VMC
Consulting

FAST KRIGING SPECIFICATION

Fast Kriging (FKR) algorithm, also called Tessellated Partitions Surface with Kriging (TPS-KR)

Fast Kriging (FKR) also called Tessellated Partitions Surface with Kriging (TPS-KR)

VMC Consulting

ABSTRACT: **Artificial intelligence in general and metamodel-assisted optimizations rely on response surfaces to forecast the output of functions and are vital part of these methodologies. Yet they have important limitations, since greater precisions require greater data sets, thus, training or updating larger response surfaces become computationally expensive or unfeasible. This has been an important bottle neck limitation to achieve more promising results, rendering many optimization and AI tasks with a low performance.**

To solve this challenge, a new methodology created to segment response surfaces is hereby presented. Differently than other similar methodologies, this algorithm named tessellated partitions surface has a very simple and robust operation, generating a mesh of near isopopulated partitions of inputs by regions of similitude. The great advantage it offers is that it can be applied to any data set with any type of distribution, such as random, Cartesian, or clustered, for domains with any number of coordinates, significantly simplifying any metamodel with a mesh ensemble.

This study demonstrates how one of the most known and precise metamodel denominated Kriging, yet with expensive computation costs, can be significantly simplified with a response surface mesh, increasing training speed up to 185 times.

Keywords: metamodel segmentation, artificial neural network, support vector machine, kriging.

1. Introduction

Artificial intelligence (AI) plays an important role in society presently, due to its capacity to automate several complex or laborious tasks which before were only accomplished by humans. Present design practices of aeronautical, spatial, electrical and automotive systems also rely on metamodels used in AI to achieve approximations of precise and expensive models. Several metamodels are available in the public domain, yet the precision and the computation cost of these metamodels are often an object of research.

The tessellated partitions surface (TPS) is a novel algorithm which can create a mesh for a data set with any type of distribution or number of coordinates. It can be applied to any metamodel, thus expensive metamodels can be segmented and have their training time significantly reduced without compromising accuracy.

2. Related work

In the field of AI, Rumelhart et al [1] described the backpropagation method, which is applied to train multi-level artificial neural networks (ANNs). These ANNs are a metamodel which consists of a mathematical model that mimics the flow of information observed in organic neuronal tissues of several living species, allowing many living species to have intelligence. This methodology has been very efficient in text, image, and speech recognition, among other applications. Among some of its generalized applications, ANNs can be applied to forecast the response of a mathematical function, called regression, and for category classification.

Deep learning [2] (DL), is a technique created by Lecun, Bengio, and Hinton, where many layers of ANNs are trained over data set to recognize several levels of patterns. It has unsupervised training, different than ANNs, and has many important applications for the automation of complex and challenging tasks, substituting human work in many fields of activity. DL and other areas of AI have as their basic building blocks, response surfaces such as ANNs, Kriging [3] (KR), support vector machine [4] (SVM), polynomial regression [5], radial basis functions [6] among others models and statistical inferences, and often require large parallel computing infrastructure and large data sets for their often computationally expensive training tasks.

Several global optimization algorithms (GOAs) are popular choices to find the extrapolated points of a function. These GOAs [7], [8], [9], [10], [11], have broad applications in engineering design, logistics, AI, among other areas.

In optimization, a metamodel, which is also called a response surface or surrogate model, can be applied to reduce computation costs by dismissing unpromising candidate solutions and suggesting promising alternatives. To evaluate the efficiency of a response surfaces, many benchmark functions were created [6].

Yet precise metamodels often require large data sets and are expensive in computation costs. Some studies aiming to reduce training times of KR have been published. Fuhg et al. in 2020 [12] published a review of the state-of-the-art of adaptative sampling methods for

KR. Bouhlel et al. researched ways to reduce the computation cost of the KR model [13], by substituting the inversion of a matrix in the KR methodology with a kernel with few parameters defined by least squares and using adaptative sampling to not overload the KR model. In another study, van Stein et al. [14] described several methodologies to segment a KR model into clusters. Some of these techniques are based on slicing the domain in a Cartesian segmentation, yet those methodologies have limited application for data set with clustered distribution. Other methodologies, as described by Wang and Simpson [15] include a fuzzy analysis of the landscape of the function, to segment in locally optimal regions. Yet these proposed methodologies cannot be applied to any type of data set and have several operational coefficients which must be adjusted by the user according to each function.

