Hierarchical Surrogate Modeling for Illumination Algorithms

Alexander Hagg
Bonn-Rhein-Sieg University of Applied Sciences
Grantham-Allee 20
Bonn, Germany 53757
alexander.hagg@h-brs.de

ABSTRACT

Evolutionary illumination is a recent technique that allows producing many diverse, optimal solutions in a map of manually defined features. To support the large amount of objective function evaluations, surrogate model assistance was recently introduced. Illumination models need to represent many more, diverse optimal regions than classical surrogate models. In this PhD thesis, we propose to decompose the sample set, decreasing model complexity, by hierarchically segmenting the training set according to their coordinates in feature space. An ensemble of diverse models can then be trained to serve as a surrogate to illumination.

CCS CONCEPTS

•**Computing methodologies → Bagging;** •**Mathematics of computing → Bio-inspired optimization;**

KEYWORDS

surrogate modeling, evolutionary illumination, bagging

1 INTRODUCTION

Computer-automated design was first introduced in 1963 by Kamentsky and Liu [14], who created a computer program to design character recognition logic based on the processing of data samples. Many applications and methods have since been developed. Today, optimization of high-dimensional problems, such as in computational fluid dynamics [8] or robotics [7], can be extremely expensive in terms of computational effort.

To decrease the necessary effort to optimize an expensive objective function, approximative models are used to serve as a *surrogate* for these simulations. *Surrogate-assisted optimization* (SAO) is a technique which, supported by models from machine or statistical learning, is typically used when the objective function is complex,

data is scarce or evaluation of potential solutions is expensive. A surrogate model only needs to learn an approximation of the objective function, also called the *response surface*, close to optimal solutions. In evolutionary optimization (EO) specifically, the approximation accuracy requirement devolves to a ranking accuracy requirement. The model only needs to accurately compare solutions.

The necessary number of evaluations can be further reduced by using Bayesian optimization (BO) [5]. The technique uses a prior over the objective function and evidence from known samples to select the best next observation based on a utility function, also called an *acquisition function*. This function balances exploration, sampling from uncertain areas, and exploitation, choosing samples that are most likely to perform well.

Both techniques, surrogate-assistance and the acquisition function, are used for *data efficient learning*. In this paradigm, learning requires a modeling technique that can not only accurately predict the objective function, but also estimate its prediction confidence.

SAO is used to find a single solution, or a Pareto front of solutions in multi-objective optimization. To find more interesting designs or design principles, *illumination* algorithms, introduced by Mouret et al. [17], are used to explore the relationship between user-defined features and the maximal performance. For example, if an engineer wants to design a car, the relationship between the volume of the car's luggage space and its turning radius can be illuminated. The engineer can use this feature relationship as a basis for design decisions. The method needs many evaluations of the objective function, which can be greatly reduced using surrogate-assisted illumination (SAIL), which was introduced by Gaier et al. [9].

2 SURROGATE-ASSISTED OPTIMIZATION

The SAO process is depicted in Figure 1. Initial training of the surrogate model is performed on a small training set, which is evenly spread over parameter space. The expensive objective function is used to retrieve samples' fitness value. Then, a surrogate model is created based on the initial sample set.
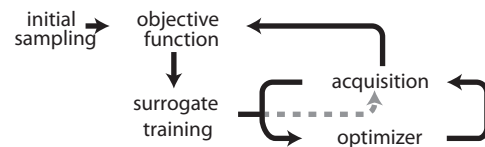

Figure 1: Surrogate assisted optimization.

The surrogate model is updated in an online fashion. The optimization is used to *explore* the model for optimal solutions. In Bayesian optimization, to acquire new samples, the model's confidence about its prediction is included. A common acquisition

function is upper confidence bound (UCB) sampling [1]. The uncertainty of the model is added to the predicted fitness to overestimate the solution's performance. The weighting of fitness and confidence controls the exploratory and exploitative behavior of the optimizer. This helps lowering *regret*: even if the sample is not of high fitness, the model benefits from this additional knowledge.

