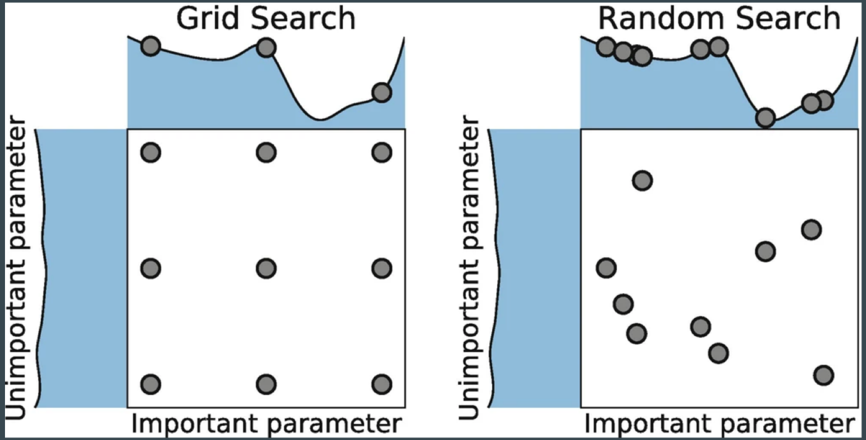
Hypercube: a DOE-informed hyperparameter optimization machine

ISYE 6413

Kaiwen Wang, Xiaochen Yan

Problem

- Good hyperparameters -> good models (CV & metrics)
- Approaches: searches, Sequential Monte Carlo, Optimization, etc.
- Traditional methods -> expensive and suboptimal
- Novel methods -> lack interpretability



Methods

- Latin hypercube (maximin)
- Response surface methodology
- Factorial design, regression, ANOVA

