A Benchmark of Contemporary Metamodeling Algorithms

Can Bogoclu, Dirk Roos

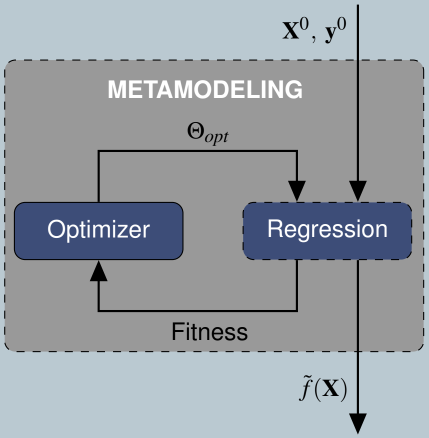
Computer Simulation and Design Optimization
IMH
Institute of Modeling and High-Performance Computing

Hochschule Niederrhein
University of Applied Sciences

IMH
Institut für Modellbildung und Hochleistungsrechnen
Institute of Modelling and High-Performance Computing

Inhalt

Overview of Algorithms

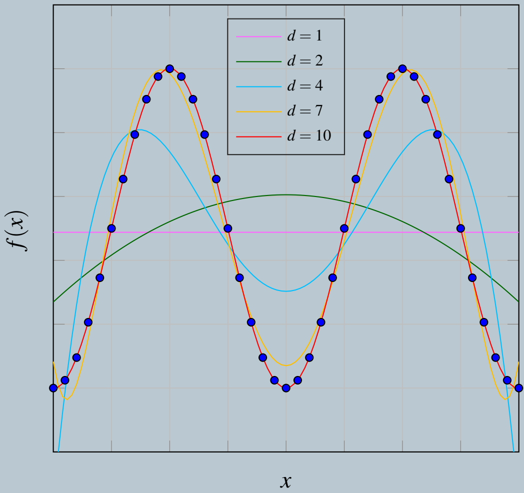


- (p) Ordinary Least Squares
- (p) An/Isotropic Moving Least Squares
- (p) An/Isotropic Kriging
- (np) Radial Basis Functions
- (np) Support Vector Regression
- (np) Artificial Neural Networks
- (p) Gaussian Process [1]
- (p) Ordinary Kriging [2]
- (p) Interpolating MLS [2]

[1] ASCMO
[2] optiSlang

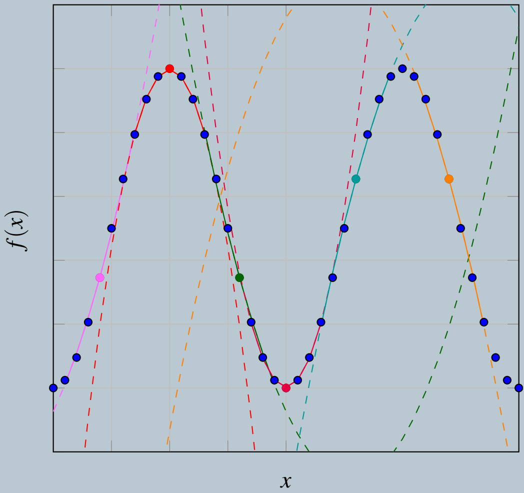
Ordinary Least Squares

- $\tilde{f}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \cdot \mathbf{c}$
- Determine coefficient vector $\mathbf{c}$, that minimizes the sum of squared errors (linear solution)
- $\mathrm{SSE} = \sum_{i=0}^{m} (y_i^0) - \tilde{f}(\mathbf{X}_i^0))^2$
- No free variables. Only the polynomial degree can be varied
- Number of coefficients for n input dimensions and polynomial order of d: $\binom{n+d}{d}$.
  $n = 2,\ d = 10 \rightarrow 264$
  $n = 3,\ d = 5 \rightarrow 2016$



Moving Least Squares

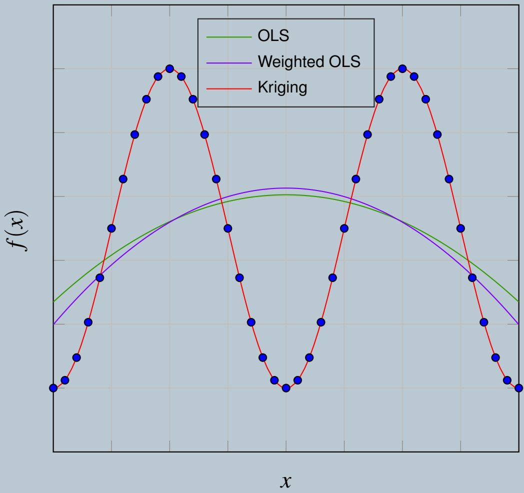
- $\tilde{f}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \cdot \mathbf{c}(\mathbf{x})$
- Local coefficients $\mathbf{c}(\mathbf{x})$ instead of global $\mathbf{c}$ for every approximated point $\mathbf{x}$
- Weights $0 < q_i(\mathbf{x}, \mathbf{x}_i^0) < 1$ near influence domain defined by radius θ_r and $q_i(\mathbf{x}, \mathbf{x}_i^0) = 0$ outside
- $\mathbf{c}(\mathbf{x})$ determined using a weighted least squares (linear problem)
- θ_r determined through nonlinear optimization
- (Isotropic) MLS: $\theta_r \in \mathbb{R}$
  Anisotropic MLS [3] $\theta_r \in \mathbb{R}^n$



[3] (Cremanns 2014)

Kriging (Gaussian Process) [1] [2] [4]

