Exact Inference for Gaussian Process Regression in case of Big Data with the Cartesian Product Structure

Belyaev Mikhail[1,2,3], *Burnaev Evgeny*[1,2,3], Kapushev Yermek[1,2]

[1]Institute for Information Transmission Problems, Moscow, Russia
[2]DATADVANCE, llc, Moscow, Russia
[3]PreMoLab, MIPT, Dolgoprudny, Russia

ICML 2014 workshop on New Learning Frameworks and Models for Big Data

Beijing, 2014

Approximation task

Problem statement

- Let $y = g(x)$ be some unknown function.
- The training sample is given

$$\mathcal{D} = \{x_i, y_i\},\ g(x_i) = y_i,\ i = 1, \ldots, N.$$

- The task is to construct $\hat{f}(x)$ such that:

$$\hat{f}(x) \approx g(x).$$

- Factors:

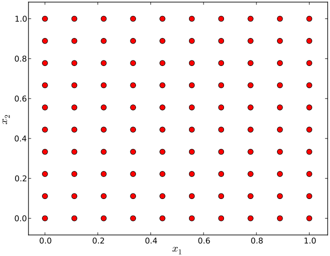
$$s^k = \{x^k_{i_k} \in \mathrm{X}^k,\ i_k = 1, \dots, n_k\},\ \mathrm{X}^k \in \mathrm{R}^{d_k},$$
$$k = 1, \dots, K;$$

d_k — dimensionality of the factor s^k.

- Factorial Design of Experiments:

$$\mathbf{S} = \{x_i\}_{i=1}^N = s^1 \times s^2 \times \cdots \times s^K.$$

- Dimensionality of $x \in \mathbf{S}$: $d = \sum_{k=1}^K d_k$.
- Sample size: $N = \prod_{k=1}^K n_k$



Example of Factorial DoE

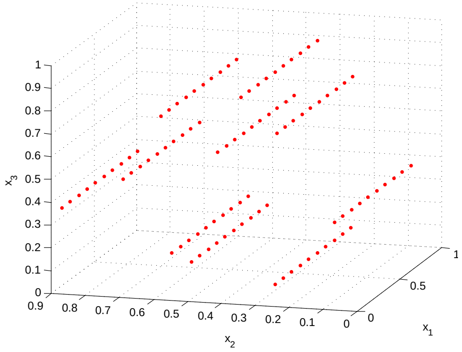


Figure : Factorial DoE with multidimensional factor

1. Independent groups of variables — factors.
2. Training sample generation procedure:
   - Fix values of the first factor.
   - Perform experiments varying values of other factors.
   - Fix other value of the first factor.
   - Perform new series of experiments.
3. Take into account knowledge from a subject domain.

Data properties

- Has special structure
- Can have large sample size
- Factors' sizes can differ significantly

Modeling of pressure distribution over the aircraft wing:

- Angle of attack: $s^1 = \{0, 0.8, 1.6, 2.4, 3.2, 4.0\}$.
- Mach number: $s^2 = \{0.77, 0.78, 0.79, 0.8, 0.81, 0.82, 0.83\}$.
- Wing points coordinates: s^3 — 5000 point in $\mathbb{R}^3$.

The training sample:

- $\mathbf{S} = s^1 \times s^2 \times s^3$.
- Dimensionality $d = 1 + 1 + 3 = 5$.
- Sample size $N = 6 * 7 * 5000 = 210000$.

- Universal techniques:
  Disadvantages: don't take into account sample structure $\Rightarrow$ low approximation quality, high computational complexity
- Multivariate Adaptive Regression Splines [Friedman, 1991]
  Disadvantages: discontinuous derivatives, non-physical behaviour
- Tensor product of splines [Stone et al., 1997, Xiao et al., 2013]
  Disadvantages: only one-dimensional factors, no accuracy evaluation procedure
- Gaussian Processes on lattice
  [Dietrich and Newsam, 1997, Stroud et al., 2014]
  Disadvantages: two-dimensional grid with equidistant points

The aim is

to develop **computationally efficient** algorithm **taking into account special features** of factorial Design of Experiments

- Function model

$$g(x) = f(x) + \varepsilon(x),$$

where $f(x)$ — Gaussian process (GP), $\varepsilon(x)$ — Gaussian white noise.