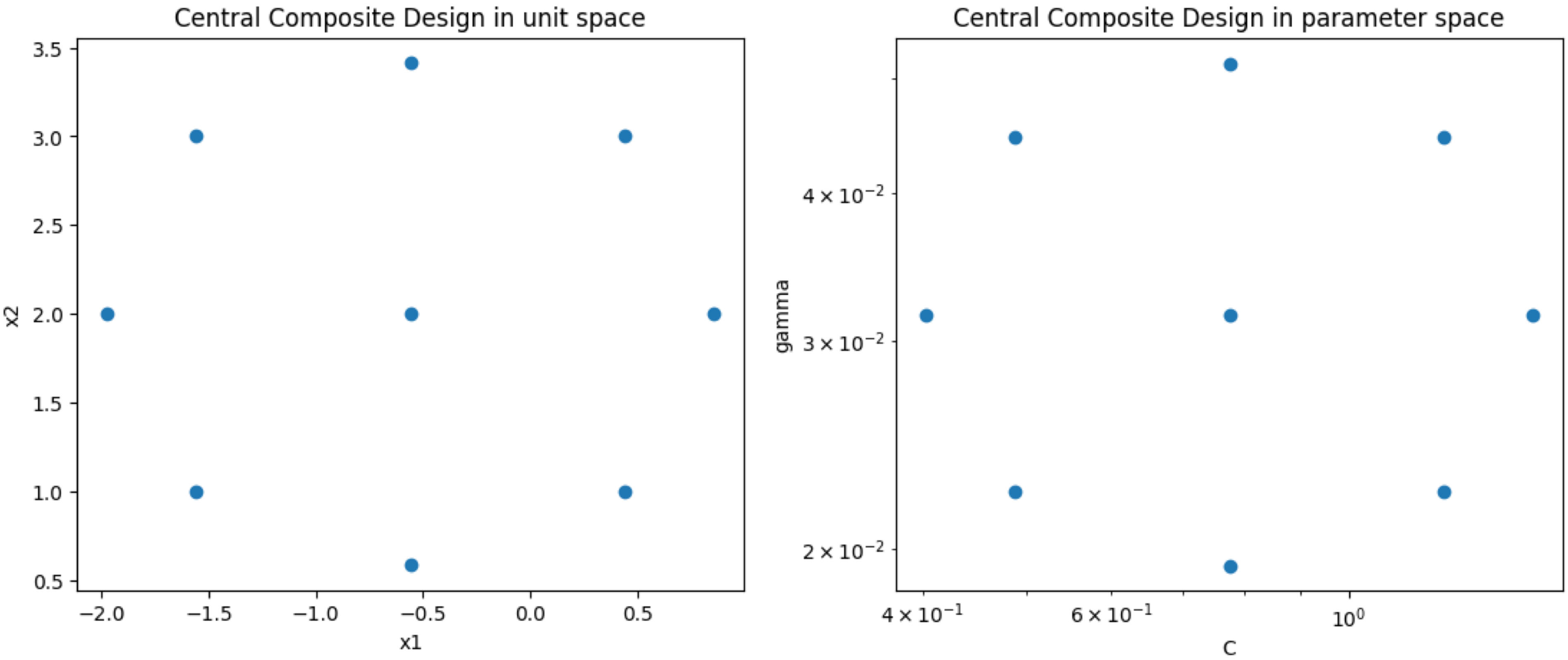
Implementation

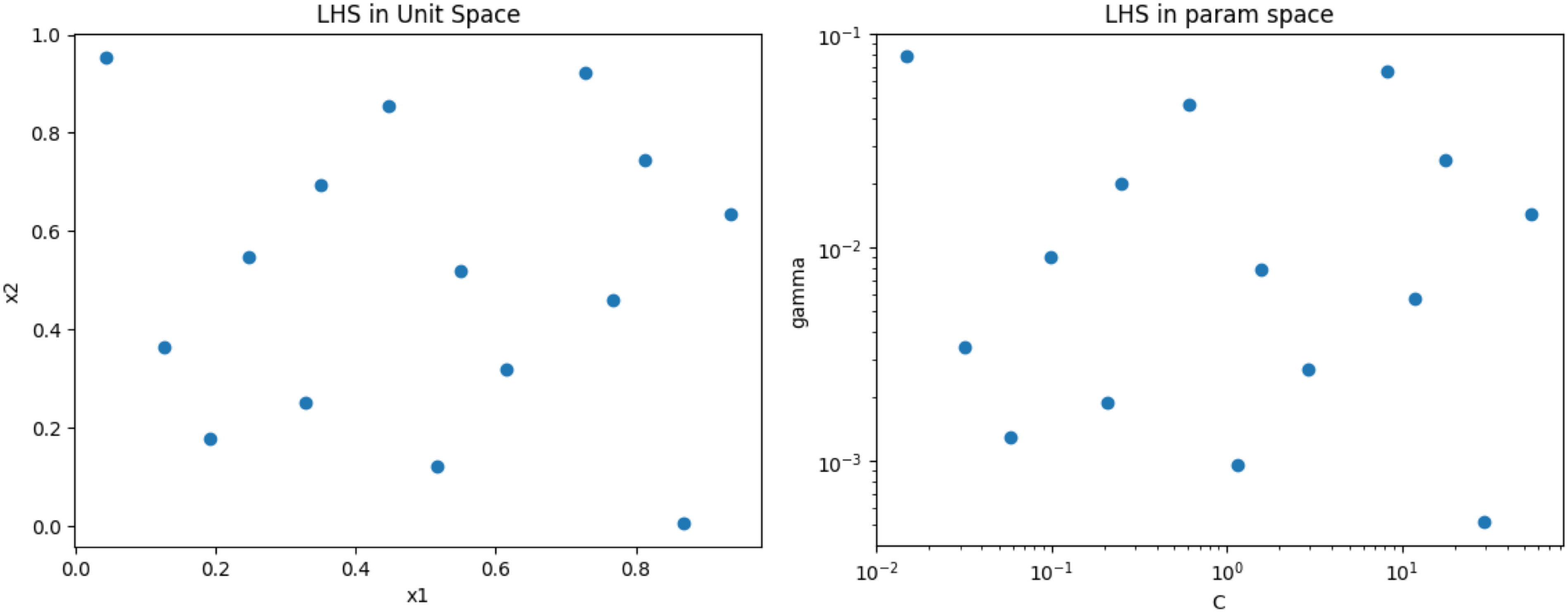
LHSTuner: Latin hypercube + Gaussian Process

Surf: First & second order designs + steepest ascent (sequential)

OFAT: factorial design + regression/ANOVA

GitHub: https://github.com/lazayxc/Hyperparameter_Tuning_with_DOE


Central Composite Design in unit space
x2
x1
3.5
3.0
2.5
2.0
1.5
1.0
0.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
Central Composite Design in parameter space
gamma
C
4×10^{-2}
3×10^{-2}
2×10^{-2}
4×10^{-1}
6×10^{-1}
10^{0}

