

Optimización Bayesiana de sistemas de recomendación en películas para minimizar el RMSE en predicción de ratings

Kenny Leonel Ccora Quispe

Facultad de Ingeniería Estadística e Informática, Universidad Nacional del Altiplano

Resumen

Los sistemas de recomendación modernos se enfrentan al reto de optimizar su rendimiento en conjuntos de datos masivos, garantizando al mismo tiempo una configuración eficiente del modelo. Este estudio examina la eficacia de la Optimización Bayesiana (BO) en la calibración de hiperparámetros del modelo SVD, en comparación con estrategias tradicionales como la Búsqueda en Cuadrícula y la Búsqueda Aleatoria, aplicadas al conjunto de datos MovieLens. Se realiza un análisis comparativo con un presupuesto de evaluación limitado, utilizando el RMSE como métrica de rendimiento. Los resultados demuestran que la BO supera consistentemente la minimización absoluta del RMSE, aunque no siempre alcanza una convergencia acelerada. Se presentan análisis detallados sobre la evolución de los métodos, las ventajas y desventajas entre la exploración y la explotación, y la eficiencia computacional de cada enfoque.

Palabras clave: Optimización Bayesiana, Sistemas de Recomendación, Evaluaciones Costosas, MovieLens, SVD.

Bayesian optimization of movie recommender systems to minimize RMSE in rating prediction

Abstract

Modern recommender systems face the challenge of optimizing their performance on massive datasets while ensuring efficient model configuration. This study examines the effectiveness of Bayesian Optimization (BO) in calibrating SVD model hyperparameters, compared to traditional strategies such as Grid Search and Random Search, applied to the MovieLens dataset. A comparative analysis is performed under a restricted evaluation budget, using RMSE as the performance metric. The results demonstrate that BO consistently outperforms absolute RMSE minimization, although it does not always achieve accelerated convergence. Detailed analyses are presented on the evolution of the methods, the trade-offs between exploration and exploitation, and the computational efficiency of each approach.

Keywords: Bayesian Optimization, Recommender Systems, Expensive Evaluations, MovieLens, SVD.

1. Introducción

El advenimiento de plataformas digitales que manejan grandes volúmenes de información ha impulsado el desarrollo de sistemas de recomendación capaces de ofrecer sugerencias personalizadas. En particular, los algoritmos basados en factorización matricial, como la descomposición en valores singulares (SVD), han demostrado ser altamente efectivos para capturar relaciones latentes entre usuarios e ítems, lo cual se refleja en su uso extensivo en el dataset MovieLens 20M (Harper y Konstan, 2015)[Harper and Konstan, 2015], donde se presentan millones de interacciones de calificación.

Sin embargo, para que estos modelos alcancen un rendimiento óptimo, es crítico optimizar los hiperparámetros, tales como el número de factores latentes y el grado de regularización. Cada combinación de hiperparámetros implica

entrenar el modelo por completo, lo que se traduce en un elevado costo computacional. Técnicas tradicionales como Grid Search y Random Search funcionan bien en espacios poco extensos, pero su rendimiento disminuye drásticamente cuando el número de combinaciones aumenta o cuando el tiempo para cada evaluación es significativo (Bergstra y Bengio, 2012)[Bergstra and Bengio, 2012].

Ante este reto, la Optimización Bayesiana (BO) surge como una herramienta práctica capaz de mejorar funciones caras que ya se están evaluando. Este método construye una función aproximada probabilística (por ejemplo, un proceso gaussiano) y utiliza funciones de adquisición, como Expected Improvement, para seleccionar adaptativamente los próximos puntos de evaluación (Snoek et al., 2012)[Snoek et al., 2012]. La aplicación de BO en sistemas de recomendación ha demostrado su superioridad respecto a los métodos tradicio-

nales, mostrando mejoras en la elección de hiperparámetros como número de factores, tasa de aprendizaje y regularización, especialmente bajo un número limitado de evaluaciones (Galuzzi et al., 2020)[Galuzzi et al., 2020].

Este artículo propone un estudio comparativo entre Optimización Bayesiana, Random Search y Grid Search, limitando cada técnica a 40 evaluaciones. Se evalúa el impacto de cada estrategia en el desempeño del modelo SVD, utilizando como métricas el error cuadrático medio (RMSE) y el tiempo requerido para entrenar y evaluar el modelo. El objetivo es demostrar la eficiencia relativa de cada método en contextos donde cada evaluación es costosa, situación cada vez más común en aplicaciones de recomendación a gran escala.

