - Gating network decides which point is best predicted by each expert
  - Expectation Maximization based algorithm for computing the gating network
  - Scales up to the order of 10^3 data observations

[1] V. Tresp. Mixtures of Gaussian processes. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13, pages 654–660.* MIT Press, 2001.

[2] C. E. Rasmussen and Z. Ghahramani. (2002). Infinite Mixtures of Gaussian Process Experts. In *Advances in Neural Information Processing Systems 14*. The MIT Press, 2002.

Mixture of Experts: Blocks

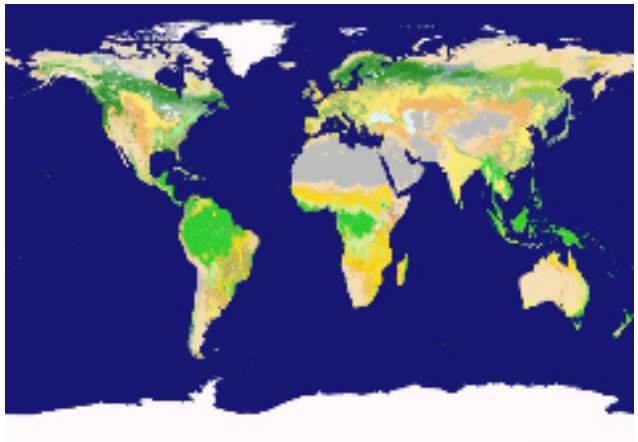
- Size of training data: 10^6



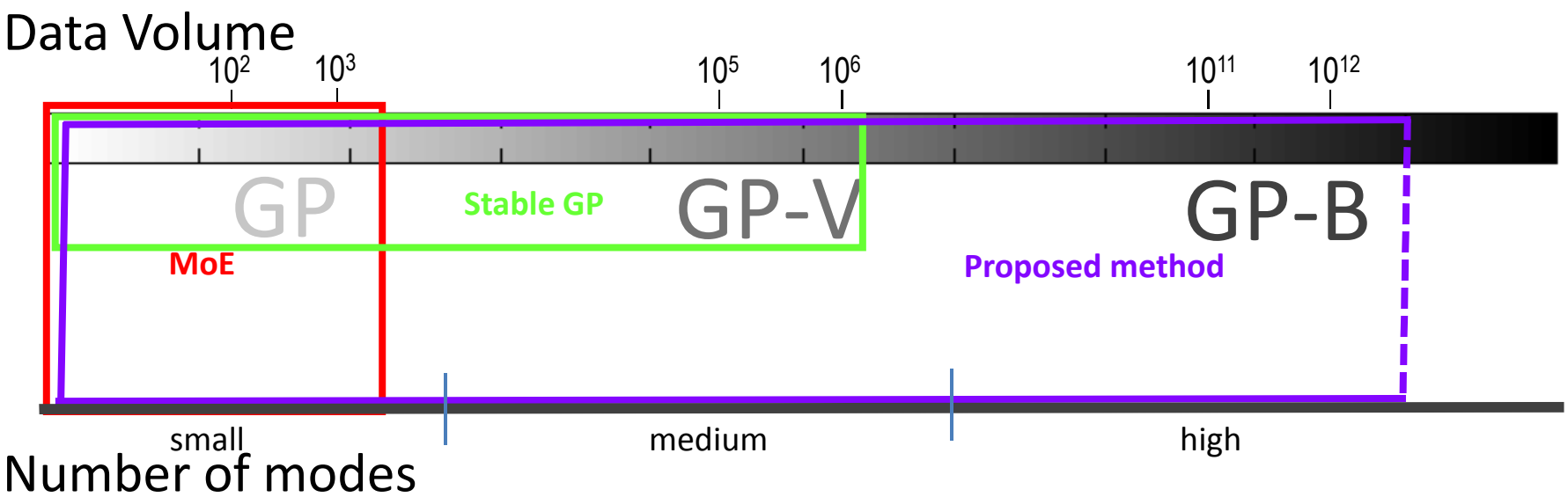
Real-life Data Sets: Multimodality

Napa

California



World



GP-B Regression

- Identify inherent multi-modality of data and partition data based on the number of observed modes – most earth science data sets are multimodal
- Build Gaussian Process experts on each data subset
- Make final prediction a weighted combination of the prediction of the individual experts

