Operational Learning of Approximate Analytical Methods for Performance Evaluation of Manufacturing Systems

Andrea Matta [1], Li Na [1], Ziwei Lin [1], George J. Shanthikumar [2]

[1]Shanghai Jiao Tong University, Department of Industrial Engineering and Management

[2]Purdue University, Krannert School of Management

Outline

Motivation

- Analytical methods and simulation are frequently used alternatively and not concurrently. Can we save time if we use both at the same time ?
- A large computational time is spent in simulation to validate analytical methods and is not used anymore. Why do we not keep these data to improve the developed analytical method ?
- Which properties an analytical method should have to be used in couple with simulation ?
- Can we continuously improve the estimation of system performance ?

Analytical methods

- Analytical methods are frequently used to estimate the perfomance of production systems with stochastic behaviour.
- Approximations can be introduced to make analytical methods solvable for practical applications.
- Queueing theory and Markov chains are the most utilized frameworks under which analytical methods are developed.
- In general, the larger the system complexity the larger the approximation introduced in the analytical method.
- Approximate analytical methods are fast in providing system performance estimates. Nevertheless, this advantage is counterbalanced by the bias of the provided estimates.
- A large attention is usually given to evaluate the expected bias of the analytical method under a set of numerical experiments, however little efforts are put on understanding the regularity of the bias.

Simulation

- Complements analytical methods, as it is used when an accurate estimate of the system performance is needed.
- Very flexible as it is scalable to any desired detail level.
- Computational extensive.
- Provides not exact results because of the finiteness of simulation runs.
- Also used to quantitatively assess the bias of newly developed analytical methods.

Research objective

Given a system we want to study and

- A simulation model for high accuracy (fidelity) estimation of system performance
- An analytical model for low accuracy (fidelity) estimation of system performance
- High-fidelity model is time consuming to run compared with low-fidelity model.

Objective

To develop a meta–model (or surrogate model) that uses both the simulation and the analytical model output for estimating the system performance.

This problem is known in literature as multi-fidelity regression modeling.

Related literature

- Current research contributions extend parametric models to consider outputs from different sources, which are mainly simulation models with different detail levels. Some references:
  - Co-Kriging merges data from different fidelity sources by extending Kriging estimator. (*Forrester A.I.J. and Keane A.J. 2009. Recent Advances in Surrogate-Based Optimization, Progress in Aerospace Sciences*).
  - A Bayesian approach to combine results from different simulators. (*Goh J., Bingham D., Holloway J.P., Grosskpf M.J., Kuranz C.C., Rutter E. 2013. Prediction and Computer Model Calibration Using Outputs from Multi-fidelity Simulators, Technometrics*).
- A parallel research uses locally weighted regression and smoothing method (LOESS) to combine simulation results with those provided by a Jackson queuing network. (*Chen R., Xu J., Chen C.H., Lee L.H. 2015. An Effective Learning Procedure for Multi-fidelity Simulation Optimization with Ordinal Transformation, submitted to CASE2015 Conference*).

Assumptions and notation

- $Y(\mathbf{x})$ is a univariate random variable, $\mathbf{x} \in \mathcal{R} \subset \mathbb{R}^d$ is a $1 \times d$ vector and $\mathcal{R}$ is the set containing all the admissible system configurations.
- We assume someone else has already defined a DOE and executed experiments for us. The DOE consists of
  - n design points in the set $\mathcal{N} = \{1 \dots n\}$
  - each design point $i \in \mathcal{N}$ represents the system configuration $\mathbf{x_i^0} = [x_{i1}^0, \dots, x_{ik}^0, \dots x_{id}^0] \in \mathcal{R}$, where x_{ik}^0 is the system variable k at design point i.
  - y_i^{0s} is the output of simulation experiment at design point i.
  - y_i^{0a} is the output of analytical method at design point i.
- We assume that simulation experiments are infinitely accurate and not affected by noise, i.e. they provide *oracle* outputs.
- We assume that analytical methods are not exact in all the space $\mathbf{x} \in \mathcal{R}$, thus deviations from simulation oracles may exist.

We want to estimate:

$$y(\mathbf{x}) = \mathbb{E}(Y | \mathbf{y^0}(\mathbf{x^0}), y^a(\mathbf{x})),$$

where $y^a(\mathbf{x})$ is the response of analytical method at point $\mathbf{x}$.