- GP is fully defined by its mean and covariance function.
- The covariance function of $f(x)$

$$K_f(x, x') = \sigma_f^2 \exp\left(-\sum_{i=1}^{d} \theta_i^2 (x^{(i)} - x'^{(i)})^2\right),$$

where $x^{(i)}$ — i-th component of vector, $\boldsymbol{\theta} = (\sigma_f^2, \theta_1, \dots, \theta_d)$ — parameters of the covariance function.

- The covariance function of $g(x)$:

$$K_g(x, x') = K(x, x') + \sigma_{noise}^2 \delta(x, x'),$$

$\delta(x, x')$ — Kronecker delta.

Maximum Likelihood Estimation

- Loglikelihood

$$\log p(\mathbf{y}\,|X,\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}_g^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_g| - \frac{N}{2}\log 2\pi,$$

where $|\mathbf{K}_g|$ — determinant of matrix $\mathbf{K}_g = \|K_g(x_i,x_j)\|_{i,j=1}^N$, $x_i, x_j \in \mathbf{S}$.

- Parameters $\boldsymbol{\theta}^*$ are chosen such that

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}}(\log p(\mathbf{y}\,|X,\boldsymbol{\theta}))$$

- Prediction of $g(x)$ at point x

$$\hat{f}(x) = \mathbf{k}^T \mathbf{K}_g^{-1} \mathbf{y},$$

where $\mathbf{k} = (K_f(x_1, x), \dots, K_f(x_n, x))$.

- Posterior variance

$$\sigma^2(x) = K_f(x, x) - \mathbf{k}^T \mathbf{K}_g^{-1} \mathbf{k}.$$

Issues:

1. High computational complexity: $\mathcal{O}(N^3)$.
   In case of factorial DoE the sample size N can be very large.

2. Degeneracy as a result of significantly different factor sizes.

Issues:

1. High computational complexity: $\mathcal{O}(N^3)$.
   In case of factorial DoE the sample size N can be very large.
2. Degeneracy as a result of significantly different factor sizes.

- Loglikelihood:

$$\log p(\mathbf{y}\,|X, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T \mathbf{K}_g^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_g| - \frac{N}{2}\log 2\pi,$$

- Derivatives:

$$\frac{\partial}{\partial\theta}\left(\log\ p(\mathbf{y}|\mathbf{X}, \sigma_f, \sigma_{noise})\right) = -\frac{1}{2}\mathrm{Tr}(\mathbf{K}_g^{-1}\mathbf{K}') + \frac{1}{2}\mathbf{y}^T\mathbf{K}_g^{-1}\mathbf{K}'\mathbf{K}_g^{-1}\mathbf{y},$$

where θ is a parameter of covariance function (component of θ_i, σ_{noise} or $\sigma_{f,i}, i = 1, \ldots, d$), and $\mathbf{K}' = \dfrac{\partial \mathbf{K}}{\partial \theta}$

Definition

Tensor $\mathcal{Y}$ is a K-dimensional matrix of size $n_1 * n_2 * \dots * n_K$:

$$\mathcal{Y} = \left\{ y_{i_1, i_2, \dots, i_K}, \; \{i_k = 1, \dots, n_k\}_{k=1}^{K} \right\}.$$

Definition

The Kronecker product of matrices A and B is a block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

- Operation vec:

$$\mathrm{vec}(\mathcal{Y}) = [\mathcal{Y}_{1,1,\ldots,1}, \mathcal{Y}_{2,1,\ldots,1}, \ldots, \mathcal{Y}_{n_1,1,\ldots,1}, \mathcal{Y}_{1,2,\ldots,1}, \ldots, \mathcal{Y}_{n_1,n_2,\ldots,n_K}].$$

- Multiplication of a tensor by a matrix along k-th direction

$$\mathcal{Z} = \mathcal{Y} \otimes_k \mathrm{B} \quad \Leftrightarrow \quad \mathcal{Z}_{i_1,\ldots,i_{k-1},j,i_{k+1},\ldots i_K} = \sum_{i_k} \mathcal{Y}_{i_1,\ldots,i_k,\ldots,i_K} \mathrm{B}_{i_k\, j}.$$

- Connection between tensors and the Kronecker product:

$$\mathrm{vec}(\mathcal{Y} \otimes_1 \mathrm{B}_1 \cdots \otimes_\mathrm{K} \mathrm{B}_\mathrm{K}) = (\mathrm{B}_1 \otimes \cdots \otimes \mathrm{B}_\mathrm{K})\mathrm{vec}(\mathcal{Y}) \qquad (1)$$

  Complexity of computation of the left part — $O(N \sum_k n_k)$, of the right part — $O(N^2)$.