GP-B Algorithm has 3 steps:

1. Partitioning the data set.
2. Training the models based on the partitions
3. Predicting unknown data using the model built

GP-B Regression – Data Partitioning

1. Based on domain knowledge and exploratory data analysis, identify the number of clusters in the data. Let number of clusters = k.
2. Use spectral clustering for partitioning the data into k clusters.
   - Nyström approximation with s samples [1]: $O(nds)$
   - t-nearest neighbor based parallel spectral clustering [2]: $O(n^2d) + O(n^2 \log t)$
   - Fast spectral clustering (KASP) [3]: Can scale upto 10^6 elements

[1] C. Fowlkes, S. Belongie, F. Chung, J. Malik. (2004). Spectral grouping using the Nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26,* 214–225.

[2] W. Chen, Y. Song, H. Bai, C. Lin, E. Chang. (2010). Parallel Spectral Clustering in Distributed Systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*

[3] D. Yan, L. Huang, M. Jordan. (2009). Fast Approximate Spectral Clustering Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 907-916

GP-B Regression: Training

1. Input: Result of data clustering
2. Fit a mixture of Gaussian distributions to the data clusters.
3. Compute entropy of each data point based on the distribution parameters.
4. Identify high entropy points and remove all those points from each of the clusters to form a separate cluster with those points.
5. Recompute distribution parameters for each of the k existing distributions and the new (k+1st) cluster
6. Train hyperparameters and covariance matrices for each of these Gaussian Process experts.

GP-B Regression: Testing

- Input: Test points, Gaussian process expert model
- Output: Prediction of target corresponding to each input
- Final prediction is given by the following equation:

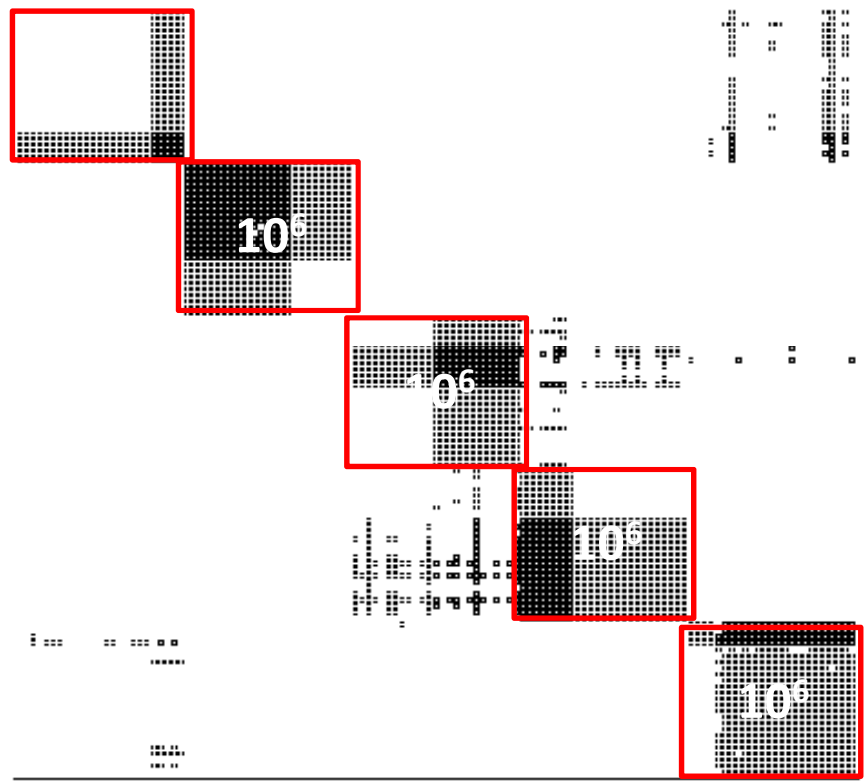
$$\sum_{j=1}^{k} g_j(\mathbf{x})\hat{\mathbf{y}}_j^*$$

where $\hat{\mathbf{y}_j}^* = K_j^*(\lambda_j^2 I + K_j)^{-1} y_j$ and

$g_j(\mathbf{x})$ is the weight of the expert prediction calculated as the cluster membership probability of the test point.

