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

Coauthors: Kamalika Das, Ph.D. and Santanu Das, Ph.D.

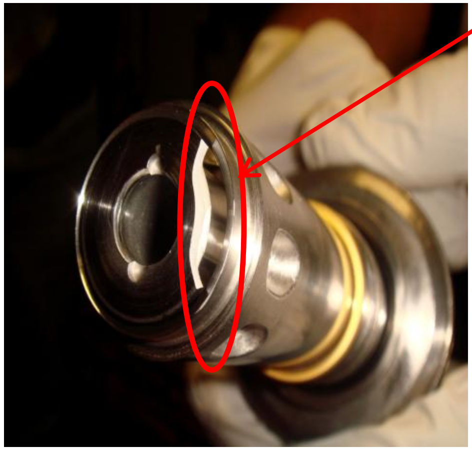
NASA Data Systems

- Earth and Space Science
  - Earth Observing System generates ~21 TB of data per week.
  - NASA 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.

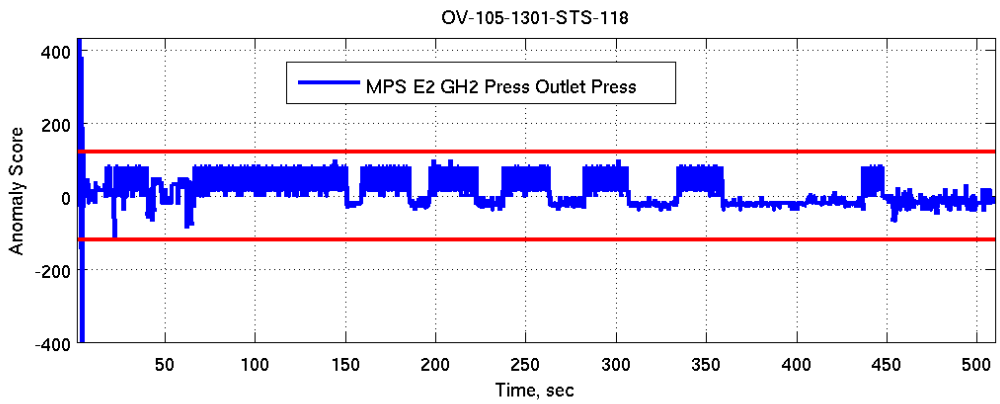
Developing *Virtual* Sensors

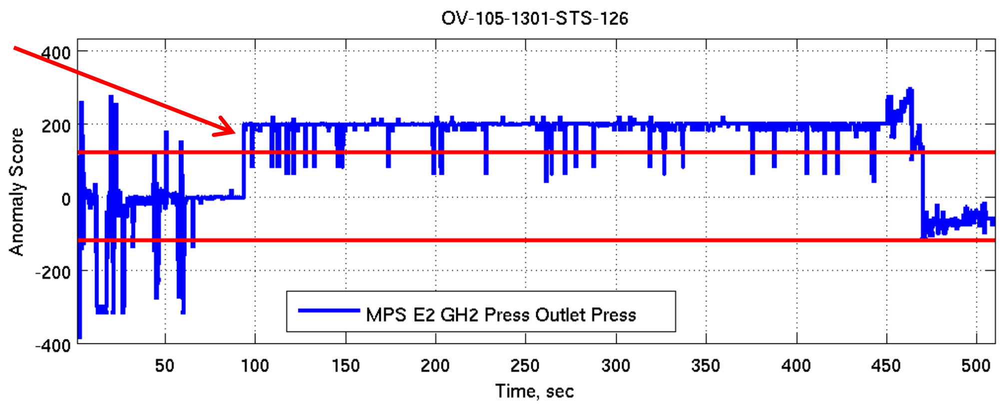
- Virtual Sensors predict the value of **one sensor measurement** by learning the nonlinear correlations between its values and **potentially hundreds** of other sensor measurements.