Covariance function

- Form of the covariance function:

$$K_f(x, y) = \prod_{i=1}^{K} k_i(x^i, y^i), \quad x^i, y^i \in s^i,$$

  where k_i is an arbitrary covariance function for i-th factor.

- Covariance matrix:

$$\mathbf{K} = \bigotimes_{i=1}^{K} \mathbf{K}_i,$$

  $\mathbf{K}_i$ is a covariance matrix for i-th factor.

Proposition

Let $\mathbf{K}_i = \mathbf{U}_i \mathbf{D}_i \mathbf{U}_i^T$ *be a Singular Value Decomposition (SVD) of the matrix* $\mathbf{K}_i$*, where* $\mathbf{U}_i$ *is an orthogonal matrix, and* $\mathbf{D}_i$ *is diagonal. Then:*

$$|\mathbf{K}_g| = \prod_{i_1,\dots,i_K} \mathcal{D}_{i_1,\dots,i_K},$$
$$\mathbf{K}_g^{-1} = \left(\bigotimes_k \mathbf{U}_k^T\right)\left(\bigotimes_k \mathbf{D}_k + \sigma_{noise}^2 \mathbf{I}\right)^{-1}\left(\bigotimes_k \mathbf{U}_k\right), \tag{2}$$
$$\mathbf{K}_g^{-1}\mathbf{y} = \mathrm{vec}\left[((\mathcal{Y} \otimes_1 \mathbf{U}_1 \cdots \otimes_K \mathbf{U}_K) * \mathcal{D}^{-1}) \otimes_1 \mathbf{U}_1^T \cdots \otimes_K \mathbf{U}_K^T\right],$$

where $\mathcal{D}$ *is a tensor of diagonal elements of the matrix* $\sigma_{noise}^2 \mathbf{I} + \bigotimes_k \mathbf{D}_k$

Proposition

Calculation of the loglikelihood using (2) *has the following computation complexity*

$$\mathcal{O}\left(N\sum_{i=1}^{K} n_i + \sum_{i=1}^{K} n_i^3\right).$$

Assuming $n_i \ll N$ (number of factors is large and their sizes are close) we get

$$\mathcal{O}\left(N\sum n_i\right) = \mathcal{O}\left(N^{1+\frac{1}{K}}\right).$$

Proposition

The following statements hold

$$\mathrm{Tr}(\mathbf{K}_g^{-1}\mathbf{K}') = \left\langle \mathrm{diag}\left(\hat{\mathbf{D}}^{-1}\right), \bigotimes_{i=1}^{K} \mathrm{diag}\left(\mathbf{U}_i \mathbf{K}_i' \mathbf{U}_i\right) \right\rangle,$$

$$\frac{1}{2}\mathbf{y}^T\mathbf{K}_g^{-1}\mathbf{K}'\mathbf{K}_g^{-1}\mathbf{y} = \langle \mathcal{A}, \mathcal{A} \otimes_1 \mathbf{K}_1^T \otimes_2 \cdots \otimes_{i-1} \mathbf{K}_{i-1}^T \otimes_i$$
$$\otimes_i \frac{\partial \mathbf{K}_i^T}{\partial \theta} \otimes_{i+1} \mathbf{K}_{i+1}^T \otimes_{i+2} \cdots \otimes_K \mathbf{K}_K^T \Big\rangle,$$

where $\hat{\mathbf{D}} = \sigma_{noise}^2 \mathbf{I} + \bigotimes_k \mathbf{D}_k$, *and* $\mathrm{vec}(\mathcal{A}) = \mathbf{K}_g^{-1}\mathbf{y}$.
The computational complexity is

$$\mathcal{O}\left(N \sum_{i=1}^{K} n_i + \sum_{i=1}^{K} n_i^3\right).$$

Issues:

1. High computational complexity: $\mathcal{O}(N^3)$.
   In case of factorial DoE the sample size N can be very large.
2. Degeneracy as a result of significantly different factor sizes.