GP-B Performance Analysis

- For number of modes k and number of observations n, prediction is $O((n/(k+1))^3)$
  - Higher scalability
  - Decomposability for distributed computation
  - Higher interpretability as different models predict different geographical regions accurately



Use numerical approximation technique for each of the experts individually

GP-B Empirical Evaluation

Data set	Modes	Size	Details
Tanh[1]	2	900 x 3	$x_{t+1} = \begin{cases} 2(1-x_t^2)-1 \text{ if } s=1 \\ tanh(-1.2x_t^2+\epsilon_{t+1}) \text{ if } s=0 \text{ and } \epsilon \sim \mathcal{N}(0,0.1) \end{cases}$
D[2]	4	16,000 x 2	Computer generated univariate time series describing motion of a damped driven particle in an asymmetric 4-d four-well potential
California[3]	10**	150,000,000 x 4	MODIS 8 day surface reflectance BRDF-adjusted from Terra and Aqua measured in 7 different wavelengths

- Experimental Setup
  - Centralized experiments: 64-bit 2.33 GHz quad core Dell Precision 690 desktop running Red Hat Enterprise Linux version 5.4 having 20GB of physical memory.
  - Parallel experiments: 64-bit Linux cluster consisting of 16 slave nodes; each node is a dual processor 1-U server containing two, quad-core Intel Xeon processors @ 2.66GHz totaling 128 cores and 128GB Ram (1Gb/Core).
  - The algorithms were implemented in Matlab and run on version R2010a.

** http://casoilresource.lawr.ucdavis.edu/drupal/files/images/GIS_veg1.jpg

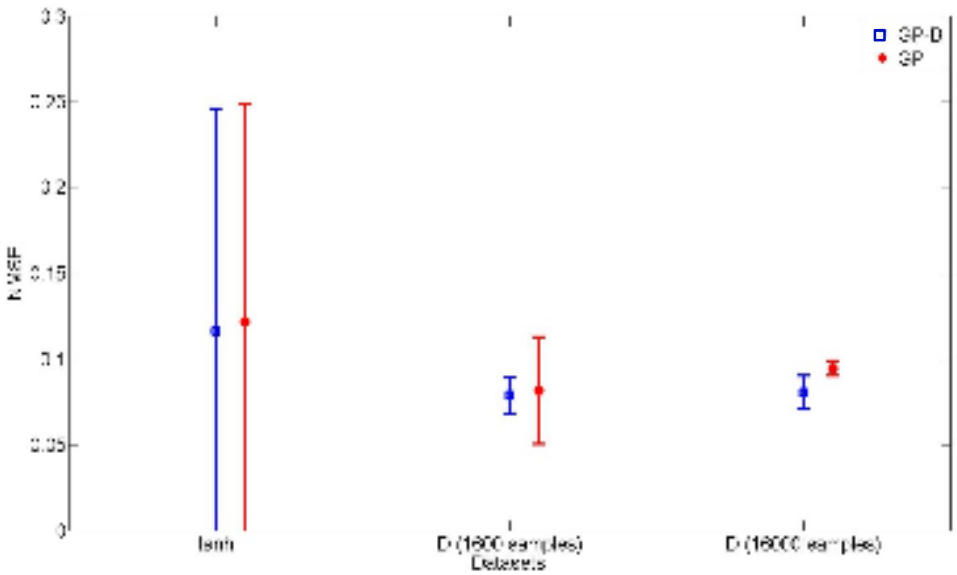
1 A. Weigend, M. Mangeas, and A. Srivastava, "Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting," *Int. J. Neural Syst., vol. 6, pp*. 373–399, 1995.