Space Shuttle Example: Detecting Anomalies in the Main Propulsion System



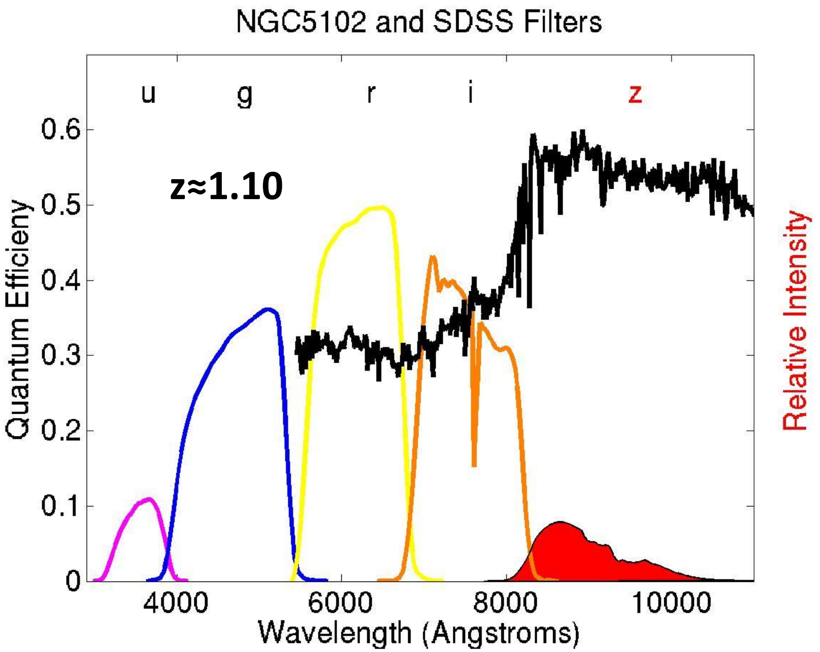
Broken Valve

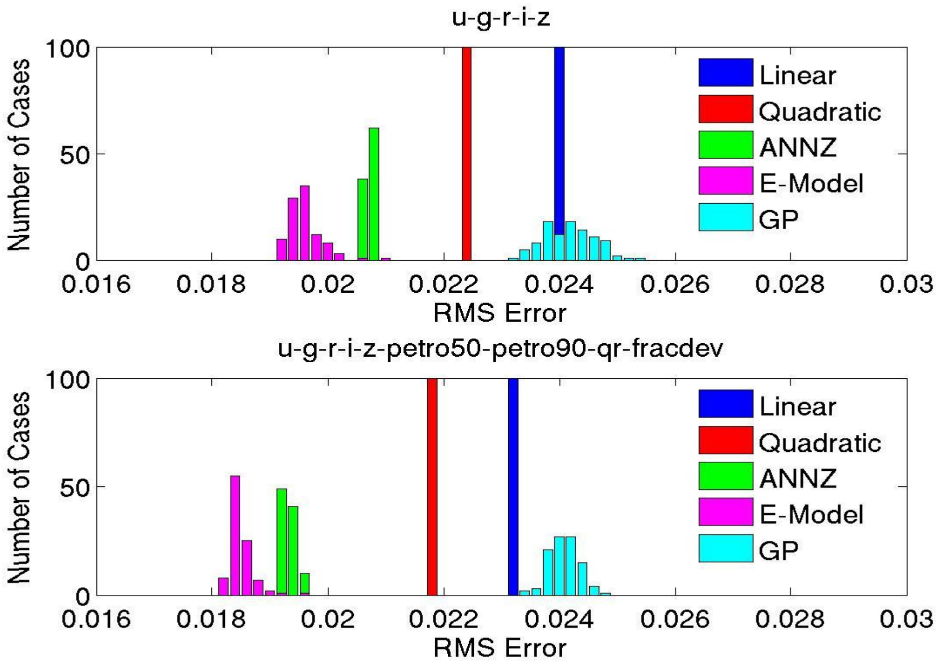




B. Matthews, A. N. Srivastava, et. al., "Multidimensional Anomaly Detection on the Space Shuttle Main Propulsion System: A Case Study", submitted to AIAA Journal on Aerospace Computing, Information, and Communication, 2010.

