Hyperparameter Optimization via Sequential Uniform Designs; **Paper Summary**

William Jardee willjardee@gmail.com
Physics Major
Montana State University
Bozeman, MT 59715, USA

Editor: William Jardee

1. Introduction

In their paper, Yang and Zhang introduce a sequential uniform design (SeqUD) algorithm to search a hyperparameter space for an absolute optimum within provided bounds. The algorithm is motivated by the need of more efficient searching algorithms that can achieve understanding of the characteristics of a space in fewer points. This is achieved by blending uniform design theory and a sequential process. They boast that their algorithm for augmenting provided uniform designs to sets with higher resolution "marks a non-trivial contribution to the uniform design community" (Yang and Zhang, 2021). The paper is very clear on the fact that what was analyzed did not represent an optimized algorithm, but instead proposed the efficiency of such a method.

2. Background and Related Material

The theoretical backbone of SeqUD is the CD_2 function. This function is an equivalent form of the l_2 discrepancy equation, which is a decedent of the star-discrepancy (Hickernell, 1998).[1] According the Yang, et al. a uniform design is defined as a point set $D_n^* = \{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n\}$ that minimizes some criterion (2021):

$$D_n^* \leftarrow \min_{D_n \subset C^s} \phi(D_n)$$

This is changed to a vector function in **augmented uniform design**, and is brought up in the **Contributions** section. This criterion is defined from the star-discrepancy (Yang and Zhang, 2021)

$$\phi(D_n^*) = \sup_{x \in C^s} \left| \frac{|D_n \cap [0, x)|}{N} - \text{Vol}([0, x)) \right|$$

This measures the point-set's deviation from the uniform distribution. The aim is to minimize this value to maximize the quality of coverage, and consequentially the ability to

1. If you're interested in mathematical theory, I would advise you to read the paper by Hickernell. The derivation is rather fun and has some interesting tricks in it. For the point of brevity I will only include the basics of primary results of that paper.

sample the space. This methodology is not bound to any space size or number of points, offering versatility.

One of the necessities of effectively using this equation is to project it into the unit hyper-cube. Once that is done, the problem can be projected to a 2 dimensional problem and a computationally efficient form of the star-discrepancy can be used, called the CD_2 or l_2-discrepancy (Hickernell, 1998):

$$\mathrm{CD}_2(D_n)^2 = \Big(\frac{13}{12}\Big)^2 - \frac{1}{2^{s-1}\,n}\sum_{k=1}^{n}\prod_{j=1}^{s}\Big(2 + \Big|x_{kj} - \frac{1}{2}\Big| - \Big|x_{kj} - \frac{1}{2}\Big|^2\Big) + \frac{1}{2^s\,n^2}\sum_{k=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{s}\Big[2 + \Big|x_{ki} - \frac{1}{2}\Big| + \Big|x_{ji} - \frac{1}{2}\Big| - |x_{ki} - x_{ji}|\Big]$$

The key take-aways from expressing the uniform distribution are:

1. CD_2 should arise from an error bound
2. Projections of sets of D_n into a lower dimensional space should not increase l_p
3. CD_2 should be easy to compute

The Hickernell paper also points out how the value should be easy to interpret and that the measure of uniformity is invariant along certain reflections - this point is used by Yang, et al. to simplify the star-discrepancy equation from the one proposed in earlier literature (Hickernell, 1998).

The sequential portion of SeqUD is taken from a paper by Jin, et al. Jin, et al. propose using the uniform design along with a element-wise exchange method to quickly generate spaces to search (2005). Element-wise exchange can be though of implementing a hill-climb method to solve Sudoku. Your algorithm already generates random boards such that all the rows have the integers 1-9. To create a new board to test, you want to preserve this "balanced" characteristic, so, you only allow the algorithm to trade two elements within a row. This way the "balanced" characteristic of each row is preserved. SeqUD uses this same idea to search the uniform design space for the best set of points to use in the hyperparameter testing. This paper also introduces the concepts of Threshold Accepting and an Adaptive Threshold to balance the exploration vs. exploitation of the uniform design space (Jin et al., 2005).

The final inspiration used in the paper is the primary idea also used in the hyperband algorithm. This algorithm starts with a wide scope initially, but at each step narrows the search space and increases the resources allocated to each element of the search. Over time the algorithm goes from large scope, to high resolution.

3. Contributions

Not much new theory is introduced in this paper. However, the paper does propose an augmented uniform design method (Yang and Zhang, 2021):

$$D^*_{n_2} \leftarrow \min_{D_{n_2} \subset C^s} \phi\Big(\begin{bmatrix} D_{n_1} \\ D_{n_2} \end{bmatrix}\Big)$$

The paper does not go into detail about their notation and the correct interpretation of $\begin{bmatrix} D_{n_1} \\ D_{n_2} \end{bmatrix}$. Referring to Figure 5 in the paper, the most logical interpretation seem to be the concatenation of the original D_{n_1} with the a number of new points equal to the difference between n_1 and n_2 such that the points provide the best measure of uniform design.