Because Gaussian process regression (GPR) models are effective with small sample sets and include an uncertainty measure, they are often used in optimization [5, 7–9, 13, 20]. A GPR describes a random distribution of functions which are defined by the mean function *m* and the covariance function *k*. GPR models are interpolative. The predicted value depends on the proximity to a known sample and its value, which is defined by the covariance function.

Optimization performance does not depend on the surrogate model's prediction accuracy alone. Depending on the chosen optimization technique, *overfitting* of the surrogate model can have an adverse effect on the convergence speed of the optimizer. As can be seen in Figure 2, in gradient based approaches, a model which does not fit the data perfectly can offer a smoother, easier traversable response surface, a phenomenon also called the *blessing of uncertainty*, which is described in work by Ong et al. [18]. The same is valid in EO but to a lesser degree. Even though EO only depends on rank comparisons between solutions, they still depend on a *virtual gradient*, which is induced by the fact that solutions are only mutated in small steps. Local optima can still lead the algorithm into *traps*, leading to longer convergence times [18].

EO have shown to be better at divergent search than classical, gradient-based approaches [6]. Whereas *exploitation* of existing knowledge to improve solutions is an aspect that is central to both fields, many EO techniques exist that focus on *exploration* of the solution space, also known as *novelty search*. All techniques benefit from surrogate models that are not too accurate, creating a smoother objective function for the optimizer.
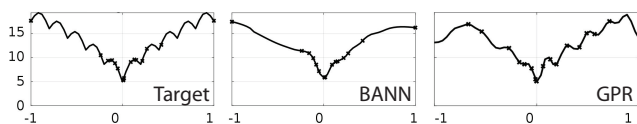
Figure 2: It is easier to find the global minimum of the objective function (left) in the smoother surrogate model (middle) than in the more accurate one (right).

3 SURROGATE ASSISTED ILLUMINATION

In recent work, Mouret and Clune introduced a new divergent optimization technique, Multi-dimensional Archive of Phenotypic Elites (MAP-Elites), which can illuminate the relationship between features and their impact on optimal solutions [17]. The technique is a mixture of optimization and novelty search. Solutions are mapped from their high-dimensional parameter space onto a low-dimensional map of features (Figure 3). The space is discretized into niches in which individuals are similar with respect to the features.

To initialize MAP-Elites, a set of random solutions is first evaluated and assigned to bins. If a bin is empty, the solution is placed inside. If another solution is already occupying the bin, the new

solution replaces it if it has a higher fitness, otherwise it is discarded. As a result, each bin contains the best solution found so far (*elite*). To produce new solutions, parents are chosen randomly from the elites, then mutated and evaluated, and finally assigned to a bin based on their feature values. Child solutions have two ways of joining the breeding pool, either by discovering an unoccupied bin or out-competing an existing solution for its bin. By repeating this process, the feature space gets increasingly explored, resulting in an increasingly optimal collection of solutions.

The technique allows engineers to generate a large number of optimal solutions that can be used to easily switch strategies in robotics control [7] or model the optimal solutions in feature space [9].
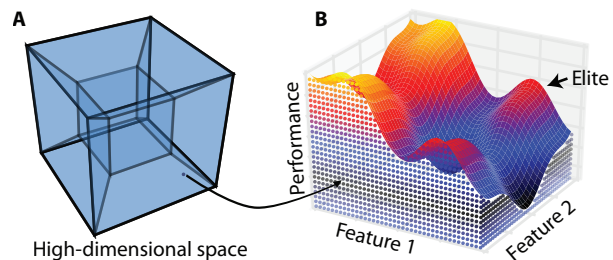
Figure 3: MAP-Elites projects high-dimensional samples onto a low-dimensional feature map (taken from [17]). The surface (right) shows the elites' objective function values.

SAIL is an extension to this algorithm. Gaier et al. [9] use a GPR model to support the illumination process by UCB sampling to add knowledge to the surrogate model during optimization. The GPR model needs to predict the objective function based on all 11 dimensions of the parameter space. In the evaluation case that was defined, optimization of a 2D airfoil, the number of dimensions and samples is not very large. When optimizing 3D shapes however, the dimensionality can easily go up to 88 or 96 dimensions [4, 11].