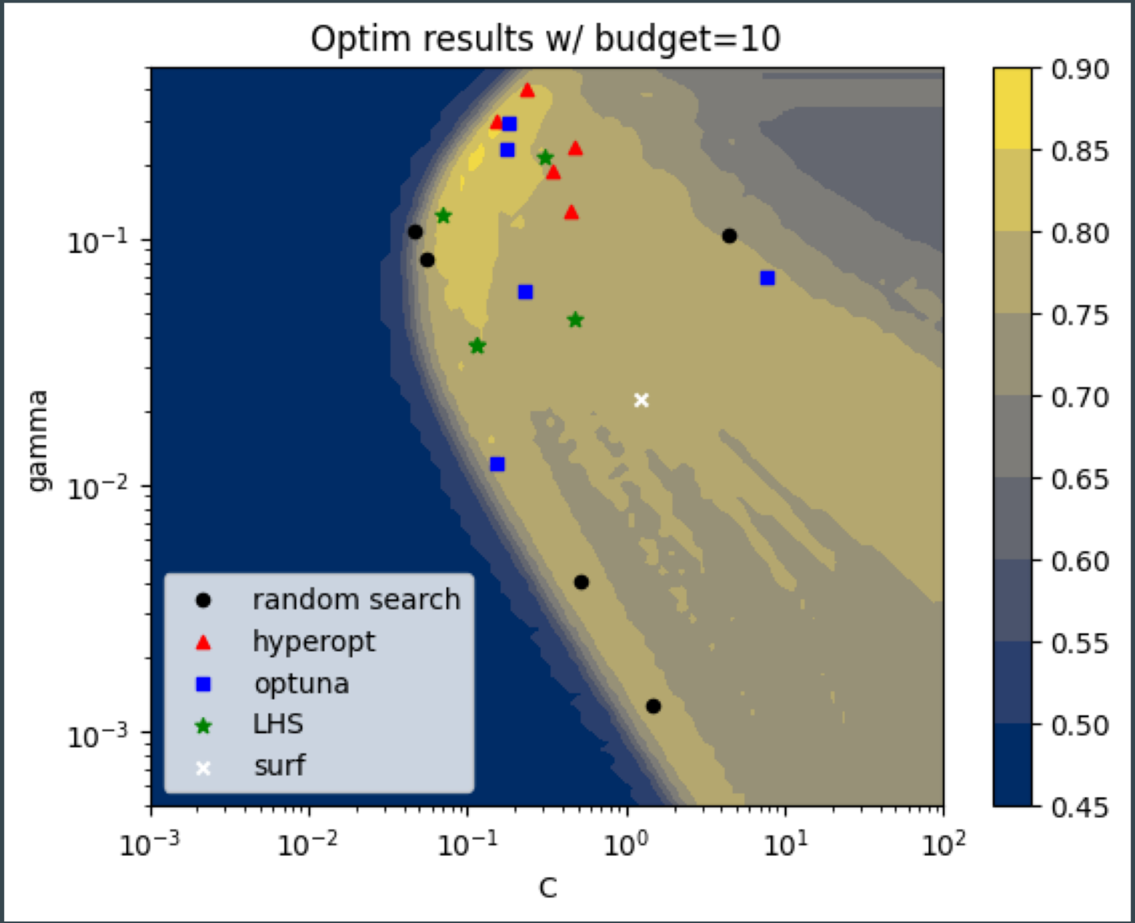

LHS in Unit Space
x1
x2
0.0
0.2
0.4
0.6
0.8
1.0
LHS in param space
C
gamma
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{-3}
10^{-2}
10^{-1}