Issues:

1. High computational complexity: $\mathcal{O}(N^3)$.
   In case of factorial DoE the sample size N can be very large.

2. Degeneracy as a result of significantly different factor sizes.

Example of degeneracy

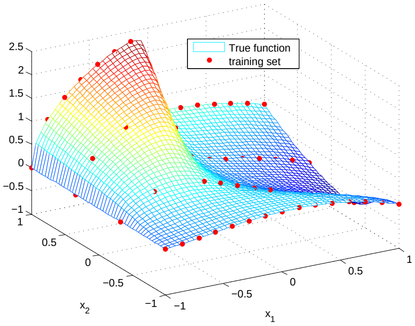


Figure : Original function

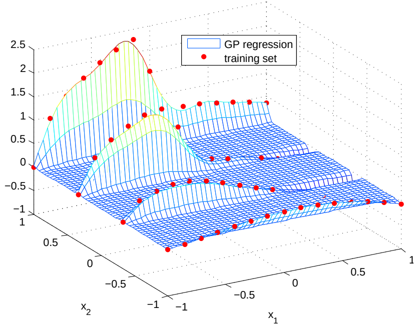


Figure : Approximation obtained using GP from GPML toolbox

- Prior distribution:

$$\frac{\theta_k^{(i)} - a_k^{(i)}}{b_k^{(i)} - a_k^{(i)}} \sim \mathcal{B}e(\alpha, \beta),\ \{i = 1, \dots, d_k\}_{k=1}^{K},$$

$$a_k^{(i)} = \frac{c_k}{\max\limits_{x,y \in s^k} \left(x^{(i)} - y^{(i)}\right)}, \quad b_k^{(i)} = \frac{C_k}{\min\limits_{x,y \in s^k, x \neq y} \left(x^{(i)} - y^{(i)}\right)}$$

where $\mathcal{B}e(\alpha, \beta)$ is the Beta distribution, c_k and C_k are parameters of the algorithm (we use $c_k = 0.01$ and $C_k = 2$).

- Initialization

$$\theta_k^{(i)} = \left[\frac{1}{n_k}\left(\max_{x \in s^k}(x^{(i)}) - \min_{x \in s^k}(x^{(i)})\right)\right]^{-1}.$$

Regularization

Loglikelihood:

$$\log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}, \sigma_f, \sigma_{noise}) = -\frac{1}{2}\mathbf{y}^T \mathbf{K}_y^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K}_y| - \frac{N}{2} \log 2\pi$$

$$+ \sum_{k,i} \left((\alpha - 1) \log \left(\frac{\theta_k^{(i)} - a_k^{(i)}}{b_k^{(i)} - a_k^{(i)}} \right) + (\beta - 1) \log \left(1 - \frac{\theta_k^{(i)} - a_k^{(i)}}{b_k^{(i)} - a_k^{(i)}} \right) \right) -$$

$$-d \log(\mathrm{B}(\alpha, \beta)).$$

Loglikelihood:

$$\log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}, \sigma_f, \sigma_{noise}) = -\frac{1}{2}\mathbf{y}^T \mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{N}{2}\log 2\pi$$

$$+\sum_{k,i}\left((\alpha-1)\log\left(\frac{\theta_k^{(i)} - a_k^{(i)}}{b_k^{(i)} - a_k^{(i)}}\right) + (\beta-1)\log\left(1 - \frac{\theta_k^{(i)} - a_k^{(i)}}{b_k^{(i)} - a_k^{(i)}}\right)\right) -$$