3.1 Algorithm

The algorithm works by stringing together a collection of the aforementioned components:[2]

1. **Initial Uniform Design**: The whole space is initiated. This uses the old UD methods proposed by Hickernell. The next two steps are repeated until a certain depth is reached (Yang, et al. chose 100 iterations, but also explored 1000 iterations).
2. **Subspace Zooming & Level Doubling**: The algorithm pinpoints the most optimal point sampled and centers a new search space on that, with half the width of each dimension and the same number of points - effectively doubling the resolution of the search along each dimension.
3. **Uniform Design Augmentation**: With the new dimensions the existing points in that space are taken and the total number of points are expanded to fill that space (according to a UD).

3.2 AugUD Results

The authors measure the success of the AugUD by running 406 commonly used uniform design tables through the algorithm and measuring how much the resulting uniform design was improved. For a benchmark, the authors used an existing package, DiceDesign, to test their improvement against an already widely accepted method. Their results show that about 80% of the sampled uniform designs were improved by both algorithms. The AugUD improved the set by an average of 0.2433%, while the benchmark improved them by 0.1875%. It is pointed out that there was much more improvement made on higher dimensional sets than lower dimensional ones.

3.3 SeqUD Results

For the rest of the paper, the authors compare the performance of their SeqUD algorithm against other common non-sequential HPO algorithms, such as grid search and Sobol sequence search, as well as sequential algorithms, such as Gaussian-process expected improvement and sequential model-based algorithm configuration. All of the algorithms were run on two synthetic data sets initially. One of which was very smooth and had a singular, clear peak. The other had on the order of 10 local maximum, and was still smooth in nature. From their results, "SeqUD achieve[d] significantly better performance as compared to all the benchmarks" (Yang and Zhang, 2021). This was true for the test on 100 iteration and 1000 iterations.

A similar experiment was run on real world data with machine learning algorithms. There were a total of 40 data sets processed through a collection of the support vector

2. These are the same labels used in the paper

machine, extreme gradient boosting, and a pipeline optimization. The metrics that they used to measure the results were relative rank and time cost (in seconds). The results of the regression and classification tasks were segregated and also divided on the three machine learning algorithms used. According to their data, the SeqUD algorithm was lead to the best performance rankings and outperformed the other sequential algorithms in time cost. The time cost seems to be comparable to the non-sequential algorithms, but only just.

The algorithm was also compared against a complete pipeline optimization of AutoML in the form of AutoSKlearn. Compared against the optimized implementation, the SeqUD algorithm had a 42% out-performance rate. This result is attributed to the optimized nature of the pipeline, which has a larger scope than the SeqUD algorithm as well. The paper points out that "it is possible to enhance SeqUD with such practical pipeline optimization techniques in future work" (Yang and Zhang, 2021).

4. Discussion

This paper boasts some impressive achievements that, if fully accurate, could mean some notable steps forward for AutoML. The paper is very clear about their own shortcoming - the most notable of which was the scope of experimentation. As data sets get very large, their results start to level out relatively close to those of other methods. Their limited number of machine learning algorithms used means that the versatility of the algorithm wasn't fully measured. They did measure a SVM method; however, the hyperparameter space of a SVM is hardly misbehaved, thus the performance on very difficult problems wasn't addressed. I would have liked to see them push their algorithm on larger data sets and more machine learning algorithms, as well as show the performance of the algorithm taking advantage of the parallelization that is enable with the CD_2 calculation method. Between this and the fact that the authors claim that the code is not fully optimized yet, this paper seems to be a much stronger proof of concept than a fully realized package.

The paper draws a strong inspiration from the Jin and Chen as well as Hickernell papers, but does invoke some changes to the equations and parameters. It would have been nice to be provided with more direct justification on some of the choices. For example, in the original Hickernell piece, the subset of the hypercube used in the star-discrepancy equation has a much more rigorous definition, but Yang, et al. invoke a symmetry to simplify the statement (Hickernell, 1998).[3] Yang and Zhang also decide to incorporate exploration and exploitation into one adaptive threshold equation, while the paper they got their inspiration from split the two elements into two different equations with separately tuned parameters (Jin et al., 2005). The decision to simply copy the parameters defined in that previous paper was slightly disappointing, as it would be nice to have an explanation on the choice of the only parameters used in their algorithm.

Apart from those complaints, the proposed method seems to have promise. Seeing a rigorous approach to hyperparameter tuning is a much needed breath of air. It will be interesting to see if another follow up paper is published addressing the need for a larger experiment and more care taken to optimize the algorithm.

3. This symmetry is brought up in the Yang and Zhang paper, but it is not used to justify the modification to the equation.

References

Fred Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67(221):299–322, 1998. doi: 10.1090/s0025-5718-98-00894-1.

Ruichen Jin, Wei Chen, and Agus Sudjianto. An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134(1): 268–287, 2005. doi: 10.1016/j.jspi.2004.02.014.

Zebin Yang and Aijun Zhang. Hyperparameter optimization via sequential uniform designs. *Journal of Machine Learning Research*, 22(149):1–47, 2021.