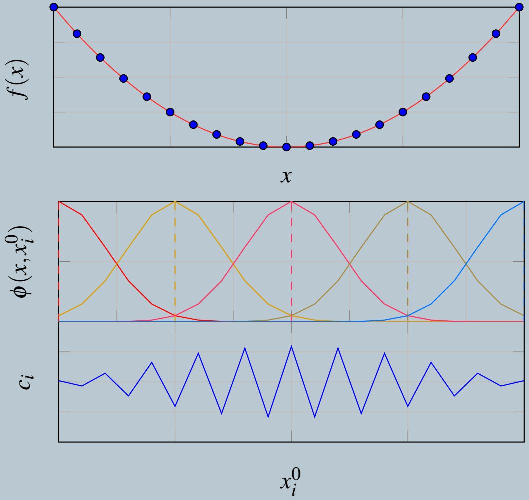
- $\tilde{f}(\mathbf{x}) = \mathbf{P}(\mathbf{x})\hat{\mathbf{c}} + \mathbf{q}(\mathbf{x})\mathbf{Q}(\mathbf{X}^0)^{-1}(\mathbf{y}^0 - \mathbf{P}_0^T\hat{\mathbf{c}})$
- Spatial correlation as weighting function $\mathbf{q}(\mathbf{x}, \mathbf{X}^0, \theta)$ and correlation matrix $\mathbf{Q}(\mathbf{X}^0, \mathbf{X}^0, \theta)$
- Determine θ by optimizing maximum likelihood function (Gaussian Process) or SSE
- (Isotropic) Kriging: $\theta \in \mathbb{R}$ Anisotropic Kriging $\theta \in \mathbb{R}^n$



[4] DACE Toolbox, python/sk-learn

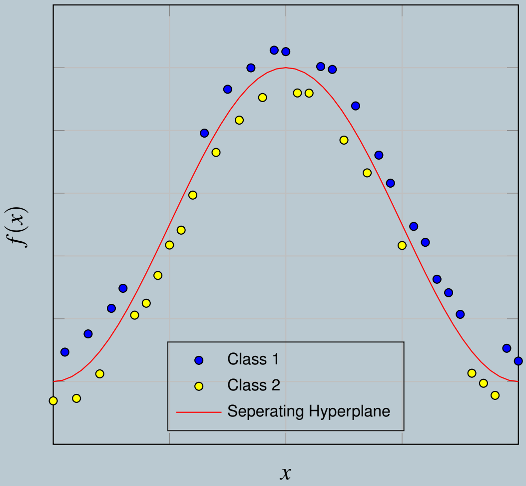
Radial Basis Functions

- $\tilde{f}(\mathbf{x}) = \sum_{i=1}^{m} c_i \phi\left(\mathbf{x}, \mathbf{x}_i^0\right)$
- RBF ϕ: Functions of distance between the support points and the prediction point
- c_i determined using the least squares (linear problem), free parameter $\hat{\theta}$ of RBF through optimization
- Tests with Gaussian, Multiquadratic and Cubic RBF



Support Vector Regression [5]

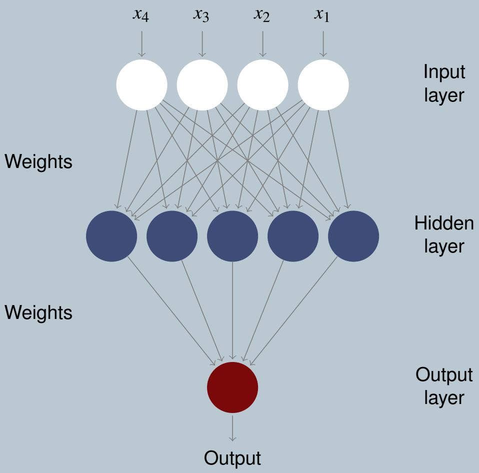
- $\tilde{f}(\mathbf{x}) = \sum_{i=r}^{m} \mathbf{c}_i \mathbf{Q}(\mathbf{x}, \mathbf{x}_i^0) + b$
- Goal: Find the optimal separating hyperplane which maximizes the margin of the training data (Linear non-parametric algorithm)
- Nonlinear transformation of input data $\phi(\mathbf{x})$ into a higher dimensional Hilbert space for linearization
- Kernel trick allows inexplicit definition of $\phi(\mathbf{x})$: $\mathbf{Q}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$



[5] libsvm, python/sk-learn

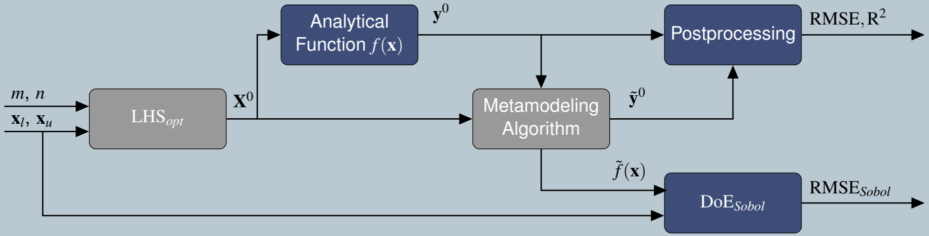
Artificial Neural Networks [6]

- Output layer: $\tilde{f}(\mathbf{x}) = \sum_{j=1}^{l_2} c_j \, \phi_j$
- Input and hidden layers: $\phi(\mathbf{x}) = q\left(\sum_{i=1}^{l_1} b_i \, w_i(x_i, \hat{\theta})\right)$
- Weights depend on the distance between the input point and the center point of a neuron
- Linear weights $\mathbf{b}$, λ and central vectors $\mathbf{w}_{0,i}$ are found in a two step optimization (least squares)
- Smoothness parameter $\hat{\theta}$ found through optimization



[6] MATLAB Neural Network Toolbox

Overview Of Testing Process



m - Number of observations ($32, 64, 96, 128$ sampling points)

n - Number of dimensions (features)

$\mathbf{x}_l,\ \mathbf{x}_u$ - Lower and upper bound

RMSE - Root mean squared error $\mathrm{RMSE} = \sqrt{\mathrm{SSE}/m}$

R^2 - Generalized $\mathrm{R}^2 = 1 - (\sum(y_i^0 - \tilde{y}_i^0)^2)/(\sum(y_i^0 - E(y_i^0))^2)$