Virtual Sensors for Estimating the Large Scale Structure of the Universe

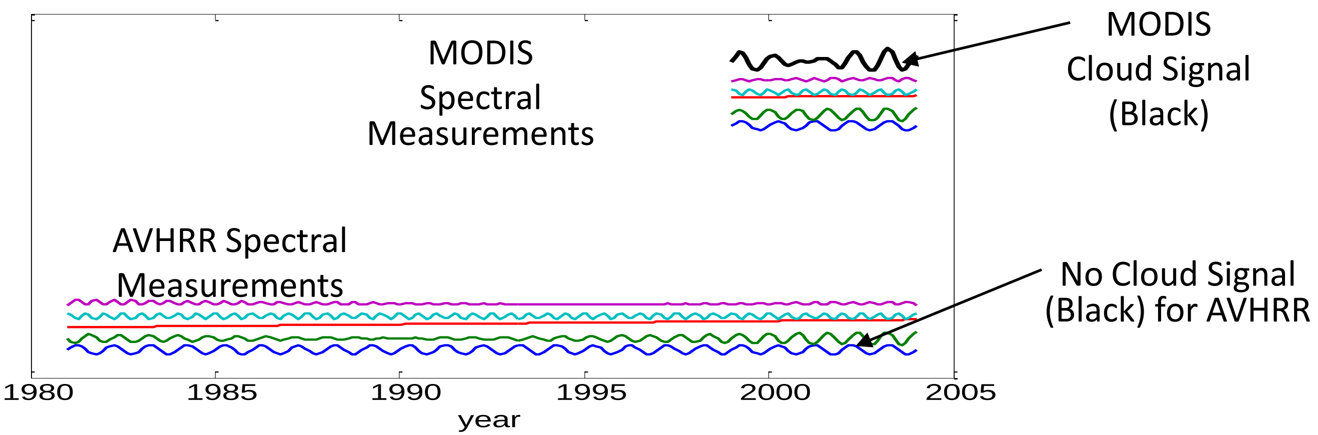




- M. Way, and A. N. Srivastava, "Novel Methods for Predicting Photometric Redshifts," Astrophysical Journal, 2006.
- L. Foster, A, A. Waagen, N. Aijaz, M. Hurley, A. Luis, J. Rinsky, C. Satyavolu, M. J. Way, P. Gazis, and A. N. Srivastava, "Stable and Efficient Gaussian Process Calculations," Journal of Machine Learning Research, 10(Apr):857--882, 2009.
- M. Way, L. Foster, P. Gazis, and A. N. Srivastava, "New Approaches to Photometric Redshift Prediction," Astrophysical Journal, 2009.

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 Methods for Virtual Sensors

- Build a prediction model that offers
  - Interpretability
  - Confidence in the prediction
  - Scalability
- Choices of Regression Functions
  - Linear regression
  - Generalized Linear Models such as Elastic Nets* (perform Lasso and Ridge Regression simultaneously)
  - Neural networks
  - Support vector machines & Gaussian Process Regression

* J. Friedman, T. Hastie, R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent", Journal of Statistical Software, 2010.

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 $O(n^3)$

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 – $O(nm^2)$
- Can be numerically unstable

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

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

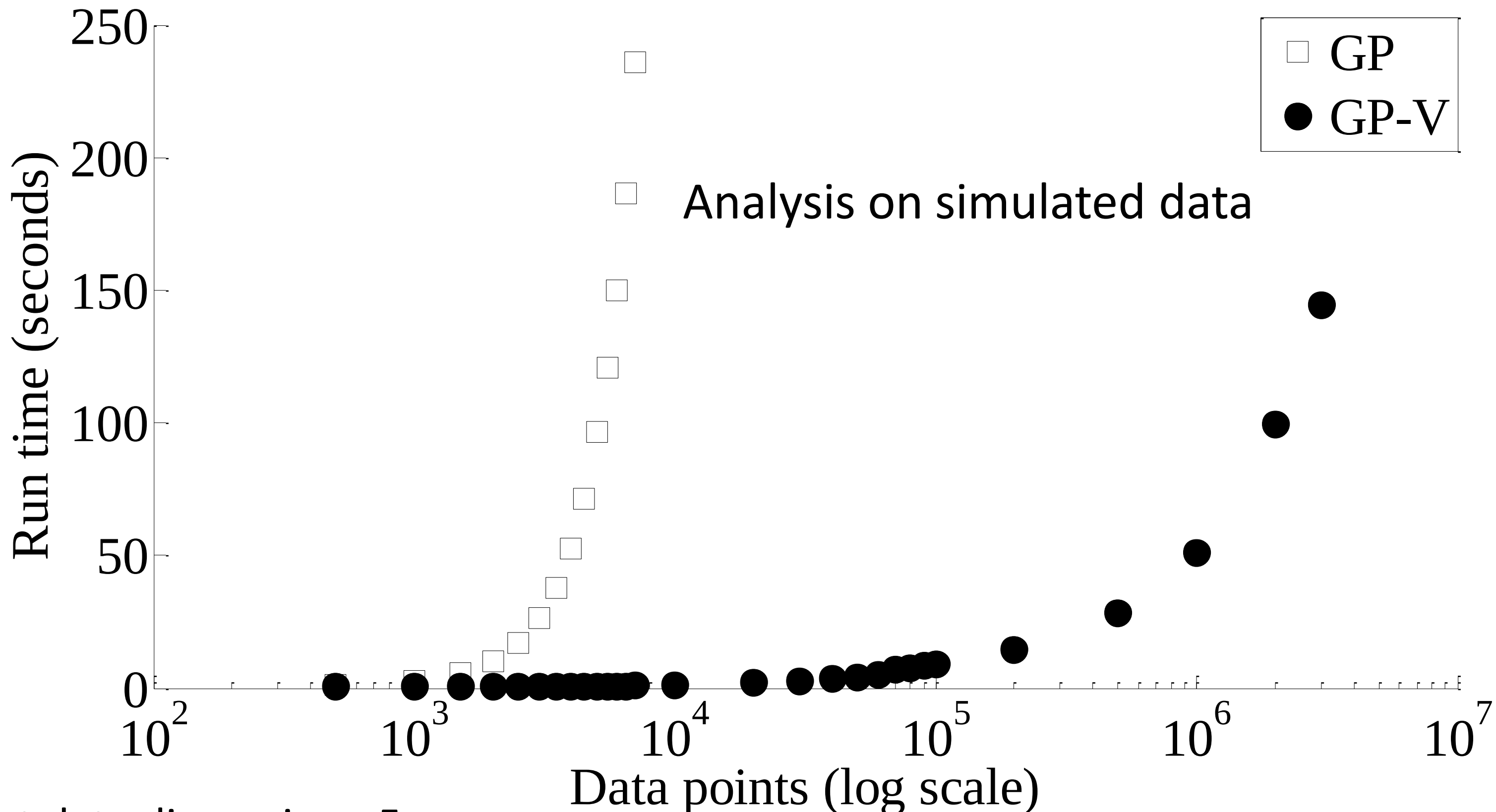
- Inverting $m \times m$ instead of $n \times n$ matrix