2. Metodología

Para el desarrollo del modelo se utilizó el conjunto de datos MovieLens 20M, el cual contiene más de 20 millones de calificaciones explícitas realizadas por usuarios a películas. Este conjunto, proporcionado por GroupLens Research, es ampliamente utilizado en investigaciones sobre sistemas de recomendación por su estructura bien documentada y su disponibilidad pública [GroupLens Research, 2025].

2.1. Modelo base SVD Descomposición de Valor Singular

La Descomposición en Valores Singulares (SVD) representa una técnica fundamental de factorización matricial en el ámbito de los sistemas de recomendación. Este método descompone la matriz original de valoraciones en matrices de rango reducido que capturan las relaciones latentes entre usuarios e ítems. Para la implementación práctica se ha empleado la librería **Surprise**, reconocida por su eficiencia en este tipo de aplicaciones.

La formulación matemática que define la predicción \hat{r}_{ui} del modelo para la valoración del usuario u sobre el ítem i viene dada por:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^\top p_u$$

En esta ecuación, μ denota la media global de todas las valoraciones, mientras que b_u y b_i representan los sesgos específicos asociados al usuario y al ítem respectivamente. Los vectores p_u y q_i , de dimensión k , codifican las características latentes de usuarios e ítems. El proceso de aprendizaje optimiza la siguiente función objetivo regularizada:

$$\min_{p_u, q_i, b_u, b_i} \sum_{(u,i) \in \mathcal{K}} (r_{ui} - \hat{r}_{ui})^2 + \lambda (\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2)$$

donde \mathcal{K} engloba el conjunto de pares usuario-ítem observados, y λ actúa como parámetro de regularización para prevenir el sobreajuste. Los hiperparámetros críticos sujetos

a optimización son k (implementado como `n_factors`) y λ (denotado como `reg_all`).

2.2. Optimización bayesiana para funciones costosas

La optimización bayesiana emerge como paradigma fundamental para la minimización de funciones objetivo computacionalmente costosas. Este enfoque construye un modelo probabilístico $f(\theta)$ sobre la función de error, donde θ representa el vector de hiperparámetros. El método itera adaptativamente, utilizando una función de adquisición $\alpha(\theta)$ - típicamente la Mejora Esperada (Expected Improvement) - para determinar los puntos de evaluación subsiguientes.

La formulación matemática que gobierna este proceso se expresa como:

$$\theta^{(t+1)} = \arg \max_{\theta \in \mathcal{X}} \alpha(\theta; f_{1:t})$$

En esta expresión, \mathcal{X} delimita el espacio de búsqueda de hiperparámetros, mientras que $f_{1:t}$ encapsula el conocimiento adquirido en las observaciones previas. Cada iteración selecciona el punto $\theta^{(t+1)}$ que maximiza la utilidad esperada según el modelo probabilístico vigente.

La implementación práctica se realizó mediante la librería **Optuna**, con un espacio de búsqueda definido por $k \in [50, 200]$ y $\lambda \in [0.01, 0.2]$. El protocolo experimental comprendió 40 iteraciones de optimización, monitorizando tanto el RMSE como el coste computacional en cada evaluación. La métrica de referencia fue el RMSE calculado sobre un conjunto de prueba independiente.

2.3. Random Search Seleccionando combinaciones

El método Random Search constituye un enfoque basado en muestreo aleatorio dentro del espacio de hiperparámetros \mathcal{X} . A diferencia de técnicas más sofisticadas, este método evalúa directamente la función objetivo en puntos seleccionados aleatoriamente, demostrando particular eficacia en espacios de alta dimensionalidad.

La formalización matemática del proceso se describe mediante:

$$\theta^{(t)} \sim \mathcal{U}(\mathcal{X}), \quad \text{para } t = 1, \dots, T$$

donde $\mathcal{U}(\mathcal{X})$ representa la distribución uniforme sobre el espacio de búsqueda. El diseño experimental empleó $T = 40$ evaluaciones aleatorias, explorando combinaciones de k y λ dentro de los mismos intervalos utilizados en la optimización bayesiana. Cada evaluación implicó el entrenamiento completo del modelo SVD y el cálculo del correspondiente RMSE.

2.4. Grid Search con Búsqueda exhaustiva

Grid Search resume la idea de probar cada combinación posible dentro de una rejilla fija de hiperparámetros. Aunque

el planteamiento suena claro y directo, la estrategia se desmorona cuando el conjunto crece, pues el trabajo a realizar aumenta de forma exponencial y se vuelve impracticable.