RMSE_{Sobol} - Root mean squared error of 16384 Sobol points

Inhalt

Functions With High Modality

Analytical functions:

- ✓ Ackley's function
- ✓ Cross-in-tray function
- ✓ Eggholder function
- ✓ Hölder table function
- ✓ Lévy function No.13

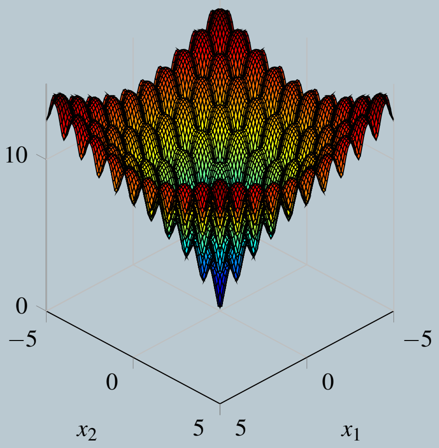
- Functions with many local extrema
- Complicated structure, hard to reproduce
- Algorithm performance in absence of a suitable solution
- Ability to handle failure (Simplification of an overly complex problem)

Aims:

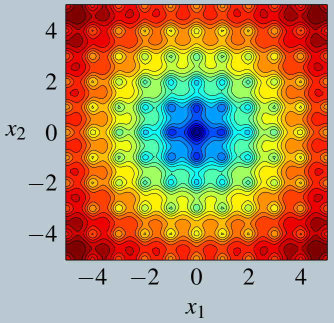
- Introduce functions, that are hard to approximate due to complexity or high variance
- Test the limits of structural complexity of metamodeling algorithms

Ackley's Function (Ackley 1987)

$$f(x) = -20\,exp\left(-0.2\sqrt{0.5(x_1^2+x_2^2)}\right) - exp\left(0.5(cos(2\pi x_1)+cos(2\pi x_2))\right) + e + 20$$



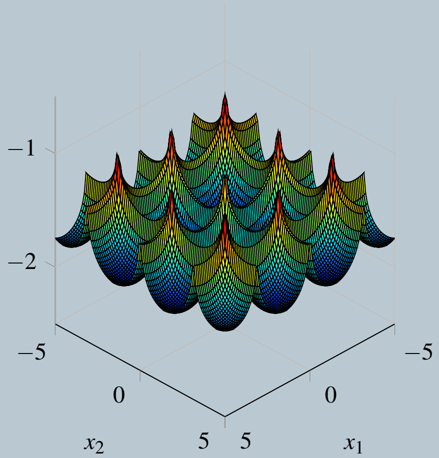
- $-5 \leq x_{1,2} \leq 5$



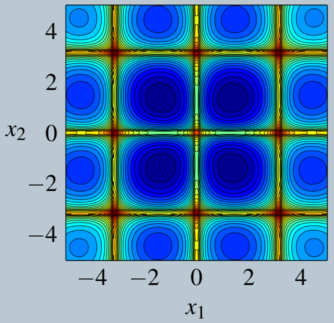
Cross-In-Tray Function (Mishra 2006)

Approx.

$$f(x) = -10^{-4}\left(\left|\sin(x_1)\sin(x_2)\exp\left(\left|100-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right|\right)\right|+1\right)^{0.1}$$

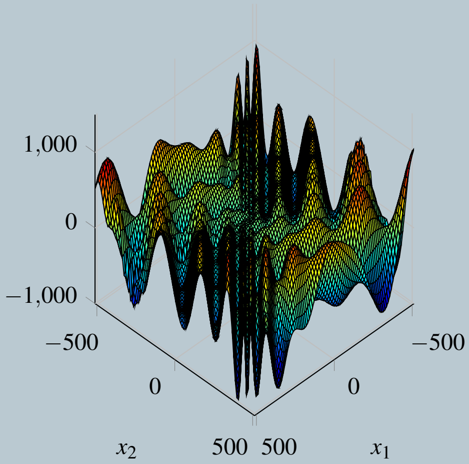


- Reduced boundaries: $-5 \leq x_{1,2} \leq 5$

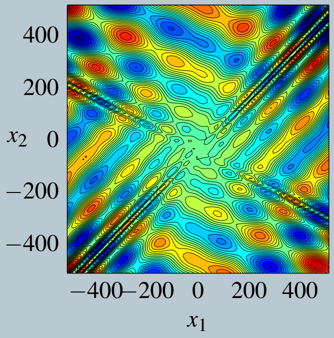


Eggholder Function (Mishra 2006)

$$f(x) = -(x_2+47)\sin\left(\sqrt{\left|x_2+\frac{x_1}{2}+47\right|}\right) - x_1 \sin\left(\sqrt{|x_1-x_2-47|}\right)$$

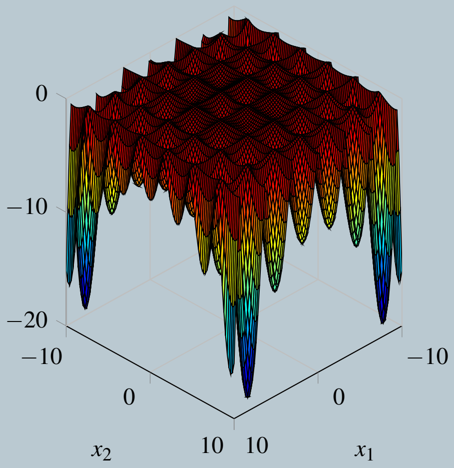


- $-512 \leq x_{1,2} \leq 512$

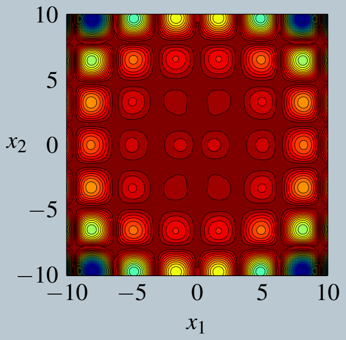


Hölder Table Function (Mishra 2006)

$$f(x) = -\left|sin(x_1)\,cos(x_2)\,exp\left(\left|1 - \pi^{-1}\sqrt{x_1^2 + x_2^2}\right|\right)\right|$$