- 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: Scaling to 3 million points



Input data dimension =5

Number 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

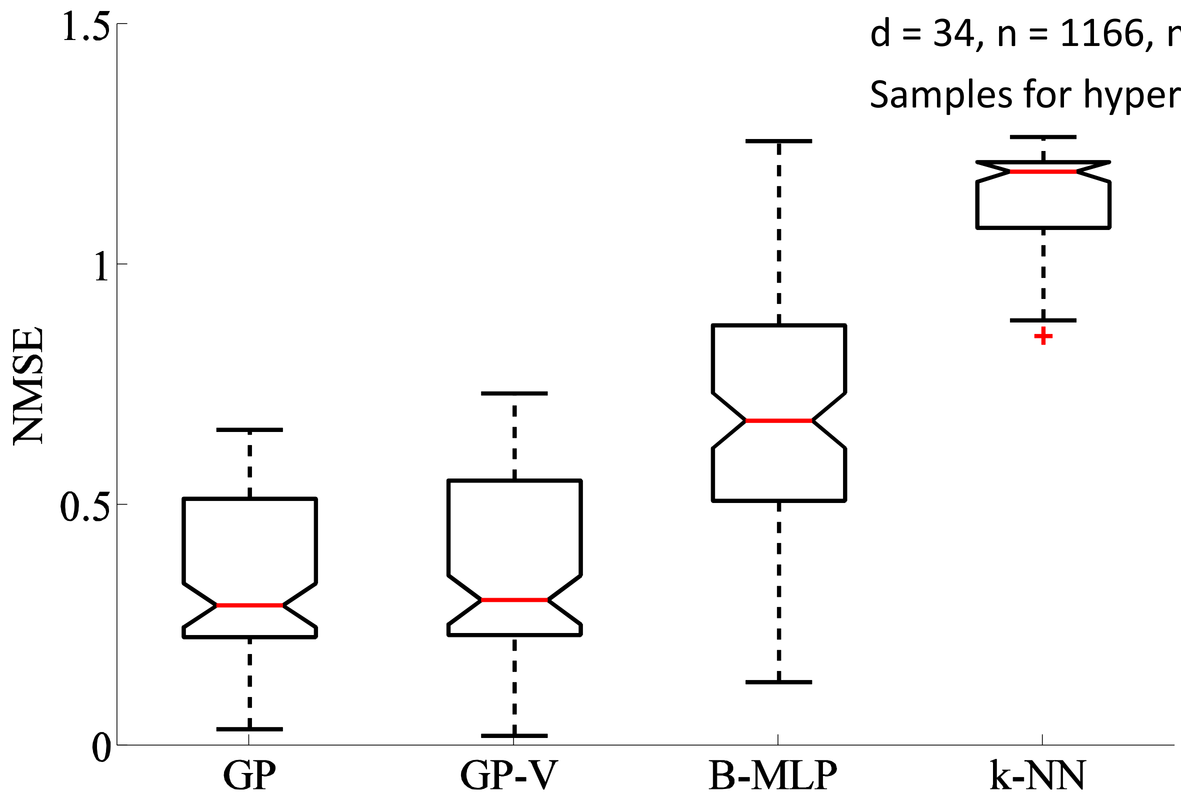
Accuracy does not degrade with approximation

Stable GP Results

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.

Conclusion

- New Gaussian Process regression algorithm for Virtual Sensors in Earth Science data.
- Have shown a method to scale from 10^2 points to 10^6 points
- 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

For more information please see: dashlink.arc.nasa.gov/member/ashok

APPENDIX

Gaussian Process Regression

- Gaussian Process regression uses Bayesian inference under additive Gaussian noise assumption to learn a function on a given data set with a confidence measure*:

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

- 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:

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

* C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Machine Learning,” MIT Press, 2006.

Low-rank Approximations

- Numerical approximation techniques exist such as Subset of Regressors, Q-R decomposition, V method
  - Numerical instability can be a problem
- ***Solution:*** ***Stable GP*** (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.