$$-d\log(\mathrm{B}(\alpha,\beta)).$$

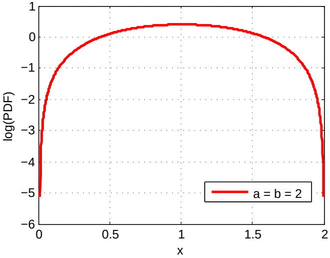


Figure : Penalty function

Example of regularized approximation

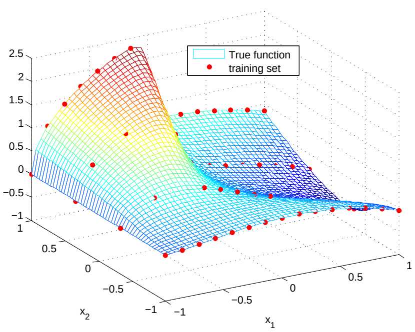


Figure : Original function

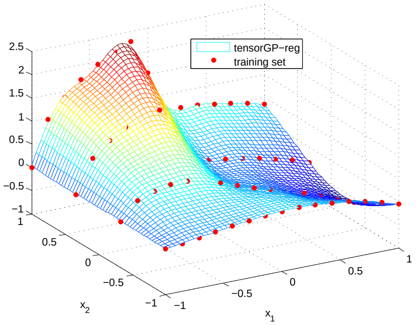


Figure : Approximation obtained using developed algorithm with regularization

- Set of 34 functions (usual artificial test functions used for testing of global optimization algorithms)
- Dimensionality — from 2 to 6.
- Sample sizes from 80 to 216000 with full factorial DoE
- Quality criteria — training time and approximation error:

$$\mathrm{MSE} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (\hat{f}(\mathbf{x}_i) - g(\mathbf{x}_i))^2$$

- Tested algorithms:
  - MARS — Multivariate Adaptive Regression Splines
  - SparseGP — sparse Gaussian Processes (GPML toolbox)
  - tensorGP — developed algorithm without regularization
  - tensorGP-reg — developed algorithm with regularization

- T problems, A algorithms.
- e_{ta} — approximation error (or training time) of a-th algorithm on t-th problem.
- $\tilde{e}_t = \min_a e_{ta}$.

$$\rho_a(\tau) = \frac{\#\{t : e_{ta} < \tau \tilde{e}_t\}}{T}$$

- The higher the curve is the better works the corresponding algorithm.
- $\rho_a(1)$ — fraction of problems for which a-th algorithm worked the best.

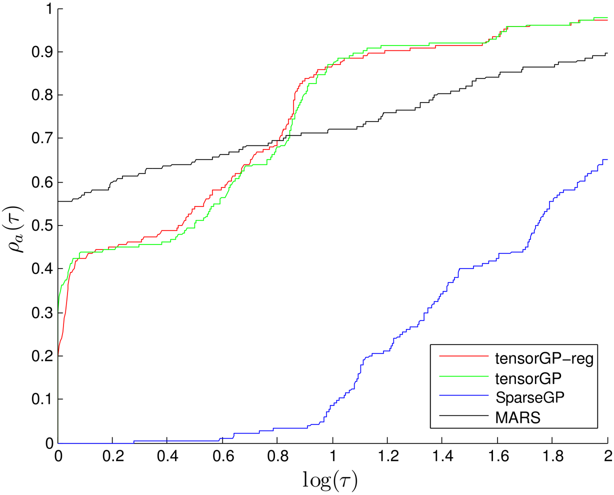


Figure : Dolan-Moré curves. Quality criterion — training time

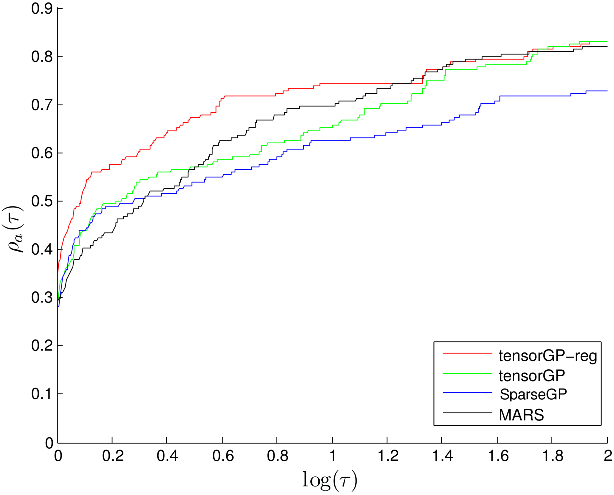


Figure : Dolan-Moré curves. Quality criterion — MSE

Objective functions:

- p_1 — contact pressure.
- $S_{r_{max}}$ — maximum radial stress.
- w — weight of disc.