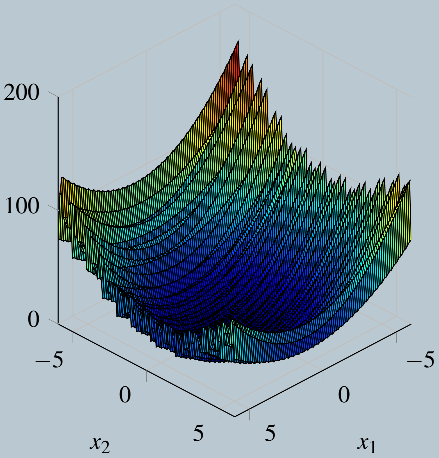


- $-10 \leq x_{1,2} \leq 10$

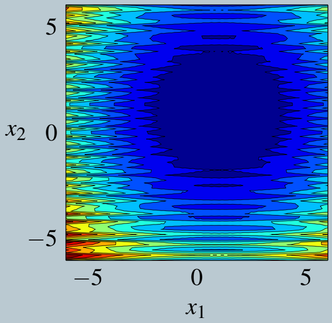


Lévy Function No.13 (Mishra 2006)

$$f(x) = sin^2(3\pi x_1) + (x_1-1)^2\left(1+sin^2(3\pi x_2)\right) + (x_2-1)^2\left(1+sin^2(2\pi x_2)\right)$$

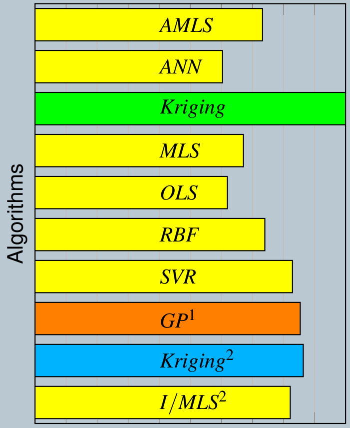


- $-6 \leq x_{1,2} \leq 6$

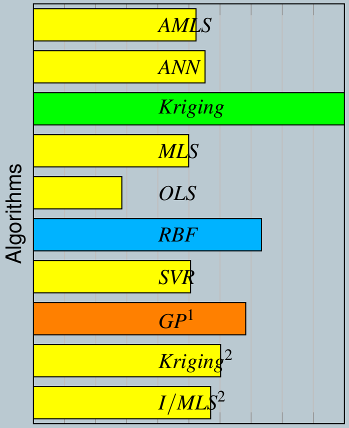


Results (High Modality)

Generalized R^2

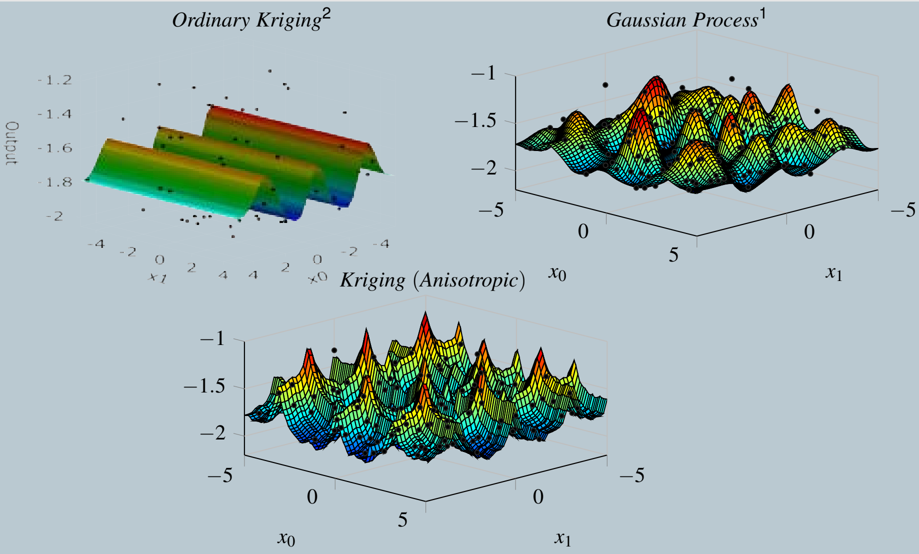


RMSE (Validation)



Visualisation - Cross-In-Tray (96 pts.)

Orig.



Boundary Emphasized Functions

Analytical functions:

- ✓ Beale's function
- ✓ Booth's function
- ✓ Goldstein-Price function
- ✓ Six-hump camel function
- ✓ Three-hump camel function

- More complex structure or higher variance near the boundaries compared to the inner regions
- Prediction errors near the boundaries overweight the prediction errors at inner region
- Less neighboring data points compared to inner regions

Aims:

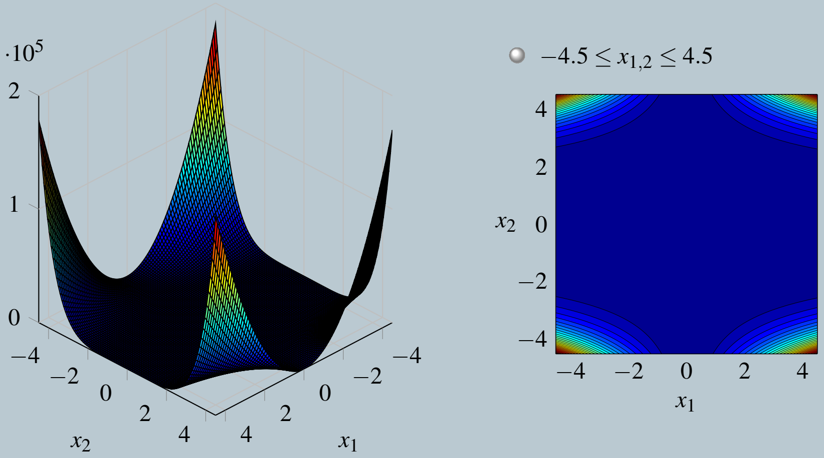
- Performance test of algorithms explicitly near boundaries

Beale's Function (JaYa 2013)

Approx.