Kernel regression: estimator

A pure Kernel estimator can be calculated using the Nadaraya–Watson estimator:

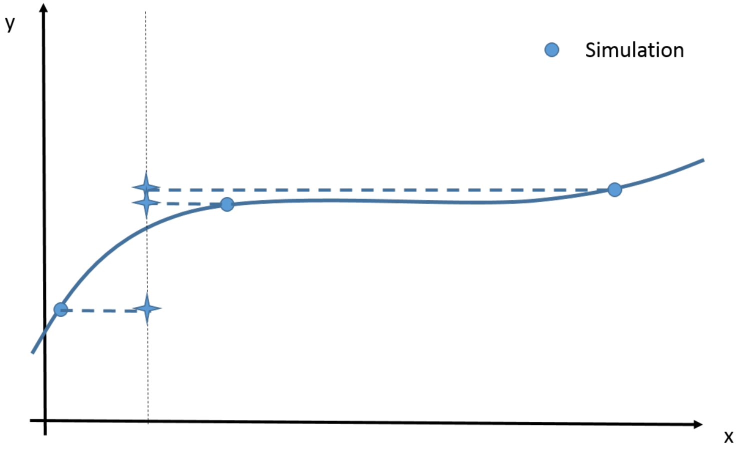
$$\widehat{y}_K(\mathbf{x}) = \frac{\sum_{i\in\mathcal{N}} K\left(\mathbf{x_i^0} - \mathbf{x}\right) y_i^{0s}(\mathbf{x}_i^0)}{\sum_{i\in\mathcal{N}} K\left(\mathbf{x_i^0} - \mathbf{x}\right)}, \tag{1}$$

where $K(.)$ is a d–dimensional kernel function. The kernel function, $K(\mathbf{z_i})$, is chosen as the widely applied Gaussian Kernel:

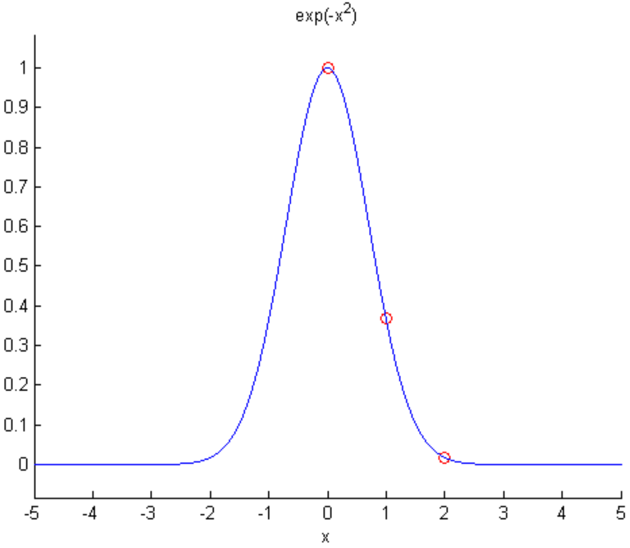
$$K(\mathbf{z_i}) = \prod_{k\in\mathcal{K}} e^{-(1/2\theta_k)z_{ik}^2}, \tag{2}$$

where θ_k represents a correlation coefficient and $z_{ik} = (x_{ik}^0 - x_k)$.