The geometrical shape of the disc is parametrized by 6 input variables $x = (h_1, h_2, h_3, h_4, r_2, r_3)$ (r_1 and r_4 are fixed)

Training sample (generated from computational physical model):

- Sample size — 14400
- Factor sizes — $\{1, 8, 8, 3, 15, 5\}$

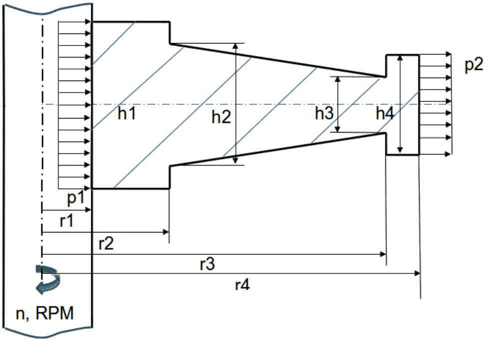


Figure : Rotating disc of an impeller

Results of experiment

Table : Approximation errors of p_1

	MAE	MSE	RRMS
MARS	4.4644	6.5120	0.1166
SparseGP	76.9313	86.7034	1.5530
tensorGP-reg	0.3020	0.3981	0.0070

$$\mathrm{MAE} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |\hat{f}(\mathbf{x}_i) - g(\mathbf{x}_i)|,$$

$$\mathrm{MSE} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (\hat{f}(\mathbf{x}_i) - g(\mathbf{x}_i))^2$$

$$\mathrm{RRMS} = \sqrt{\frac{\sum_{i=1}^{N_{test}} (\hat{f}(\mathbf{x}_i) - g(\mathbf{x}_i))^2}{\sum_{i=1}^{N_{test}} (\bar{y} - g(\mathbf{x}_i))^2}}, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i.$$

2D-slices of approximations (other parameters are fixed)

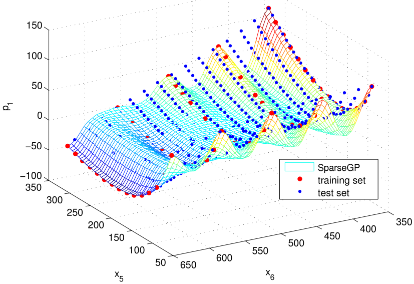


Figure : GPML Sparse GP is applied

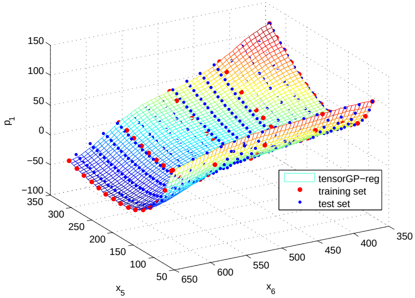


Figure : Approximation obtained using developed algorithm with regularization

Further research: missing data

Reasons for missing points:

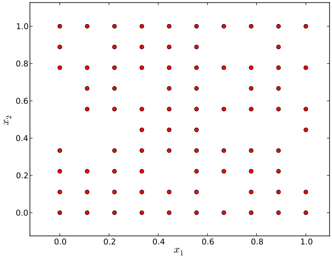
- Data generation is in progress (each point calculation is time consuming).
- Data generator failed in some points.

Incomplete Factorial DoE

- $\mathbf{S}_{full}$ — full factorial DoE.
  $N_{full} = |\mathbf{S}_{full}|$.
- $\mathbf{S}$ — incomplete factorial DoE.
  $N = |\mathbf{S}|$.
- $\mathbf{S} \subset \mathbf{S}_{full}$

$\Rightarrow$ Covariance matrix is not the Kronecker product!

$$\mathbf{K}_f \neq \bigotimes_{i=1}^{K} \mathbf{K}_i .$$



Notation

- $\{x_i\}_{i=1}^{N_{full}} = \mathbf{S}_{full}$.
- $\widetilde{\mathbf{K}}_f$ — covariance matrix for full factorial DoE $\mathbf{S}_{full}$.
- $\mathbf{W}$ — diagonal matrix such that $\mathbf{W}_{ii} = \begin{cases} 1, & \text{if } x_i \in \mathbf{S} \\ 0, & \text{if } x_i \notin \mathbf{S}. \end{cases}$
- $\tilde{\mathbf{y}}$ — vector of outputs $\mathbf{y}$ extended by arbitrary values.