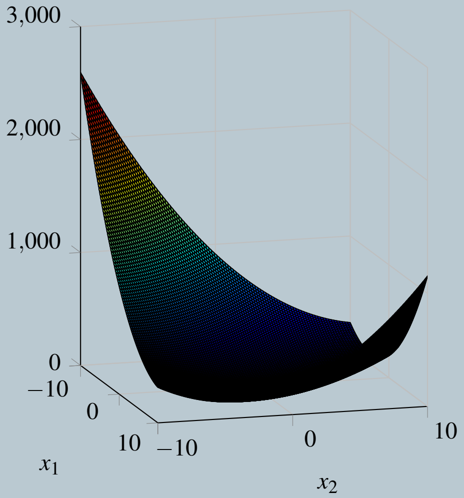
$$f(x) = (1.5 - x_1 + x_1 x_2)^2 + \left(2.25 - x_1 + x_1 x_2^2\right)^2 + \left(2.625 - x_1 + x_1 x_2^3\right)^2$$

- $-4.5 \leq x_{1,2} \leq 4.5$

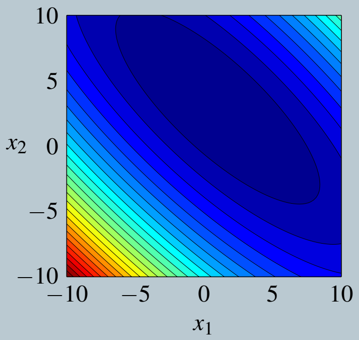


Booth's Function (JaYa 2013)

$$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

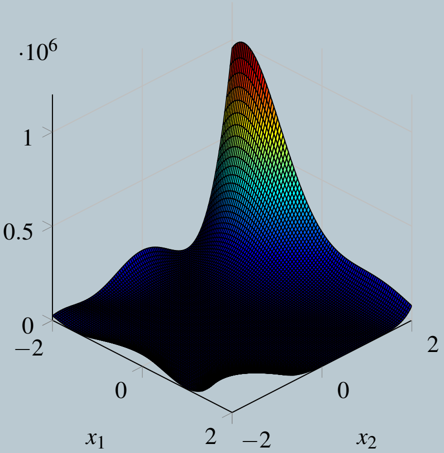


- $-10 \leq x_{1,2} \leq 10$

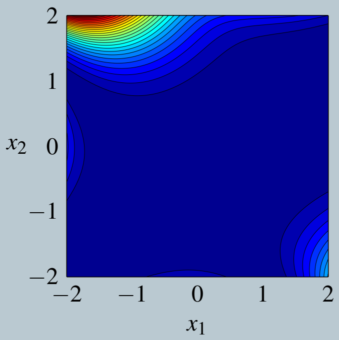


Goldstein-Price Function (GoPr 1971)

$$f(x) = \Big(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+5x_1x_2+3x_2^2)\Big)$$
$$\big(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\big)$$

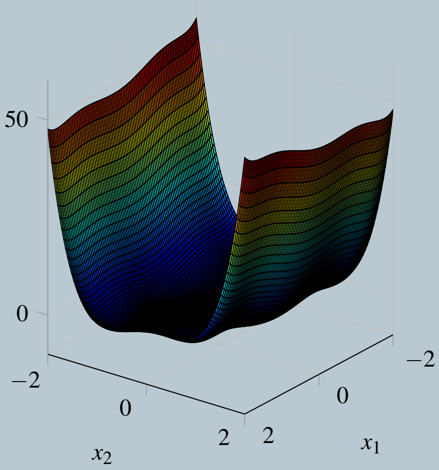


- $-2 \le x_{1,2} \le 2$

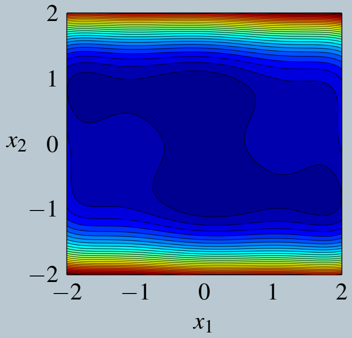


Six-Hump Camel Function (DiSz 1978)

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

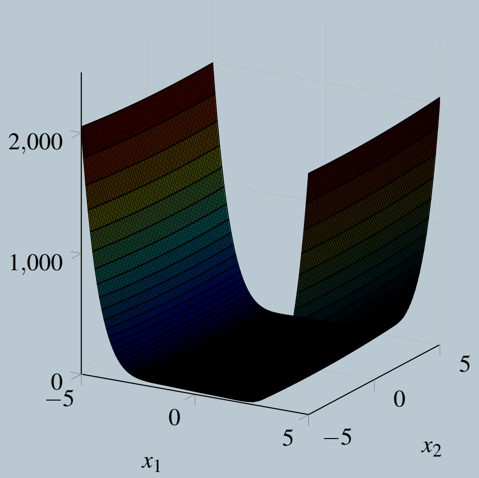


- $-2 \leq x_{1,2} \leq 2$

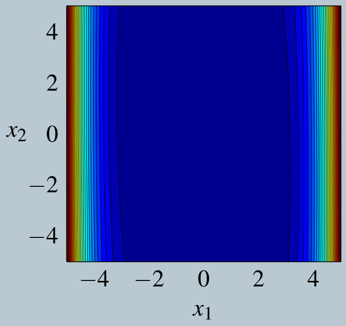


Three-Hump Camel Function (Mishra 2006)

