Time-Optimized Bayesian Optimization of Random Forest Hyperparameters

Gabriel Xu and Andrew Chen

January 31, 2025

Dr. Yilmaz Period 5

Introduction

Addressing hyperparameter optimization for decision trees and random forests.

Challenge: Traditional tuning methods are inefficient and suboptimal.

Bayesian optimization has been used but suffers from hand-specified biases.

Our approach: Optimize both the random forest and Bayesian optimization hyperparameters.

Goal: Improve classification accuracy and efficiency.

Related Work

- Grid search and random search
- “Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization” (Zhang et al., 2019)
    - Predict which hyperparameters are likely to work best
    - Gaussian process (GP) model, acquisition function
    - Manual parameters, number of initial sample points and the choice of the acquisition function
- Garrido-Merchán & Jariego-Pérez (2021)
    - Predefined heuristics to switch through acquisition functions
- Lindauer et al. (2019)
    - Tuning Bayesian Optimization hyperparameters significantly improve its efficiency and robustness
- Our approach recursively tunes the hyperparameters of Bayesian Optimization

Dataset and Features

Dataset: breast-cancer.csv from Weka.

Contains clinical and histopathological data (e.g., tumor size, patient age, hormone receptor status).

Train-test split: 80%-20%, ensuring class balance.

Preprocessing: One-hot encoding for categorical features, discretization already applied to key variables.

No normalization needed due to categorical nature of data.

Dataset and Features

40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurre...
50-59	ge40	15-19	0-2	no	1	right	central	no	no-rec...
50-59	ge40	35-39	0-2	no	2	left	left_low	no	recurre...
40-49	premeno	35-39	0-2	yes	3	right	left_low	yes	no-rec...
40-49	premeno	30-34	3-5	yes	2	left	right_up	no	recurre...
50-59	premeno	25-29	3-5	no	2	right	left_up	yes	no-rec...

Methods

Random Forest Classifier

- Ensemble learning: Multiple decision trees, majority voting for classification.
- Helps reduce overfitting and improves accuracy.
- Key hyperparameters: number of trees, tree depth, min samples per split, min samples per leaf.

Bayesian Optimization

- Probabilistic model-based approach for hyperparameter tuning.
- Uses Gaussian processes to model function and acquisition function for next hyperparameter choice.
- More efficient than grid search or random search.

Methods

Three approaches compared:

- Basic Random Forest – Hand-selected hyperparameters.
- Bayesian Optimization – Optimized random forest hyperparameters.
- Optimized Bayesian Optimization – Tuned Bayesian optimization hyperparameters recursively.

Optimizing Bayesian hyperparameters:

- init_points: Number of initial test points.
- n_iter: Number of iterations.
- Trade-off between accuracy and runtime efficiency.

RESULTS

	Basic Random Forest	Bayesian Optimization	Hyperparameter-optimized Bayesian Optimization
Runtime (seconds)	0.24361324310302	33.5937597751617	20.9699952602386
Accuracy	0.7414	0.7586	0.7586
Precision	0.7692	0.8333	0.8333
Recall	0.4545	0.4545	0.4545

Results

Performance comparison:

- Basic Random Forest: Accuracy = 0.7414, Precision = 0.7692, Recall = 0.4545, Runtime = 0.24s.
- Bayesian Optimization: Accuracy = 0.7586, Precision = 0.8333, Recall = 0.4545, Runtime = 33.59s.
- Optimized Bayesian Optimization: Same accuracy & precision, but faster runtime (20.97s).

Key finding: Optimization reduces computation time while maintaining performance.

Results - Confusion matrices

Basic Random Forest

```
[[33  3]
 [12 10]]
```

Bayesian Optimization

```
[[34  2]
 [12 10]]
```

Hyperparameter-optimized Bayesian Optimization

```
[[34  2]
 [12 10]]
```

Results

Confusion Matrix Insights:

- All models misclassified 12 recurrence cases → Low recall issue.
- Bayesian optimization improved precision but did not enhance recall.

Takeaways:

- Hyperparameter tuning can reduce runtime while maintaining accuracy.
- More data could further improve model performance.
- Future work: Investigate methods to enhance recall while keeping efficiency.

Conclusion/Future Work

- Key Findings
    - Hyperparameter tuning can reduce runtime while maintaining accuracy
    - Meta-optimized Bayesian Optimization reduced execution time while maintaining performance
    - Both models struggled to improve recall
- Future Directions
    - Test on larger, more diverse datasets for generalizability
    - Apply to other models (e.g., GBM, SVM)
    - Explore alternative surrogate models (e.g., TPE, neural networks).
    - Investigate methods to improve recall

References

Garrido-Merchán, E. C., & Jariego-Pérez, L. C. (2021). *Towards automatic Bayesian optimization: A first step involving acquisition functions*. **Lecture Notes in Computer Science, 12882**, 160–169. https://doi.org/10.1007/978-3-030-85713-4_16

Lindauer, M., Feurer, M., Eggensperger, K., Biedenkapp, A., & Hutter, F. (2019). *Towards assessing the impact of Bayesian optimization's own hyperparameters*. **Proceedings of the DSO Workshop at IJCAI 2019**. arXiv:1908.06674. https://doi.org/10.48550/arXiv.1908.06674

Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical Bayesian optimization of machine learning algorithms*. **Advances in Neural Information Processing Systems, 25**, 2951–2959. https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). *Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms*. **Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, 847–855. https://doi.org/10.1145/2487575.2487629

Zhang, Z., Li, M., Wang, Y., & Chen, W. (2019). *Hyperparameter optimization for machine learning models based on Bayesian optimization*. **Journal of Computer Science and Technology, 34**(3), 509–522. https://doi.org/10.1016/j.jcst.2019.03.002

Q/A