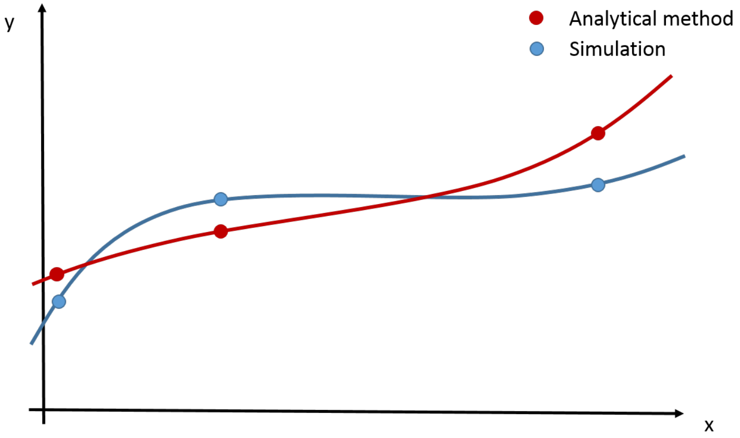
Kernel regression



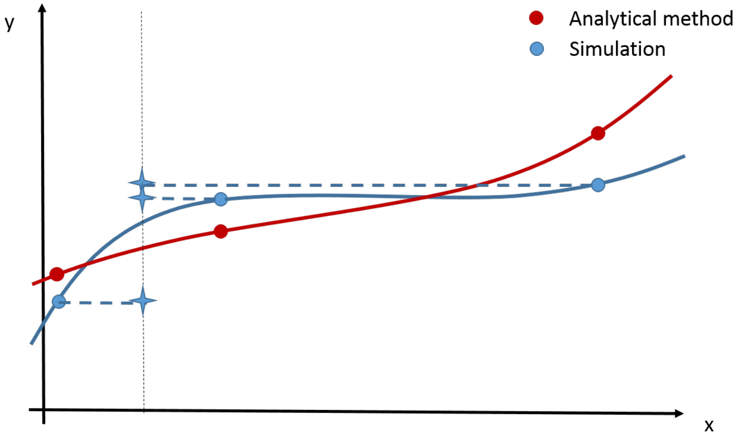
Gaussian Kernel function



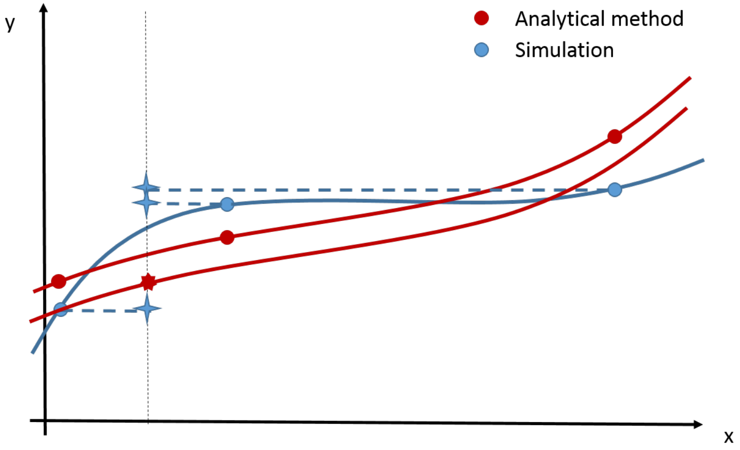
Extending Kernel with a structure



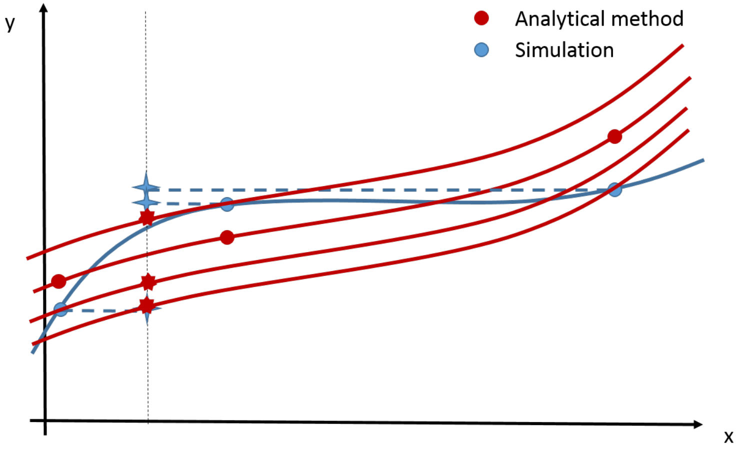
Extending Kernel with a structure



Extending Kernel with a structure



Extending Kernel with a structure



Extending Kernel with a structure

Predictions of high-fidelity observations are extended with analytical method estimates by combining them through the use of kernel function K_2:

$$\hat{y}_i(\mathbf{x}) = K_2\left(\mathbf{x_i^0} - \mathbf{x}\right) y_i^{0s} + \left[1 - K_2\left(\mathbf{x_i^0} - \mathbf{x}\right)\right] \tilde{y}_i^{aj}(\mathbf{x}) \quad \forall i, \tag{3}$$

where $\tilde{y}_i^a(\mathbf{x})$ is the response of analytical method at point $\mathbf{x}$ adjusted with the bias evaluated at $\mathbf{x_i^0}$:

$$\tilde{y}_i^a(\mathbf{x}) = y^a(\mathbf{x}) + y_i^{0s}(\mathbf{x_i^0}) - y_i^{0a}(\mathbf{x_i^0}), \qquad \forall i. \tag{4}$$

$$\widehat{y}_{EK}(\mathbf{x}) = \frac{\sum_{i\in\mathcal{N}} K_1\left(\mathbf{x_i^0} - \mathbf{x}\right) \hat{y}_i(\mathbf{x})}{\sum_{i\in\mathcal{N}} K_1\left(\mathbf{x_i^0} - \mathbf{x}\right)}, \tag{5}$$

where $K_1(.)$ is a d–dimensional kernel function.