$$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 + x_1x_2 + x_2^2$$

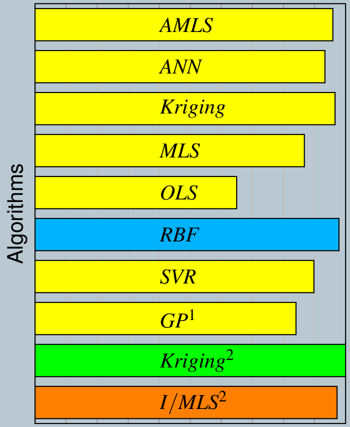


- $-5 \leq x_{1,2} \leq 5$

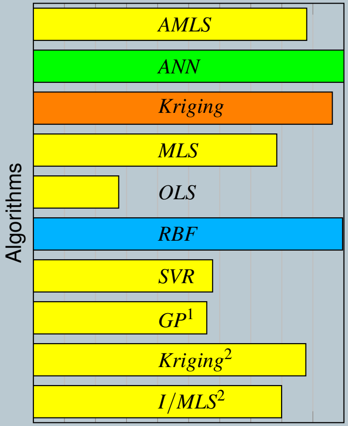


Results (Boundary Emphasized)

Generalized R^2



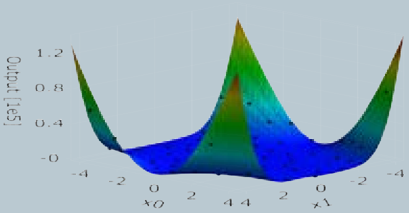
RMSE (Validation)



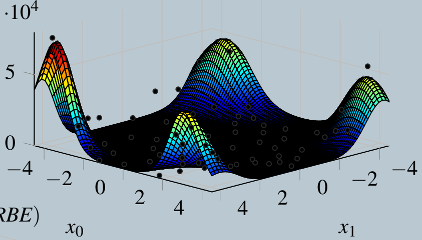
Visualisation - Beale (64 pts.)

Orig.

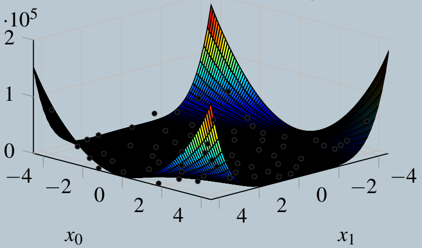
Ordinary Kriging[2]



Gaussian Process[1]



ANN (RBE)



Inner Region Emphasized Functions

Analytical functions:

- ✓ Bukin function No.6
- ✓ Easom function
- ✓ Custom probability density function
- ✓ Michalewicz function
- ✓ Ursem function

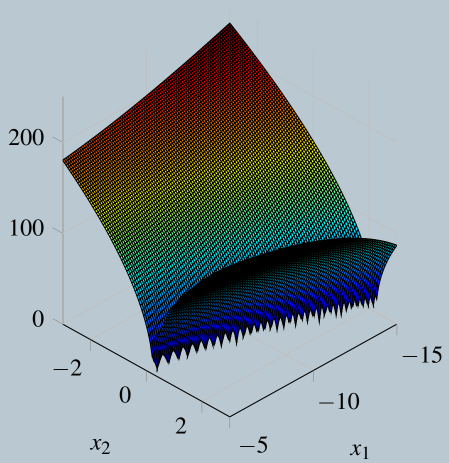
- More complex structure or high variance at the inner regions of x-subspace compared to the boundaries
- Prediction errors in the inner region should overweight the prediction errors near the boundaries

Aims:

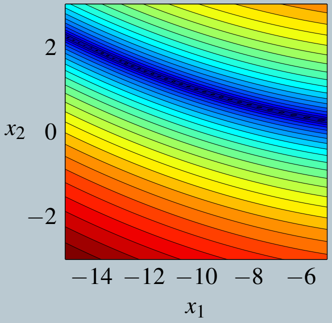
- Test the performance of algorithms explicitly at inner regions
- Dichotomy: Any given function fall into either this or to the previous category (excluding balanced out or harmonic functions like $sin(x)$)

Bukin Function No.6 (Bukin 1997)

$$f(x) = 100\sqrt{|x_2 - 0.01x_1^2|} + 0.01|x_1 + 10|$$

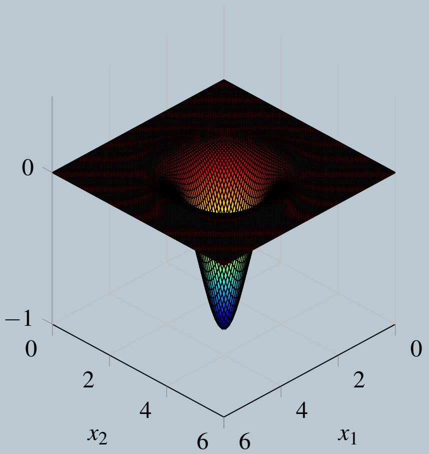


- $-15 \leq x_1 \leq -5$; $-3 \leq x_2 \leq 3$

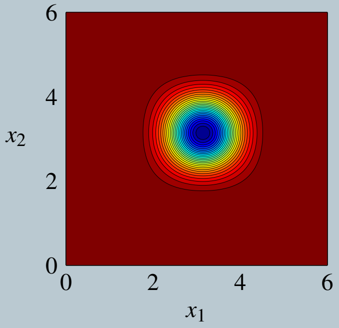


Easom Function (Easom 1990)

$$f(x) = -cos(x_1)\,cos(x_2)\,exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)$$

