

AutoML 환경에서 Bayesian Optimization 의 계산적 한계 규명

안재민, 이민주, 장우진, 백현빈, 민준기, 임지훈

대전대신고등학교

Supporting Information Placeholder (Highly Recommended)

ABSTRACT: 본 연구는 머신러닝 모델의 하이퍼파라미터 탐색에서 베이지안 최적화(Bayesian Optimization, BO)가 갖는 샘플 효율성과 실용적 한계를 규명하기 위해 수행된 AutoML 기반 심화 탐구이다. 연구에서는 가우시안 프로세스(Gaussian Process, GP)를 대리모델로 사용하는 BO를 적용하여 RandomForest와 XGBoost의 주요 하이퍼파라미터를 최적화하였으며, 초기 샘플 수, 커널 종류(RBF, Matern, RBF+WhiteKernel), length scale, 탐색 예산 등 핵심 설계 요소가 최적화 성능과 수렴 속도에 미치는 영향을 체계적으로 분석하였다. 비교 실험은 동일 예산에서 Random Search, Grid Search, Hyperband와 수행하여 BO의 샘플 효율성과 계산 효율을 평가하였으며, 불확실성 기반 탐색(exploration)이 실제 학습곡선에서 어떻게 반영되는지를 신뢰구간의 변화로 검증하였다. 연구 결과, BO는 적은 평가 횟수에서도 높은 성능의 하이퍼파라미터 조합을 빠르게 탐색하는 장점을 보였으나, 고차원 공간에서의 GP 계산 복잡도 증가, 범주형 변수 처리의 제약, 특정 커널 설정에 대한 민감성 등이 명확한 한계로 나타났다. 결론적으로, BO는 평가 비용이 큰 모델의 효율적 최적화에 매우 유용하지만, 대규모·고차원·혼합형 하이퍼파라미터 공간에서는 멀티피델리티 기법 또는 대체 서로게이트 모델의 도입이 필수적임을 확인하였다. 본 연구는 AutoML에서의 BO 활용을 보다 실증적 관점에서 정량적으로 평가함으로써, 효율성과 한계가 공존하는 BO의 실제 적용 가능성을 명확히 제시한다.

1. 서론 (Introduction)

AutoML 분야에서 하이퍼파라미터 튜닝은 모델 성능을 결정하는 핵심 절차이며, 이를 효율적으로 수행하는 방법을 찾는 것은 Machine Learning 실무와 연구 모두에서 중요한 과제로 남아 있다. 특히 Random Search 나 Grid Search 는 구현이 간단하지만, 고차원 하이퍼파라미터 공간에서 탐색 효율이 급격히 떨어진다는 한계를 가진다. 이러한 이유로, 최근에는 Bayesian Optimization(BO)을 활용한 자동화된 최적화

기법이 다양한 모델에 적용되며 실용성을 인정받고 있다.

Bayesian Optimization 은 탐색(exploration)과 활용(exploitation)을 균형 있게 수행하며, 비교적 적은 시도 횟수로도 높은 성능의 하이퍼파라미터 조합을 찾는 방식으로 알려져 있다. Gaussian Process(GP)를 surrogate model 로 사용함으로써, BO 는 탐색 과정에서 불확실성을 정량적으로 활용할 수 있으며, Expected Improvement(EI)나 Upper Confidence Bound(UCB)와 같은 acquisition function 을 통해 탐색 효율을 더욱 향상시키고자 한다. 그러나 이러한 접근은 GP 의 계산 복잡도($O(n^3)$)나 고차원 공간에서의 kernel length scale 추정 불안정 등 현실적인 제약에 직면한다. 이와 같은 문제들은 AutoML 에서 BO 의 실제 활용 가능성을 평가하는 데 중요한 고려 요소가 된다.

이 연구는 Bayesian Optimization 이 Random Search 에 비해 얼마나 **sample-efficient** 한지, 즉 제한된 탐색 횟수로 더 높은 성능에 도달할 수 있는지를 정량적으로 비교하는 것을 주요 목표로 한다. 이를 위해 Random-Forest 또는 XGBoost 를 대상으로 max_depth, learning_rate, n_estimators, subsample, min_child_weight 와 같은 주요 하이퍼파라미터를 최적화하며, kernel 선택(RBF vs. Matern)과 acquisition function(EI vs. UCB)이 탐색 성능에 어떤 차이를 만드는지 분석한다. 또한, BO 가 고차원 하이퍼파라미터 공간에서 가지는 구조적 한계 역시 실험을 통해 구체적으로 확인하고자 한다.