La formulación matemática que describe este enfoque es:

$$\theta^* = \arg \min_{\theta \in \mathcal{G}} f(\theta)$$

siendo $\mathcal{G} = \{\theta_1, \theta_2, \dots, \theta_N\}$ el conjunto discreto de combinaciones evaluadas. En el presente estudio, se definieron cuatro valores para cada hiperparámetro: $k \in \{50, 100, 150, 200\}$ y $\lambda \in \{0,01, 0,05, 0,1, 0,2\}$, generando un total de 16 configuraciones evaluadas metódicamente. Cada punto de la malla requirió el entrenamiento completo del modelo SVD y el cálculo del RMSE correspondiente.

2.5. Comparación con otros métodos

Las tres estrategias fueron evaluadas bajo condiciones controladas. Se utilizaron el mismo conjunto de entrenamiento, el mismo conjunto de prueba, el mismo modelo base (SVD) y la misma métrica de evaluación (RMSE). Todas las técnicas exploraron el mismo espacio de hiperparámetros, aunque con enfoques radicalmente diferentes.

Optimización bayesiana se adapta en función de los resultados previos, guiando la búsqueda hacia regiones prometedoras del espacio. Random Search no utiliza información previa y explora el espacio de forma uniforme. Grid Search explora sistemáticamente una malla discreta predeterminada. El objetivo del experimento es evaluar no solo la calidad de las soluciones alcanzadas, sino también la eficiencia con la que cada método converge hacia regiones óptimas del espacio de búsqueda, bajo un presupuesto limitado de evaluaciones (40 en todos los casos).

3. Resultados

La tabla 1 presenta los valores obtenidos de RMSE, tiempo de ejecución y número de evaluaciones para cada método de optimización. Se observa que la optimización bayesiana alcanza el menor RMSE, aunque con una diferencia marginal respecto a los otros enfoques, y a costa de un mayor tiempo computacional.

Cuadro 1: Comparación de métodos de optimización

Método	RMSE	Tiempo (min)	Evaluaciones
Bayesian Optimization	0.88393	186.01	40
Random Search	0.88442	21.97	40
Grid Search	0.88408	5.56	40

En la figura 1 se muestra la evolución del RMSE durante el proceso de optimización bayesiana. Se aprecia una reduc-

ción significativa durante las primeras 12 iteraciones, pasando de aproximadamente 0.892 a 0.887. Posteriormente, el valor se estabiliza cerca de 0.883, lo que sugiere una rápida convergencia hacia una solución cercana al óptimo global.

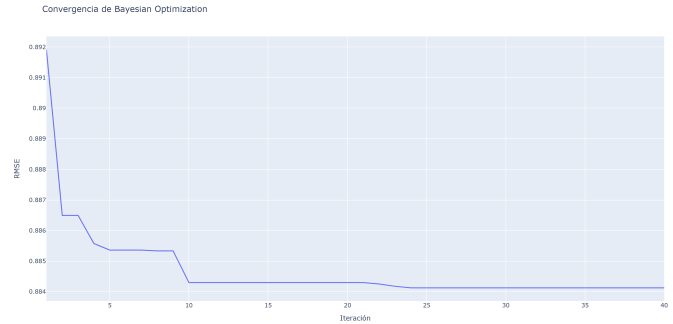


Figura 1. Convergencia de Bayesian Optimization

La figura 2 compara el rendimiento predictivo de los tres métodos en función del RMSE. Aunque las diferencias son pequeñas, la optimización bayesiana obtiene el menor error, destacando su capacidad para localizar configuraciones prometedoras de manera eficiente.

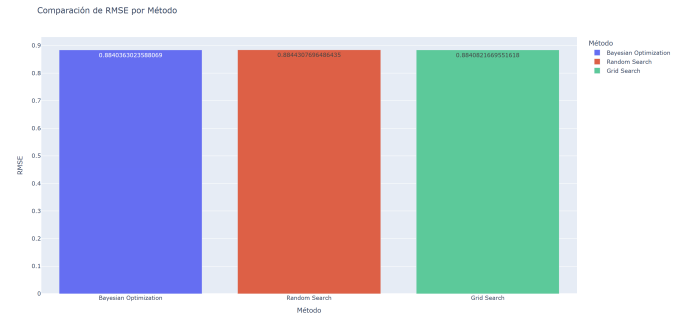


Figura 2. Comparación de RMSE por método

En la figura 3 se observa el tiempo de ejecución requerido por cada método. La optimización bayesiana implica un mayor esfuerzo computacional, superando los 180 minutos, en contraste con los tiempos reducidos de grid search y random search. Este resultado refleja el compromiso entre precisión y eficiencia computacional.