4 DEFICITS

In many aspects, SAIL is similar to SAO. Both are online methods that require a retrainable model. They can both benefit from smooth models that annotate their prediction with a confidence interval. In both cases, the amount of samples needed for an accurate model should be minimal, as evaluations are usually expensive.

On top of this, SAIL needs millions of comparisons, depending on the map's resolution, many more than most SAO methods. In SAO, the training of the surrogate is often more expensive than evaluation, but since SAIL needs to evaluate the model more often, models with low evaluation times are beneficial to the computational requirements of the optimization process.

Another difference is the fact that SAIL is divergent and finds many, *diverse* optima. Locally accurate surrogate models could be too expensive to accurately model for example the optimal regions for 64x64 bins. In this case, 4096 local models would have to be trained. Also, since the diversity of the training set is high and bins are defined by non-linear features, the optimal regions within the bins do not necessarily belong to a continuous optimal region in the objective function (Figure 4). In this simplified example, notice that samples belonging to the same bin can belong to different regions

in parameter space. Within each bin, the surrogate model only needs to be accurate towards the more optimal solutions, but this region can consist of solutions from multiple regions in parameter space. A single surrogate model would have to learn the high-order polynomial (green), although for rank comparisons in illumination it can be sufficient to learn linear trend lines (gray).
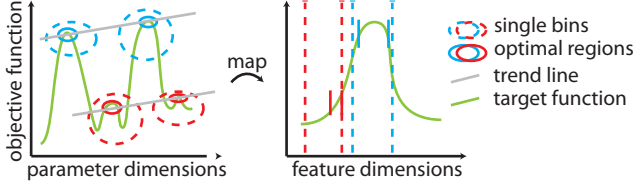
Figure 4: Left: parameter space and objective function for surrogate model. Right: map produced by non-linear feature transformation.

Training a global model that approximates the entire objective function is not feasible in optimization, as samples are expensive and we will not be able to create sufficient samples to train such a high-dimensional model. A contradiction seems to have opened up. On one hand, in order to have models that allow comparing all individuals in a bin, a local model could be trained that specializes on this task. On the other hand, tractability will be lost if every bin gets its own local model. We therefore need to train models that are more general than local surrogates but less complex than global models, placing illumination surrogates between the two techniques in terms of accuracy and efficiency (Figure 5). This surrogate technique needs to be accurate for larger and more diverse regions of the input space and we expect it to be less resource efficient, when compared to local models.
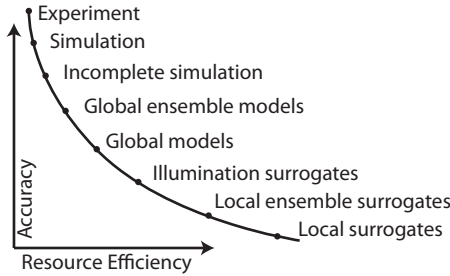

Figure 5: Trade-off between accuracy and resource efficiency [13].

SAIL will push GPR to its limits in terms of computational complexity when applied to real world problems. Training complexity is $O(n^3)$ (n = number of samples) for training and $O(n^2)$ for prediction [19]. Optimized versions exist, reducing the complexity to $O(m^2 n)$ for training and $O(m^2)$ for prediction by reducing the number of interpolation points using m pseudo-inputs [21].

Ensemble methods, like bootstrapped artificial neural networks (BANN) [3], can provide confidence measures as well, by training a *homogeneous* set of models. The ensemble's prediction is calculated by taking the mean of all its members. A confidence measure is obtained by evaluating the variation in member predictions.