2 The Santa Fe Time Series Competition data. http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html#setD

3 USGS Land processes Distributed Active Archive Center https://lpdaac.usgs.gov/lpdaac/products/modis_products_table

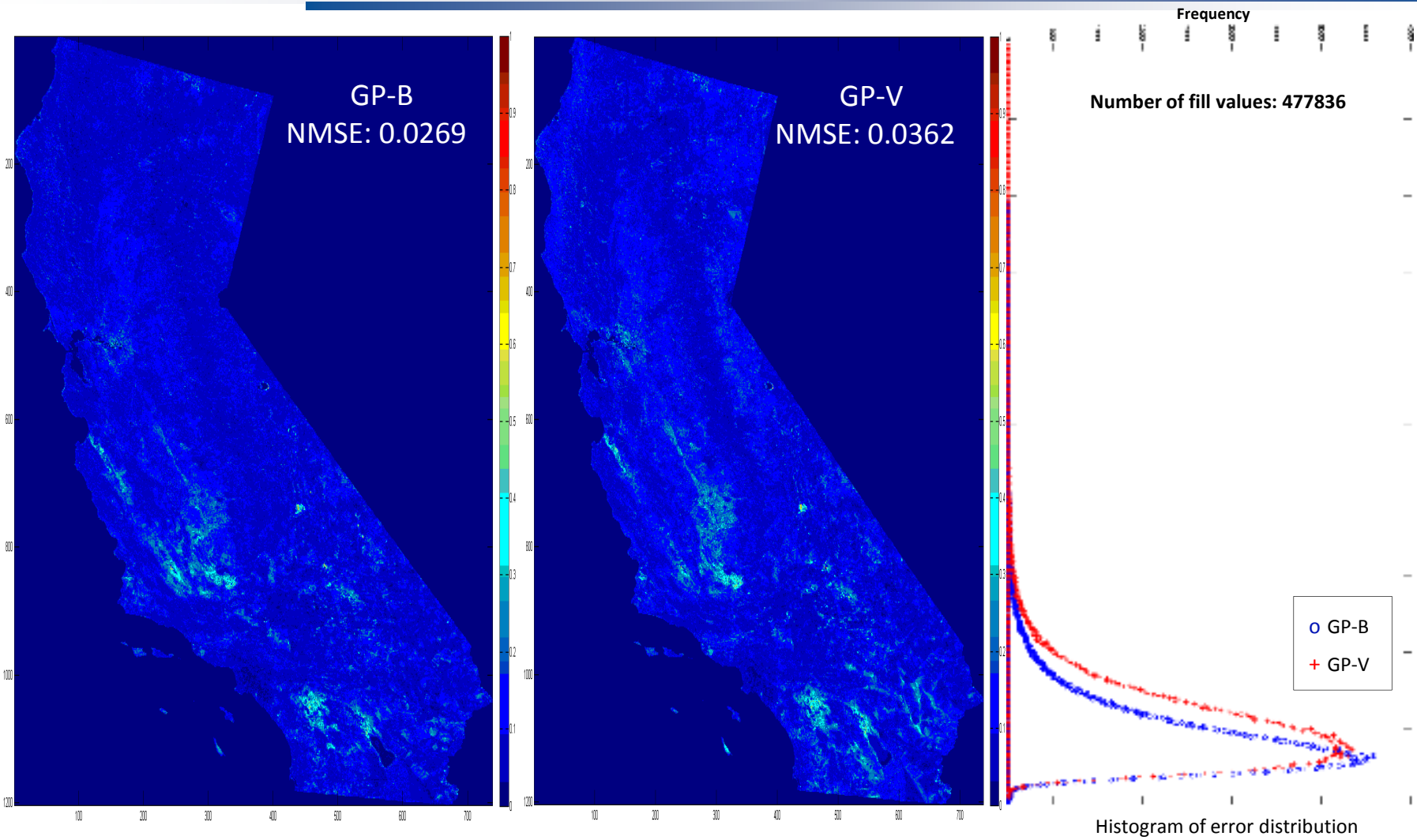
GP-B Results

- Accuracy: Plot of Normalized Mean Squared Error



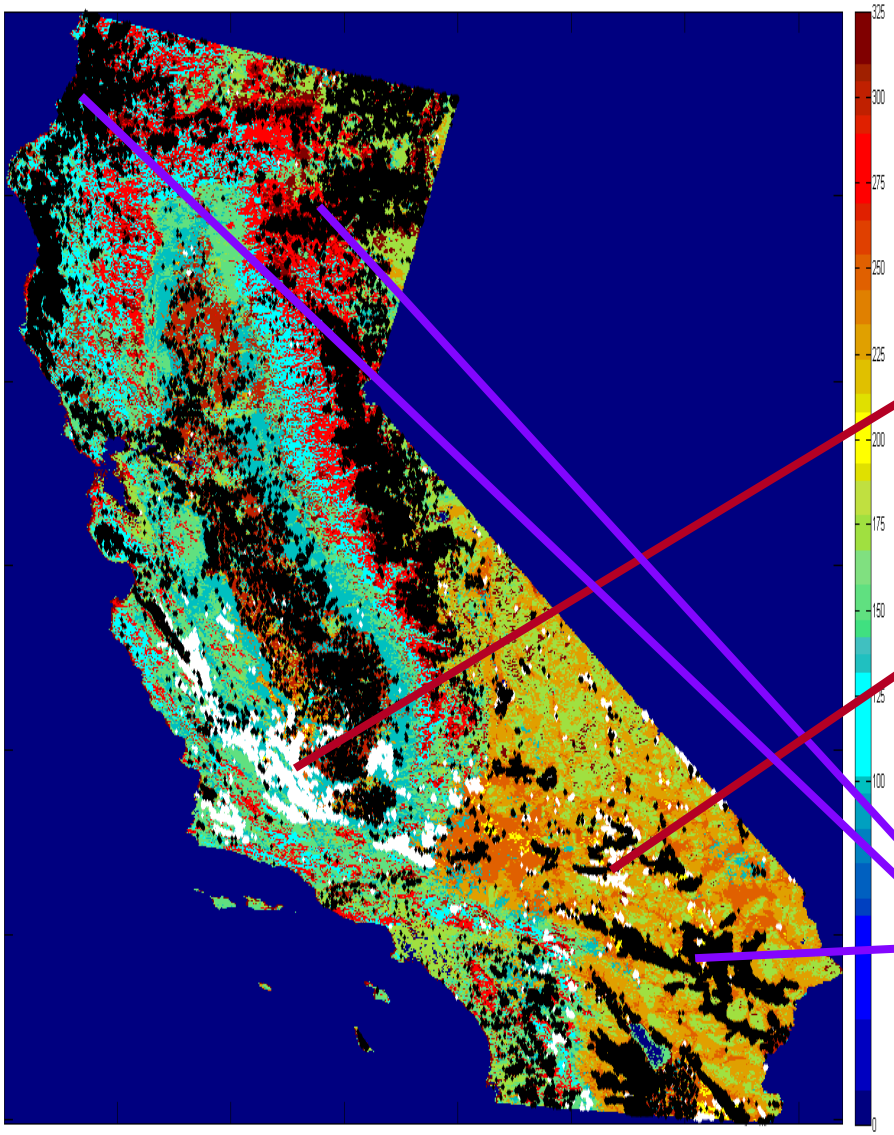
The NMSE for the first 2 data sets for GP-B is comparable with that of classical GP. The NMSE for the last data set for GP-B is comparable to that of GP-V. All results are average of 50 trials.

GP-B Results



Color map of normalized residual for prediction of band 6 using bands 1, 2 and 5 of California MODIS surface reflectance data.

GP-B Results



- Top 5% cases where GP-B performed better
- Top 5% cases where GP-V performed better

1. Is the model not good enough?
   - Performs well for other areas of cluster 1
2. Is the data noisy?
   - Poor data quality will yield better accuracy values for loose model
3. Has the model evolved over time?
   - Training on 2001 and prediction on 2002

1. Too many clusters?
   - Region corresponds to cluster 5
   - Only 2% data points in cluster 5

1. Less accurate model due to insufficient training samples for hyperparameters?
   - Smaller clusters (cluster 5, 9, 10) not well represented due to uniform sampling for hyperparameter training in GP-V

California color coded into 10 clusters based on surface reflectance.

Conclusion

- New Gaussian Process regression algorithm for Virtual Sensors in Earth Science data
- Scalability dependent on
  - Number of dimensions of input data
  - Number of modes in input data
  - Choice of clustering algorithm
- Accuracy dependent on
  - Choice of covariance function
  - Choice of number of clusters and entropy threshold
  - Sparsity in the covariance matrix constructed from the data