Wang et al. [16] in 2017 explored similar application of data sampling selection, including domain landscape analysis for KR clustering applied to optimization tasks. Liu et al. [17] described experiments to reduce computation time of the KR model for large data sets, by applying global and local sparse approximations of the overall inputs. Other alternatives presented include selecting subsets of the data set to reduce KR computation costs, and nesting KR, as described by Bachoc et al. [18] in 2021. A popular methodology also in the same filed of research, is the adaptative sampling [19], which Chellappa et al, in 2020, applied to the reduced basis method. This adaptative sampling consists of a surrogate error model generated to efficiently create sub samples of the parameter domain. There are many other publications available in the topic, aiming to reduce metamodel computation costs, which most often are based on a type of clustering or selective sampling the data set. Yet these methods often do have a non-trivial implementation, requiring many operational parameters to be adjusted according to the data set, or cannot be applied to any type of data set distribution.

3. Methodology

The TPS generates a response surface mesh for a normalized data set with a Cartesian, random, or clustered distribution, by separating the data set into partitions of data which share of similitude.

The advantage of this method is that it works equally with data sets with any number of coordinates or type of distribution while refining the mesh in regions of the domain with greater data density.

The TPS also can be applied to generate a mesh for a finite element model or similar mathematical models, discretize the domain of a function and identify its most competitive outputs, among other applications which require domain partitioning. Yet one of the most useful applications of the TPS is its combination with KR (TPS-KR), significantly reducing training its times while maintaining the same precision.

4. Numerical experiment

In order to compare the efficiency of segmented and non-segmented response surfaces, several benchmark functions used for this purpose are selected [20]. The metamodel applied in this experiment is the KR, also called Gauss Process, which is a popular metamodel in optimization and AI, used to forecast the output of a function.

Each function has its 2D domain defined in a Cartesian grid of 71 x 71 points, with a total of 5041 inputs, where the error of the output and other performance parameters are presented. Tables 1 to 5 display the comparison in terms of performance of the metamodel and the overall time delayed on training.

The metrics for metamodel performance measurement are the overall training time, average error, and variance of the error. Other metrics are also applied, which are R square, RMAE, and RAAE, as described by Jin, Chen, and Simpson [21]. It is important to note that for the TPS, each element of the metamodel was training in parallel computing using a dual-core Intel i7 processor at 2 GHz, and the overall training time includes the mesh generation.

4.1. Experiment I

The results comparing the regular KR to the segmented KR model using TPS (TPS-KR) are presented in Tables 1 to 5:

Ackley function		
response surface	KR	TPS-KR
n segments	1	189
total training time	5min 5.82sec	16.25sec
mean_error	4.537e-01	1.539e-03
variance	3.258e-01	2.703e-05
r_square	9.968e-01	1.000e+00
raae	4.502e-02	1.525e-04
rmae	2.737e-01	1.237e-02

Table 1 – Performance comparison for the Ackley function

Beale function		
response surface	KR	TPS-KR
n segments	1	189
total training time	31min 55.54sec	10.34sec
mean_error	2.082e-01	2.763e-01
variance	9.538e-02	4.266e-01
r_square	1.000e+00	1.000e+00
raae	1.617e-04	2.145e-04
rmae	3.031e-03	6.496e-03

Table 2 – Performance comparison for the Beale function

Booth function		
response surface	KR	TPS-KR
n segments	1	189
total training time	24min 36.7sec	10.66sec
mean_error	3.837e-02	1.854e-01
variance	8.935e-04	3.766e-01
r_square	1.000e+00	1.000e+00
raae	6.182e-05	2.986e-04
rmae	2.250e-04	1.948e-02

Table 3 – Performance comparison for the Booth function

Holder Table function		
response surface	KR	TPS-KR
n segments	1	189
total training time	8min 55.11sec	10.23sec
mean_error	2.967e-03	1.073e-03
variance	3.999e-05	1.684e-05
r_square	9.999e-01	1.000e+00
raae	4.186e-03	1.514e-03
rmae	9.487e-02	1.010e-01

Table 4 – Performance comparison for the Holder Table Function

Custom probability density function		
response surface	KR	TPS-KR
n segments	1	189
total training time	14min 35.03sec	10.72sec
mean_error	1.640e-04	2.135e-04
variance	1.476e-07	1.024e-07
r_square	1.000e+00	1.000e+00
raae	8.192e-04	1.066e-03
rmae	1.238e-02	1.149e-02

Table 5 – Performance comparison for the Custom Probability Density Function

From table 1 to 5 is possible to see that the TPS has significantly reduced the training time which ranged from 31 to 5 minutes to about 10 to 16 seconds. Experiments have demonstrated that for very large data sets, training time for the non-segmented model can achieve many hours or days, while the KR model with segmented mesh remains significantly lower. Also, it is noted that there is not difference in terms of the precision for the TPS. Adding to this that the TPS can be implemented on any data set, it is demonstrated that it is an important alternative to reduce computation costs of metamodels.