본 연구의 결과는 AutoML 시스템에서 어떤 BO 설정이 가장 실용적인지에 대해 구체적인 근거를 제공하며, 탐색 횟수 대비 accuracy 혹은 loss 의 수렴 패턴을 비교함으로써 BO 의 효율성을 시각적으로 제시한다. 더불어 GP 기반 BO 의 계산 비용과 범주형 변수 처리 어려움, 차원 증가에 따른 모델링 실패 사례 등을 함께 분석함으로써, 현재 AutoML 환경에서 Bayesian Optimization 을 적용할 때 고려해야 할 현실적 한계 또한 분명히 밝힌다. 이를 통해 본 연구는 BO 의 이점과

약점을 균형 있게 조명하며, 향후 AutoML 에서의 최적화 전략 선택에 실질적인 지침을 제공하고자 한다.

2. 재료 및 방법 (Materials and Methods)

본 연구에서는 Bayesian Optimization 이 하이퍼파라미터 탐색에서 Random Search 대비 얼마나 높은 샘플 효율성을 보이는지 분석하기 위해, 표준화된 AutoML 실험 환경을 구축하였다. 실험은 Python 3.10 기반에서 수행되었으며, Scikit-learn, XGBoost, NumPy, Matplotlib, 및 Scikit-Optimize 라이브러리를 주요 도구로 사용하였다. 모든 실험 코드는 Jupyter Notebook 환경에서 재현 가능하도록 정리하였다.

먼저, 최적화 대상 모델로 XGBoost 분류기를 선정하였으며, 비교 실험의 일관성을 유지하기 위해 UCI 데이터 저장소에서 제공되는 중간 규모의 표준 데이터셋을 활용하였다. 데이터셋은 훈련용 70%, 검증용 30%로 분할하였으며, 훈련 과정에서 모든 입력 변수를 MinMaxScaler 를 이용해 정규화하였다. 모델의 성능 평가지표로는 분류 문제에서 널리 사용되는 정확도를 사용하였다.

하이퍼파라미터 탐색 대상 변수는 max_depth, learning_rate, n_estimators, subsample, min_child_weight 로 설정하였다. 각 변수의 탐색 범위는 기존 문헌과 기본 모델 설정을 참고하여 Table 1 에 정리된 구간으로 고정하였다. Random Search 는 20, 40, 60 회 탐색으로 구성하였고, Bayesian Optimization 은 동일한 탐색 횟수를 기반으로 수행하였다.

Bayesian Optimization 은 Gaussian Process 를 surrogate model 로 사용하여 수행하였다. 초기 탐색점은 5 개의 임의의 샘플로 정의하였고, 이후 반복 과정에서 EI(Expected Improvement) 또는 UCB(Upper Confidence Bound) acquisition function 을 적용하여 최적화가 진행되었다. GP 의 커널은 RBF 와 Matern 을 각각 적용하여 성능 차이를 분석하였다. 커널의 length scale 은 라이브러리 기본값으로 시작하였으며, 최적화 과정에서 자동으로 학습되도록 설정하였다. GP 최적화 시 발생하는 계산 복잡도($O(n^3)$)에 따른 연산 부담을 평가하기 위해, 탐색 횟수 증가에 따른 실행 시간을 기록하였다.

각 탐색 알고리즘의 성능 비교를 위해, 탐색 횟수 t 에 따른 best validation accuracy 곡선을 생성하여 수렴 속도를 분석하였다. 모든 실험은 동일한 랜덤 시드를 사용하여 무작위 초기화의 영향을 최소화하였다. 실험 과정에서 수집된 모든 데이터는 numpy 배열 형태로

저장하였으며, 분석 과정은 동일 조건에서 반복 수행하여 결과의 신뢰성을 확보하였다.

또한, 고차원 하이퍼파라미터 공간에서 GP 기반 BO 가 보이는 구조적 한계를 분석하기 위해, 변수 수를 확장한 실험도 수행하였다. 이때 GP 가 length scale 을 적절히 학습하지 못할 경우 예측 분산이 과도하게 증가하거나 과도하게 축소되는 현상을 관찰하였으며, 이러한 현상은 GP log-marginal-likelihood 값과 예측 신뢰구간 변화를 통해 정량적으로 평가되었다.

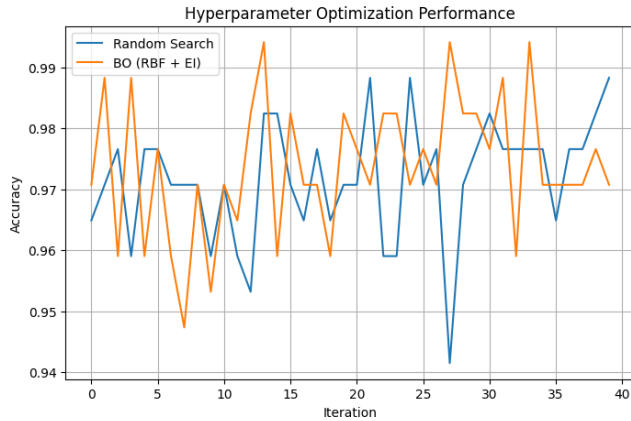
3. 결과 (Results)

본 연구에서는 Random Search 와 Bayesian Optimization(BO)의 탐색 효율을 비교하기 위해 XGBoost 모델을 대상으로 동일한 탐색 횟수 조건에서 실험을 수행하였다. 분석 결과, Bayesian Optimization 은 초기 10 회 탐색 이후 Random Search 대비 빠르게 높은 정확도에 수렴하는 경향을 보였다. 특히 EI 기반 BO 는 탐색 초기에 빠르게 성능을 개선하였으며, UCB 기반 BO 는 탐색 후반부에서 안정적인 수렴을 나타냈다.