We compared GPR with BANN[1] using a hill climber to optimize a 1D Ackley function. The hill climber was initialized from 10 fixed equidistant starting locations in every run in order to show how a naive optimization algorithm would perform from any starting position. The experiment was replicated 100 times. Figure 6 shows all optima that were found. Although the GPR-assisted runs converge quicker than the BANN, most runs do not reach the global optimum. The median and variance of the discovered minima is much larger than the ones found using BANN-assisted hill climbing. This is due to the much smoother BANN model (Figure 2). GPR-assisted illumination might not benefit as much from the blessing of uncertainty as do more global models like artificial neural networks (ANN).
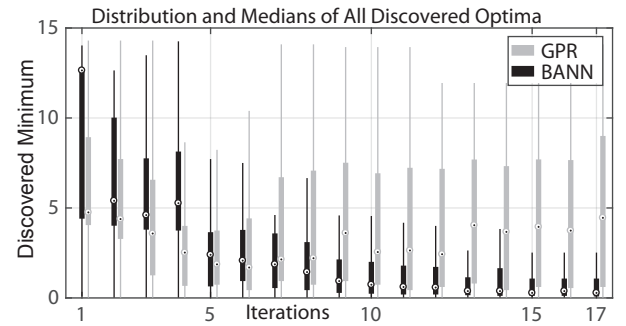
Figure 6: Comparison of surrogates assisting hill climber.

5 PROPOSAL

In order to support illuminating a high-dimensional search space, we propose to segment the objective function into simpler functions, along the same feature dimensions used by the illumination algorithm. This decomposition can simplify the modeling process, by allowing the optimal region in a bin, which can be disparate in parameter space, to be approximated by lower complexity models.
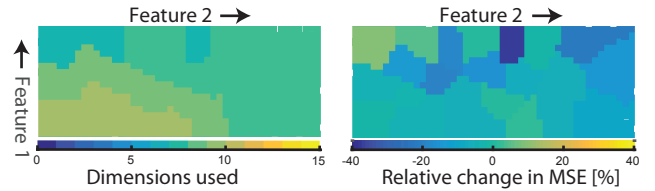
Figure 7: Dimensionality reduction and changes in modeling errors on airfoil optimization map.

Figure 7 shows a preliminary result on modeling 2D airfoils, produced by MAP-Elites. The airfoils are defined by 15 parameters, typical airfoil features are used for the map, and the objective function is the aerodynamic drag coefficient of the airfoils. The sample set is segmented using unbiased k-means. The dimensionality in every segment is reduced using principal component analysis and

---

[1]trained with Levenberg-Marquardt [15]

dimensions which explain less than 1% of total variation removed. ANN surrogates are trained to predict the drag coefficient. The segmentation leads to a varying dimensionality reduction (left) in all segments. In many segments, the error, compared to the error of a flat model trained on all samples, is reduced as well. Various segmentation methods will be evaluated.

Hypothesis 1. Using segmentation of training samples in feature space, decomposing the problem of modeling the objective function can lead to dimensionality reduction, simplifying training.

Using separate surrogates for all training set segments might be too naive, as some bins could be described by the same model and some bins might need complex models that cannot be trained with only a small amount of samples. By using a hierarchical decomposition, we can create more general models, those that are trained on large, continuous regions of the fitness space, as well as more specialized models, reusing samples in many ways. This idea is borrowed from deep learning and other hierarchical approaches in computer vision [2]. Figure 8 shows how a hierarchical decomposition of the map could be directly mapped onto a surrogate model structure. In the example above (Figure 7), every submodel on layer *n* is connected to their parent models using residual coupling [12]: models are trained to predict the discrepancy between the parent model and samples' true values.
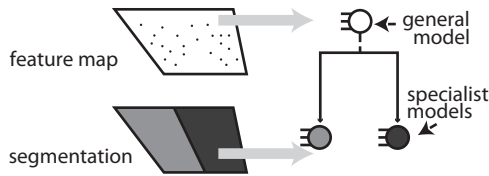
Figure 8: Hierarchical decomposition of feature map with accompanying hierarchical surrogate model.

An open question is whether to use a direct coupling between models, or to separately train models on various-sized sample subsets. In SAO and SAIL, training models is relatively cheap, because we do not have many samples. We are also willing to invest some time to reduce the amount of necessary real fitness evaluations. By training models on many subsets of the training data, we can maximize model diversity. We can use these diverse models to create a (hierarchical) ensemble, similar to *bagging*, whereby multiple models are trained to decrease the total prediction variance [3]. Gu [10] showed that diverse, *heterogeneous* ensembles often show a significantly higher accuracy than the base classifiers on their own.