4.2. Experiment II

In order to compare the influence of the data set size in the training costs related to KR and FKR, also called TPS-KR, samples from 500 2D inputs up to 5000 inputs have their training cost measured, while using one processor of the Apple 2020 Macbook M1 chip, at 3.5 GHz. The results are displayed at Figure 1:

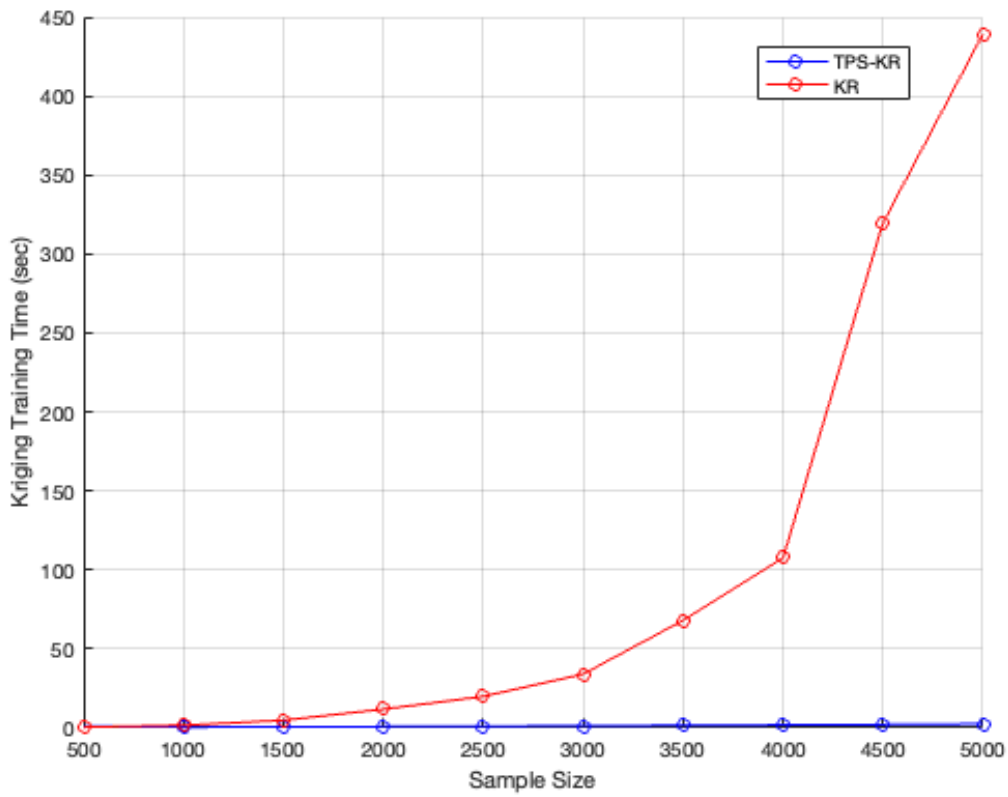

Fig. 1 - Computation costs of KR and TPS-KR vs data set size

Table 6 displays the same values of Figure 1. Is possible to see that for a dataset with 5000 inputs, KR training took 7 minutes and 23 seconds, while for TPS-KR the training time was of 2.26 seconds, a speed increase of 195 times for a dataset with 5000 inputs.

dataset size	KR	TPS-KR	ratio
500	0.46 sec	1.24 sec	0.37
1000	1.66 sec	0.43 sec	3.85
1500	4.65 sec	0.67 sec	6.94
2000	10.87 sec	0.79 sec	13.78
2500	18.8 sec	1.01 sec	18.52
3000	41.96 sec	1.26 sec	33.31
3500	1 min 11.81 sec	1.94 sec	37.08
4000	1 min 50.51 sec	1.86 sec	59.26
4500	5 min 20.5 sec	2.11 sec	152.14
5000	7 min 22.75 sec	2.26 sec	195.51

Tab. 6 – Computation costs of KR and TPS-KR vs dataset size, with efficiency ratio

5. Conclusions

Optimization and AI, in general, depend greatly on metamodels to provide accurate forecasts of functions. One of the most known metamodels is KR, which provides good accuracy to complex functions. Its great limitation, however, is that it can be very expensive in terms of computation costs for large data samples, since training a KR model for large data sets can require hours or days of processing time. As demonstrated in the experiment, with the TPS a KR model can be segmented, greatly reducing the overall computation cost. The time reduction achieved with the TPS in experiment I is from about 5 to 31 minutes to about 10 seconds, a speed increase between 20 to 185 times.