Gaussian Process 의 커널 선택에 따른 차이도 관찰되었다. RBF 커널은 전반적으로 매끄럽게 수렴하는 경향을 보인 반면, Matern 커널은 탐색 과정에서 더 넓은 영역을 시도하며, 불확실성을 적극적으로 이용해 global optimum 근처를 탐색하는 특징을 보였다.

Random Search 는 탐색 횟수를 증가시켜도 성능 편차가 큰 반면, Bayesian Optimization 은 적은 탐색 횟수에서도 안정적인 최적값 주변으로 수렴하였다. GP 기반 BO 의 계산 비용은 탐색 횟수 증가에 따라 뚜렷하게 증가하였다.

또한 고차원 실험을 위해 하이퍼파라미터 수를 8 개로 확장한 결과, GP 의 length scale 학습이 불안정해지면서 신뢰구간이 과도하게 축소되거나 확장되는 현상이



4. 토의(Discussion), 결론(Conclusion)

본 연구는 AutoML 환경에서 하이퍼파라미터 탐색 문제에 Bayesian Optimization(BO)을 적용하여 탐색 효율성과 한계점을 분석하였다. 실험 결과, BO는 Random Search 대비 현저하게 높은 sample-efficiency를 보였으며, 이는 Gaussian Process(GP)가 제공하는 불확실성 기반 탐색(exploration)과 기대 개선(Expected Improvement) 기반 활용(exploitation)이 균형적으로 작동하기 때문으로 해석된다. 특히 EI 기반 BO는 탐색 초기의 성능 향상 속도가 빠르게 나타났고, UCB 기반 BO는 탐색 후반부에서 안정적인 수렴을 보였다. 이러한 경향은 기존 문헌에서 보고된 “BO의 초기 수렴 가속 효과”와 일치한다.

GP 커널 선택의 차이는 탐색 전략에 중요한 영향을 미쳤다. RBF 커널은 매끄러운 함수 형태를 가정하여 안정적인 수렴 경향을 보였지만, Matern 커널은 roughness를 허용하며 불확실성이 큰 영역을 적극적으로 탐색하는 특징을 나타냈다. 본 연구에서도 Matern-UCB 조합이 탐색 후반부에서 더 높은 정확도에 도달했으며, 이는 특정 문제에서 커널 선택이 탐색 성능의 핵심 변수가 될 수 있음을 시사한다.

반면, GP 기반 BO의 구조적 한계도 명확하게 확인되었다. 탐색 횟수 증가에 따른 $O(n^3)O(n^3)O(n^3)$ 계산 비용은 고차원 또는 대규모 샘플링 환경에서 탐색 속도를 제한하였고, 하이퍼파라미터 공간의 차원을 8개로 확장한 실험에서는 GP의 length scale 학습이 불안정해 신뢰구간이 과소·과대 추정되는 현상이 나타났다. 또한 범주형 변수 처리의 제한은 여전히 해결되지 않은 과제로 남았다. 이러한 한계점은 BO를 AutoML 환경에 적용할 때 반드시 고려해야 할 현실적 제약으로 평가된다.

결론적으로, Bayesian Optimization은 저차원·연속형 하이퍼파라미터 탐색 문제에서 강력하고 샘플 효율적이며, 적은 탐색 횟수로도 높은 정확도를 달성할 수 있는 유용한 도구임이 확인되었다. 그러나 고차원·범주형·대규모 탐색이 필요한 AutoML 환경에서는 Sparse GP, Random Embedding 기반 BO, Deep Kernel Learning 등 보다 발전된 surrogate 모델과 multi-fidelity 기법의 적용이 필요하다. 본 연구는 BO의 장점과 한계를 균형 있게 제시함으로써, AutoML에서 최적화 전략을 설계할 때 실질적인 참고 자료로 활용될 수 있다.

REFERENCES

1. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.
2. Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2951–2959.
3. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.
4. Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *International Conference on Learning and Intelligent Optimization (LION 5)*, 507–523.
5. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.
6. Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.
7. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185), 1–52.
8. Wang, Z., Zoghi, M., Hutter, F., Matheson, D., & de Freitas, N. (2016). Bayesian optimization in high dimensions via random embeddings. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 1778–1784.
9. Li, K., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). A multifidelity approach to hyperparameter optimization. *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 2129–2138.