Hypothesis 2. Hierarchical decomposition of training samples can be used to train a diverse ensemble of surrogate models, allowing us to learn complex structure by using many shallow models.

The large set of diverse models can be used to extract confidence intervals, but it is unclear how to combine the predictions from ensemble members in a hierarchy. Whereas in normal ensemble learning we can straightforwardly look at the prediction variance over all members to construct confidence intervals, this might not be the best option for hierarchical ensembles. *Specialist* members are expected to be more accurate than the more *general* models. We can

therefore make a more informed weighting of models' prediction, putting different confidence to different layers of the hierarchy.

Hypothesis 3. Confidence can be determined from a model hierarchy in a more informed way than direct variance estimation.

The working hypotheses do not include a specific modeling technique. In order to evaluate them, we will look at three main techniques: backpropagation training (Levenberg-Marquardt), GPR models and neuroevolution. In the latter case, by training a diverse, large number of ANNs on hierarchical training segments, we mimic the mini-batch training method that was introduced by Morse et al. [16]. This technique showed that neuroevolution can rival stochastic gradient descent training on large training problems.

We aim to increase the surrogate's accuracy and the diversity of illuminated solutions in a data efficient learning context by using hierarchical surrogate models. By combining hierarchical surrogate modeling and ensemble techniques, we reuse samples in multiple ways, increasing the probability that an accurate surrogate is found. We further reduce illumination convergence time by using smoother surrogate models. Finally, we hope to contribute a data efficient model which can be used in SAO, as well as in SAIL.

REFERENCES
[1] P. Auer. 2003. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research* 3 (2003).
[2] S. Behnke. 2003. *Hierarchical Neural Networks for Image Interpretation*.
[3] L. Breiman. 1996. Bagging Predictors. *Machine Learning* 24, 421 (1996).
[4] J. Brezillon and R.P. Dwight. 2009. Aerodynamic Shape Optimization Using the Discrete Adjoint of the Navier-Stokes Equations: Applications Toward Complex 3D Configutations. (2009).
[5] E. Brochu, V.M. Cora, and N. de Freitas. 2010. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. (2010).
[6] M. Črepinšek, S.-H. Liu, and M. Mernik. 2013. Exploration and Exploitation in Evolutionary Algorithms: A Survey. *Comput. Surveys* 45, 3 (2013).
[7] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015).
[8] M. Emmerich and B. Naujoks. 2004. Metamodel Assisted Multiobjective Optimisation Strategies and their Application in Airfoil Design. *Adaptive Computing in Design and Manufacture VI* (2004).
[9] A. Gaier, A. Asteroth, and J.-B. Mouret. 2017. Feature Space Modeling Through Surrogate Illumination. *arXiv preprint arXiv:1702.03713* (2017).
[10] S. Gu. 2016. Multi-objective and Semi-supervised Heterogeneous Classifier Ensembles. April (2016).
[11] M. Hasenjäger, B. Sendhoff, T. Sonoda, and T. Arima. 2005. Three dimensional evolutionary aerodynamic design optimization with CMA-ES. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM.
[12] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[13] Y. Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 2 (2011).
[14] L.A. Kamentsky and C.-N. Liu. 1963. Computer-automated design of multifont print recognition logic. *IBM Journal of Research and Development* (1963).
[15] K. Levenberg. 1944. A method for the solution of certain non–linear problems in least squares. *Quart. Appl. Math.* 2, 1 (1944).
[16] G. Morse and K.O. Stanley. 2016. Simple Evolutionary Optimization Can Rival Stochastic Gradient Descent in Neural Networks. Gecco (2016).
[17] J.-B. Mouret and J. Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).
[18] Y.S. Ong, Z. Zhou, and D. Lim. 2006. Curse and Blessing of Uncertainty in Evolutionary Algorithm Using Approximation. In *2006 IEEE International Conference on Evolutionary Computation*. IEEE.
[19] J. Quiñonero-Candela and C. Rasmussen. 2005. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* (2005).
[20] C. Rasmussen and C. Williams. 2006. Gaussian Processes for Machine Learning. *Gaussian Processes for Machine Learning* (2006).
[21] E. Snelson and Z. Ghahramani. 2006. Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems* (2006).