In experiment II, it is achieved a speed increase of 195 times using TPS-KR for a dataset of 5000 inputs. It is noted on Figure 1 that KR training costs increases exponentially with sample size, while TPS-KR keeps much lower computation costs.

These results demonstrate that the TPS can be an important solution to many AI and optimization tasks, greatly reducing the often expensive KR training costs. It is noted, in addition, that the TPS algorithm can be applied to any metamodel, over a data set with any type of distribution, such as Cartesian, random or clustered, making of it an important alternative to metamodel assisted optimization and AI tasks in general.

6. References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[2] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 2015. doi: 10.1038/nature14539.

[3] C. E. Rasmussen, "Gaussian Processes in Machine Learning," in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71. doi: 10.1007/978-3-540-28650-9_4.

[4] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 297, pp. 273–297, 1995.

[5] R. Jin, W. Chen, and T. W. Simpson, "Comparative studies of metamodeling techniques under multiple modeling criteria," *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, doi: 10.2514/6.2000-4801.

[6] C. Bogoclu and D. Roos, "A benchmark of contemporary metamodeling algorithms," *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*, vol. 2, no. June, pp. 3344–

3360, 2016, doi: 10.7712/100016.2039.7645.

[7] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[8] C. R. Reeves and J. E. Rowe, *Genetic Algorithms - Principles and Perspectives. A guide to GA Theory.*, vol. 4, no. 1. 2003.

[9] T. H. A. Scollen, “Simulated Annleaing, Introduction, Application and Theory,” *Nove Science Publishers, Inc. New York*, 2018.

[10] R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.

[11] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013, doi: 10.1007/s00366-011-0241-y.

[12] J. N. Fuhg, A. Fau, and U. Nackenhorst, “State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging,” *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 2689–2747, 2021, doi: 10.1007/s11831-020-09474-6.

[13] M. A. Bouhlel, N. Bartoli, A. Otsmane, and J. Morlier, “Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction,” *Structural and Multidisciplinary Optimization*, vol. 53, no. 5, pp. 935–952, 2014, doi: 10.1007/s00158-015-1395-9.

[14] B. van Stein, H. Wang, W. Kowalczyk, M. Emmerich, and T. Bäck, “Cluster-based Kriging approximation algorithms for complexity reduction,” *Applied Intelligence*, vol. 50, no. 3, pp. 778–791, 2020, doi: 10.1007/s10489-019-01549-7.

[15] G. G. Wang and T. Simpson, “Fuzzy clustering based hierarchical metamodeling for design space reduction and optimization,” *Engineering Optimization*, vol. 36, no. 3, pp. 313–335, 2004, doi: 10.1080/03052150310001639911.

[16] H. Wang, B. van Stein, M. Emmerich, and T. Bäck, “Time Complexity Reduction in Efficient Global Optimization Using Cluster Kriging,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 889–896. doi: 10.1145/3071178.3071321.

[17] H. Liu, J. Cai, Y.-S. Ong, and Y. Wang, “Understanding and comparing scalable Gaussian process regression for big data,” *Knowledge-Based Systems*, vol. 164, pp. 324–335, 2019, doi: https://doi.org/10.1016/j.knosys.2018.11.002.

[18] F. Bachoc, N. Durrande, D. Rullière, and C. Chevalier, “Properties and comparison of some Kriging sub-model aggregation methods,” Feb. 2021.

[19] S. Chellappa, L. Feng, and P. Benner, “An Adaptive Sampling Approach for the Reduced Basis Method.” 2020.

[20] C. Bogoclu and D. Roos, “A benchmark of contemporary metamodeling algorithms,” *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*, vol. 2, no. June, pp. 3344–3360, 2016, doi: 10.7712/100016.2039.7645.

[21] R. Jin, W. Chen, and T. W. Simpson, “Comparative studies of metamodeling techniques under multiple modeling criteria,” *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, doi: 10.2514/6.2000-4801.