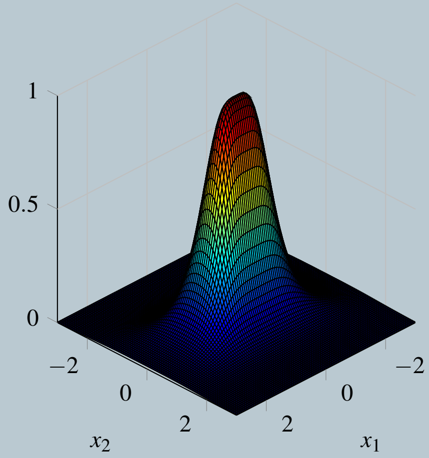


- $0 \leq x_{1,2} \leq 6$

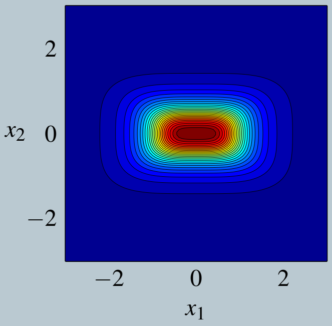


Custom Probability Density Function

$$f(x) = \left(1 + x_1^4 + 5x_2^4 + 2x_2^2\right)^{-1}$$



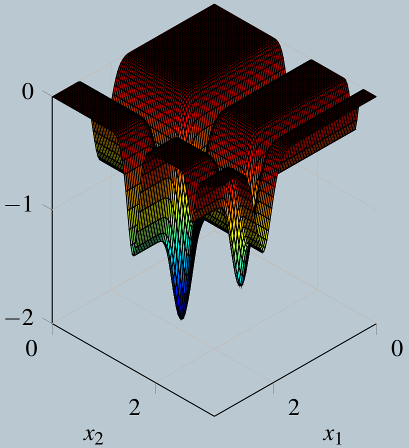
- $-3 \leq x_{1,2} \leq 3$



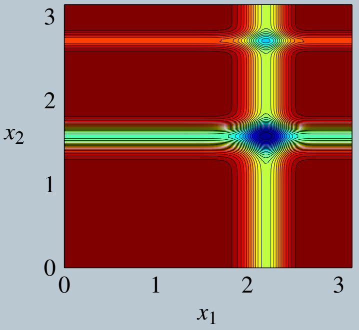
Michalewicz Function (Michalewicz 1992)

Approx.

$$f(x) = -\sum_{i=1}^{n} sin(x_i) sin^{20}\left(\frac{i \cdot x_i}{\pi}\right)$$

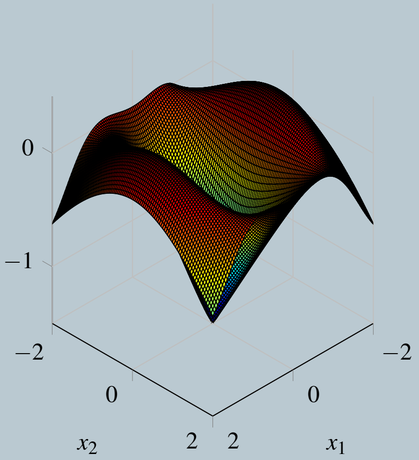


- $0 \leq x_{1,2} \leq \pi$

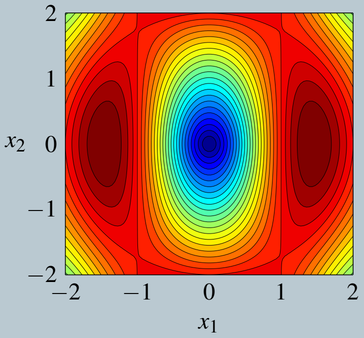


Ursem Function (Ursem 1999)

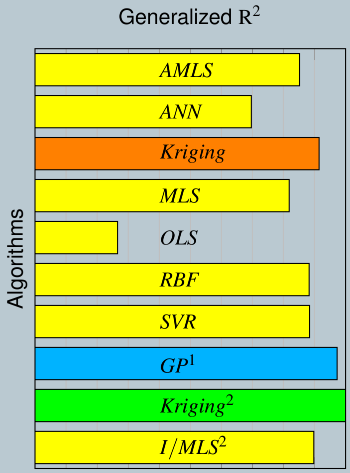
$$f(x) = -\sum_{i=1}^{n} sin(x_i) sin^{20}\left(\frac{i \cdot x_i}{\pi}\right)$$

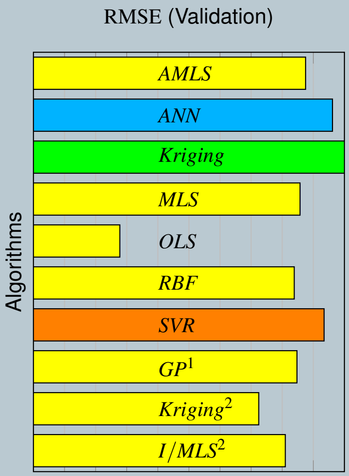


- $-2 \leq x_{1,2} \leq 2$



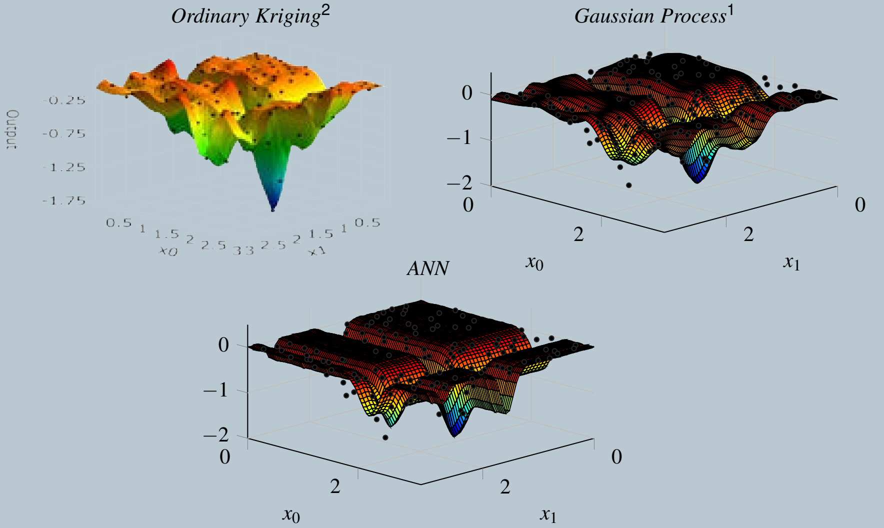
Results (Inner Region Emphasized)