Algorithm

1. **Initialization**:
   - $d \leftarrow$ *number of system configuration variables*
   - $n \leftarrow$ *number of samples in the initial design*
   - $\mathbf{x}^{\mathbf{0}} \leftarrow$ *DOE*
   - $\mathbf{x} \leftarrow$ *System configuration to evaluate*
   - θ_{1k}, θ_{2k} for $k \in \mathcal{K} \leftarrow$ *Assigned values*
2. **DOE**, for $i = 1, \ldots, n$:
   - <Execute simulation at $\mathbf{x_i^0}$ and collect y_i^{0s}>
   - <Execute analytical model at $\mathbf{x_i^0}$ and collect y_i^{0a}>
3. <Execute analytical model at $\mathbf{x}$ and collect y^a>
4. for $i = 1, \ldots, n$:
   - <Calculate $\tilde{y}_i^a(\mathbf{x})$ with equation (4)>
   - <Calculate $K_2\left(\mathbf{x_i^0} - \mathbf{x}\right) = \prod_{k \in \mathcal{K}} e^{-(1/2\theta_{2k})\left(x_{ik}^0 - x_k\right)^2}$>
   - <Calculate $\hat{y}_i(\mathbf{x})$ using equation (3)>
   - <Calculate $K_1\left(\mathbf{x_i^0} - \mathbf{x}\right) = \prod_{k \in \mathcal{K}} e^{-(1/2\theta_{1k})\left(x_{ik}^0 - x_k\right)^2}$ using equation (2)>
5. <Calculate $\widehat{y}_{EK}(\mathbf{x})$ using equation (5)>

Case 1: flow line with age dependent repair probabilities

- Flow line with K machines and $K-1$ finite capacity buffers
- Deterministic processing times
- First machine is never starved and last machine is never blocked
- Unreliable machines failing in one single mode. Operation dependent failures
- Geometrically distributed failure times
- Age dependent repair times:

$$r_i = r_{i0} + age\frac{(1 - r_{i0})}{a}, \tag{6}$$

where r_{i0} is the nominal repair probability of machine i, age is the actual duration of the failure and a is a parameter.

Case 1: flow line with age dependent repair probabilities

- $K = 8$ machines
- 10 different DOEs each one with 10 design points are randomly sample using latin hypercube sampling (LHS)
- Range parameters were defined using guidelines in *Gershwin S.B. (2011), Case Generation for Algorithm Testing, SMMSO 2011*
  - $p_i \in [0, 0.06], \forall i = 1, \ldots, K$
  - $r_i \in [0.05, 0.2], \forall i = 1, \ldots, K$
  - $N_i \in [4, 20], \forall i = 1, \ldots, K$
- $a = 1$, *age* is calculated during simulation
- Mean values of production rates and buffer levels estimated with a single simulation run with length 1,000,000 amd warm-up 100,000 time units
- For each DOE, the θ values are selected on the basis of 10 design points (10 replications) within the set $\{0.1, 1, 10, 20, 100\}$
- The mean absolute relative error (MARE) is used to quantify the accuracy on $R = 1000$ checkpoints randomly generated with LHS:

$$MARE = \frac{1}{R} \sum_{i=1}^{R} \frac{|\widehat{y}_{EK}(\mathbf{x_i}) - y^s(\mathbf{x_i})|}{y^s(\mathbf{x_i})} \quad (7)$$

Results: average MARE values

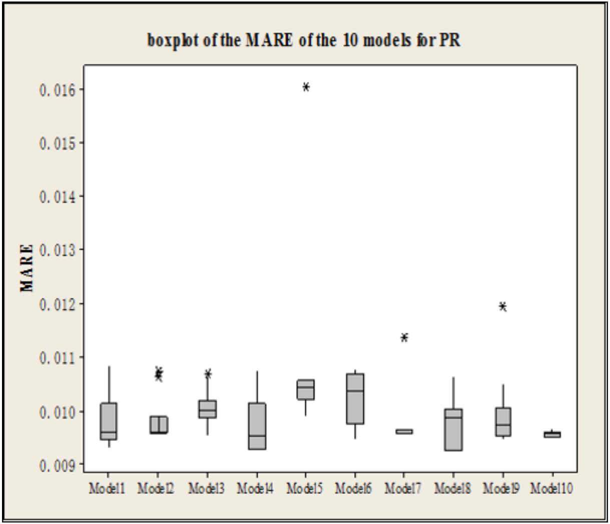
PR	DOE 1	DOE 2	DOE 3	DOE 4	DOE 5	DOE 6	DOE 7	DOE 8	DOE 9	DOE 10
EK	0.00986	0.00983	0.01005	0.00976	0.0109	0.01026	0.00979	0.0098	0.00999	0.00959
Pure Kernel	0.01064	0.01037	0.01154	0.01054	0.0122	0.01062	0.01037	0.0103	0.01059	0.01015
DDX	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119