Proposition

Let $\tilde{\mathbf{z}}^*$ be a solution of

$$(\widetilde{\mathbf{K}}_f \mathbf{W} \widetilde{\mathbf{K}}_f + \sigma_{noise}^2 \widetilde{\mathbf{K}}_f)\tilde{\mathbf{z}} = \widetilde{\mathbf{K}}_f \mathbf{W} \tilde{\mathbf{y}}. \tag{3}$$

Then the solution $\mathbf{z}^*$ of $(\mathbf{K}_f + \sigma_{noise}^2 \mathbf{I})\mathbf{z} = \mathbf{y}$, i.e.

$$\mathbf{z}^* = (\mathbf{K}_f + \sigma_{noise}^2 \mathbf{I})^{-1} \mathbf{y},$$

has the form $\mathbf{z}^* = (\tilde{\mathbf{z}}_{i_1}^*, \ldots, \tilde{\mathbf{z}}_{i_N}^*)$, where $i_k \in \{j : x_j \in \mathbf{S}\}, k = 1, \ldots, N$.

Computation of loglikelihood

- $R = N_{full} - N$ — number of missing points.
- $\widetilde{\mathbf{U}} = \bigotimes_{i=1}^{K} \widetilde{\mathbf{U}}_i$
- $\widetilde{\mathbf{D}} = \bigotimes_{i=1}^{K} \widetilde{\mathbf{D}}_i$.
- $\hat{\widetilde{\mathbf{D}}} = \widetilde{\mathbf{D}} + \sigma_{noise}^2 \mathbf{I}$.
- The change of variables

$$\tilde{\boldsymbol{\alpha}} = \begin{cases} \left(\hat{\widetilde{\mathbf{D}}}\widetilde{\mathbf{D}}\right)^{\frac{1}{2}} \widetilde{\mathbf{U}}^T \tilde{\mathbf{z}} & \text{if } R < N, \\ \left(\sigma_{noise}^2 \widetilde{\mathbf{D}}\right)^{\frac{1}{2}} \widetilde{\mathbf{U}}^T \tilde{\mathbf{z}} & \text{if } R \geq N. \end{cases} \tag{4}$$

Proposition

System of linear equations (3) can be solved using the change of variables (4) and Conjugate Gradient Method in at most $\min(R+1, N+1)$ iterations. The computational complexity of each iteration is $\mathcal{O}(N_{full} \sum_k n_k)$.

- $\widetilde{\mathbf{K}}_f + \sigma_{noise}^2 \mathbf{I} = \begin{pmatrix} \mathbf{K}_g & \mathbf{A} \\ \mathbf{A}^T & \mathbf{B} \end{pmatrix}$
- Determinant

$$|\mathbf{K}_g| = \frac{|\widetilde{\mathbf{K}}_f + \sigma_{noise}^2 \mathbf{I}|}{|\mathbf{B} - \mathbf{A}^T \mathbf{K}_g^{-1} \mathbf{A}|}. \tag{5}$$

  - $|\widetilde{\mathbf{K}}_f + \sigma_{noise}^2 \mathbf{I}|$ is computed using formulae for full factorial case.
  - $|\mathbf{B} - \mathbf{A}^T \mathbf{K}_g^{-1} \mathbf{A}|$ is computed numerically.

Proposition

The complexity of computing determinant using (5) *is* $\mathcal{O}\left(\min\{R+1, N+1\} R N_{full} \sum_k n_k\right)$.

Conclusion

The developed algorithm

- is computationally efficient;
- can handle large samples;
- takes into account features of given data;
- is proved to be efficient on a large set of toy problems as well as real world problems.

Thank you for attention!

More details are given in

- Belyaev, M., Burnaev, E., and Kapushev, Y. (2014).
  Exact inference for gaussian process regression in case of big data with the cartesian product structure.
  arXiv preprint arXiv:1403.6573

Dietrich, C. R. and Newsam, G. N. (1997).
Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix.
SIAM J. Sci. Comput., 18(4):1088–1107.

Friedman, J. (1991).
Multivariate adaptive regression splines.
Annals of Statistic, 19(1):1–141.

Stone, C., Hansen, M., Kooperberg, C., and Truong, Y. (1997).
Polynomial splines, their tensor products in extended linear modeling.
Annals of Statistic, 25:1371–1470.

Stroud, J. R., Stein, M. L., and Lysen, S. (2014).
Bayesian and maximum likelihood estimation for gaussian processes on an incomplete lattice.
arXiv preprint arXiv:1402.4281.

Xiao, L., Li, Y., and Ruppert, D. (2013).
Fast bivariate p-splines: the sandwich smoother.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 75(3):577–599.