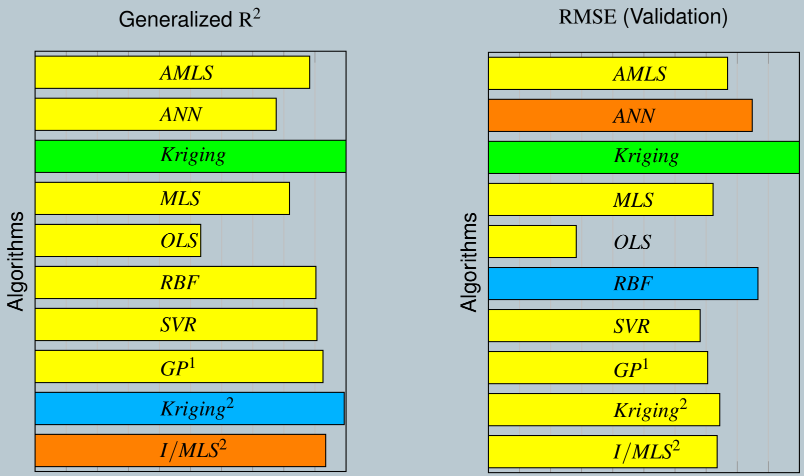


Visualisation - Michalewicz (128 pts.)

Orig.

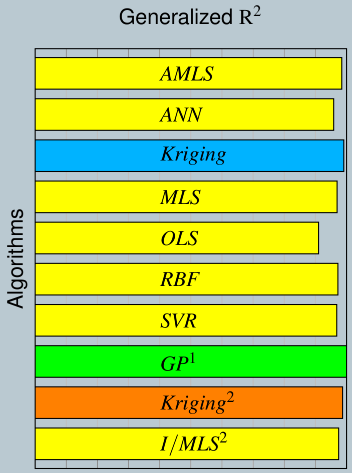


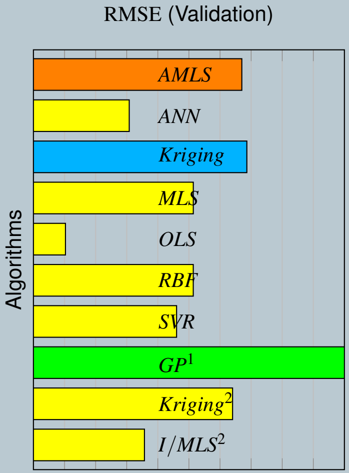
Total Scores



Inhalt

Mechanical Example (Courtesy of Bosch)





9 Input parameters, 11 responses, 568 sampling points, 860 validation points

Summary and Outlook

- All algorithms have different strengths and weaknesses.
  $\rightarrow$ Although some perform more frequently better than others, choice of an algorithm that fits the problem is vital for success ("No Free Lunch!").
- If a set of data is easy to model, most algorithms will perform good. The choice here depends on computational effectiveness. If the data is not easy to model, trying different algorithms may yield better results.
- Interpolating algorithms are better, when modeling errors are minimal and observation has little noise, while averaging algorithms can smoothen the noise.
  $\rightarrow$ Combination of interpolating and approximating algorithms may be effective in many cases

Using regression models to solve robust design optimization problems as well as to perform reliability analysis remains the main point of interest.

"All models are wrong, but some are useful."

George E. P. Box

Thanks for your attention!

Literature I

Ackley, D. H. (1987)
A Connectionist Machine for Genetic Hill-Climbing
Kluwer Academic Publishers, Boston

Bukin, A. D. (1997)
New Minimization Strategy For Non-Smooth Functions
Budker Institute of Nuclear Physics [Paper] <BUDKER-INP-1997-79>, Novosibirsk

Cremanns,K.; Roos, D. (2014)
Requirements and New Approaches of Probabilistic Optimal Design from a Practical Point of View Considering Steam Turbines
Weimar Optimization and Stochastic Days [Conference Paper], Weimar

Literature II

Dietrich, M. J.; Hartke, B. (2012)
Empirical Review of Standard Benchmark Functions Using Evolutionary Global Optimization
Institut für Physikalische Chemie, Christian-Albrechts-Universität [Paper], Kiel

Dixon, L. C.; Szego, G. P. (1978)
Towards Global Optimisation 2
Elsevier Science Ltd. North Holland, New York

Easom, E. E. (1990)
A Survey of Global Optimization Techniques
University of Louisville [Thesis], Louisville, KY

Goldstein, A. A.; Price, I. F. (1971)
On Descent From Local Minima
Mathematics of Computation [Journal], Vol. 25, No. 115

Literature III

Hansen, N.; Finck, S.; Ros, R.; Auger, A. (2009)
Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions
HAL Open Archive [Research Report] RR-6829. <inria-00362633>

Jamil, M.; Yang, X.S. (2013)
A Literature Survey of Benchmark Functions For Global Optimization Problems
Blekinge Institute of Technology [Research Paper], Karlskrona

Michalewicz, Z. (1992)
Genetich Algorithms + Data Structures = Evolution Programs
Springer-Verlag, Berlin, Heidelberg New York

Literature IV

Mishra, S. K. (2006)
Some New Test Functions for Global Optimization and Performance of Repulsive Particle Swarm Method
Department of Economics, North-Eastern Hill University [Research Paper], Shillong

Rosenbrock, H. H. (1960)
An Automatic Method for Finding the Greatest or Least Value of a Function
Computer Journal [Journal], Vol. 3, No. 3

Schaffer, R. A.; Caruana, L. J.; Eshelman, R.; Das, R. (1989)
A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization
3rd International Conf. on Genetic Algorithms [Paper], George Mason University

Literature V

Styblinski, M.; Tang, T. (1990)
Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing
Neural Networks [Journal], Vol. 3

Ursem, R. (1999)
Genetich Algorithms + Data Structures = Evolution Programs
Multinational Evolutionary Algorithms, Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on Vol.3