Results: average MARE values

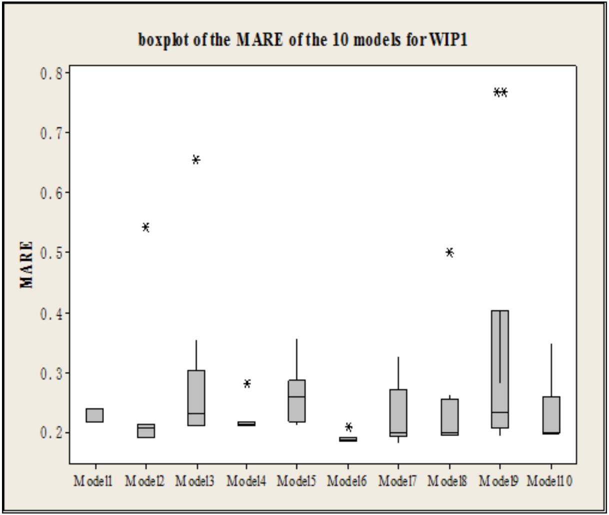
PR	DOE 1	DOE 2	DOE 3	DOE 4	DOE 5	DOE 6	DOE 7	DOE 8	DOE 9	DOE 10
EK	0.00986	0.00983	0.01005	0.00976	0.0109	0.01026	0.00979	0.0098	0.00999	0.00959
Pure Kernel	0.01064	0.01037	0.01154	0.01054	0.0122	0.01062	0.01037	0.0103	0.01059	0.01015
DDX	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119	0.4119

WIP1	DOE 1	DOE 2	DOE 3	DOE 4	DOE 5	DOE 6	DOE 7	DOE 8	DOE 9	DOE 10
EK	0.22718	0.23638	0.28809	0.22106	0.25723	0.19007	0.22688	0.24048	0.3355	0.23274
Pure Kernel	0.62311	0.63891	0.75149	0.68228	0.75716	0.74872	0.72764	0.6056	0.6945	0.72203
DDX	0.21656	0.21656	0.21656	0.21656	0.21656	0.21656	0.21656	0.21656	0.21656	0.21656

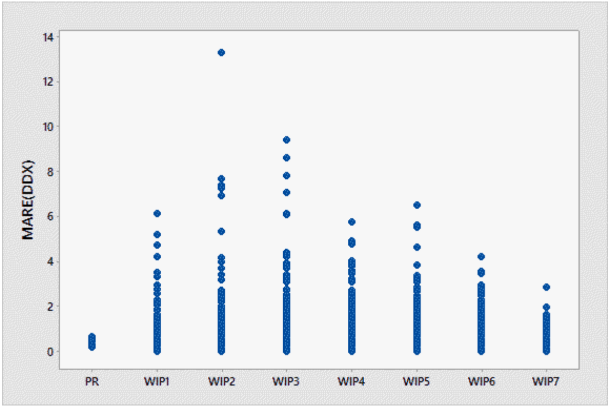
Results: production rate



Results: buffer level 1



Deviations of DDX



Density matters

- One DOE with 10 points
- Design points in $p_i \in [0, 0.9], r_i \in [0.05, 1], N_i \in [4, 20]$
- Checkpoints in $p_i \in [0, 0.06], r_i \in [0.05, 0.2], N_i \in [4, 20]$

	PR	**WP1**	**WP2**	**WP3**	**WP4**	**WP5**	**WP6**	**WP7**
EK	0.199	0.228	0.36	0.48	0.515	0.56	0.382	0.385
Pure Kernel	0.492	0.921	1.02	1.472	1.197	1.01	1.317	0.588
DDX	0.412	0.217	0.33	0.406	0.39	0.41	0.381	0.312

Some comments

- Smoothness of the error committed by analytical method matters
- Good results with few points
- Pure Kernel fails when we estimate out of the DOE domain.
- Similar results if we assume correlated failure probabilities (Case 2)
- As the number of design points increases the estimate should converge to the true estimate

Conclusions and outlook

- This paper has proposed a method to combine results from simulation and analytical method to fit meta-models for manufacturing system performance
- The method is not based on specific assumptions
- The development of new approximate analytical methods should consider more the variability of the error
- Similar results if we assume correlated failure probabilities (Case 2)
- Extension to n analytical methods
- Extension to a family of analytical methods and operationally estimating parameters
- Dynamic sampling
- Stopping policy to guarantee a maximum error
- Extension to optimization

THANK YOU

matta@sjtu.edu.cn


SJTU