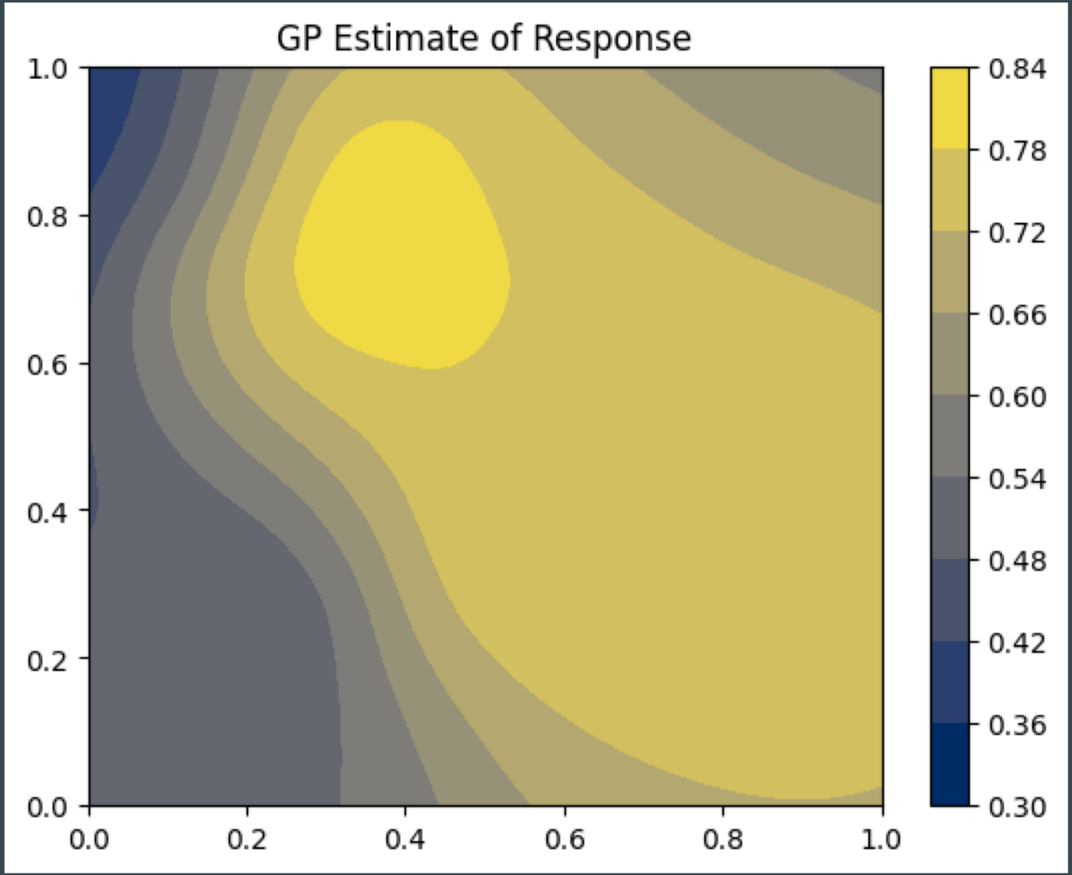
Analysis

- Synthetic classification dataset
- Models: SVM and Random Forest
- To compare: *LHSTuner, Surf,* other packages

Comparison of Design Method Efficiency (RF)			
	Total Runs	Run Time (s)	CV Score
Grid Search	3195	147.4	0.8367*
LHS	50	2.3	0.8270
Surf	150	9.9	0.8260

Comparison of Design Method Efficiency (SVC)			
	Total Runs	Run Time (s)	CV Score
Grid Search	25600	179.4	0.8550*
LHS	40	0.2	0.8352
Surf	88	0.3	0.7995


Optim results w/ budget=10
gamma
C
10^{-1}
10^{-2}
10^{-3}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
random search
hyperopt
optuna
LHS
surf



GP Estimate of Response
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
0.84
0.78
0.72
0.66
0.60
0.54
0.48
0.42
0.36
0.30

Conclusion

- Preliminary results demonstrated the potential of DoE techniques in modern ML
- *LHSTuner* is competitive against advanced algorithms (*optuna*, *hyperopt*).
- *Surf* w/ steepest ascent: expensive and sub-optimal
- GP analysis: double-edged sword

Reference

[1] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019, July). Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2623-2631).

[2] Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. Advances in neural information processing systems, 24.

[3] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).

[4] Feurer, M., Hutter, F. (2019). Hyperparameter Optimization. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5_1

[5] Frazier, P. I. (2018). A tutorial on Bayesian optimization. arXiv preprint arXiv:1807.02811.

[6] Seeger, M. (2004). Gaussian processes for machine learning. International journal of neural systems, 14(02), 69-106.

[7] Watanabe, S. (2023). Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance. arXiv preprint arXiv:2304.11127.

[8] Wu, C. J., & Hamada, M. S. (2011). Experiments: planning, analysis, and optimization. John Wiley & Sons.