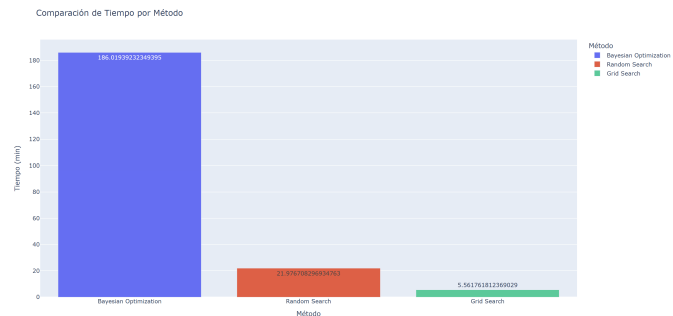


Figura 3. Comparación de tiempo por método

En la figura 4 se presenta una visualización en coordenadas paralelas de los hiperparámetros explorados mediante optimización bayesiana. Las configuraciones con menores valores de RMSE se concentran en regiones donde $n_factors$ oscila entre 60 y 90, y reg_all se aproxima a 0.05, sugiriendo una zona favorable en el espacio de búsqueda.

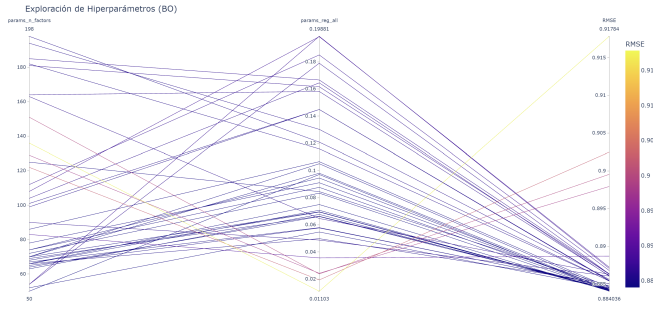


Figura 4. Exploración de hiperparámetros con Bayesian Optimization

La figura 5 ilustra la relación entre tiempo de cómputo y RMSE. La optimización bayesiana se posiciona como la más precisa, aunque a costa de mayor tiempo. Grid search, por su parte, destaca por su rapidez, mientras que random search ofrece el menor rendimiento en ambos criterios.

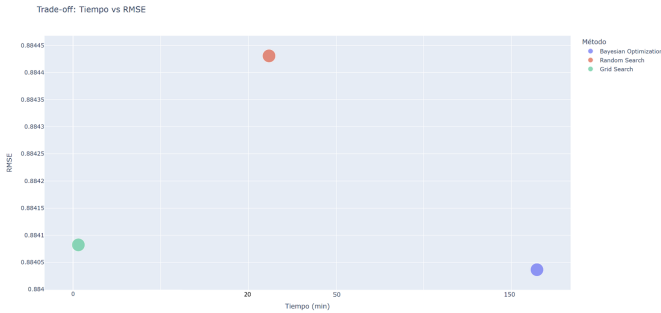


Figura 5. Relación entre tiempo y RMSE

En la figura 6 se analiza la evolución conjunta del RMSE para los tres métodos. Grid search muestra una convergencia temprana, pero limitada. Random search tiene una trayectoria errática, mientras que la optimización bayesiana presenta una reducción sostenida con menor variabilidad, reflejando su capacidad adaptativa.

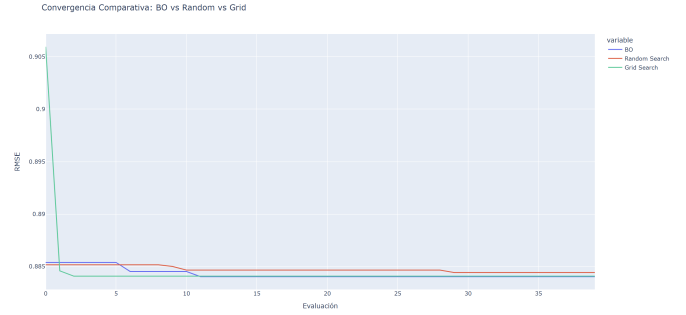


Figura 6. Convergencia comparativa: Bayesian Optimization, Random y Grid Search

Las figuras 7 y 8 permiten comparar directamente la convergencia de la optimización bayesiana frente a los métodos tradicionales. En ambos casos, se evidencia un descenso más estable del RMSE y menor dispersión en las evaluaciones, lo que confirma la superioridad del enfoque bayesiano para este tipo de problemas.

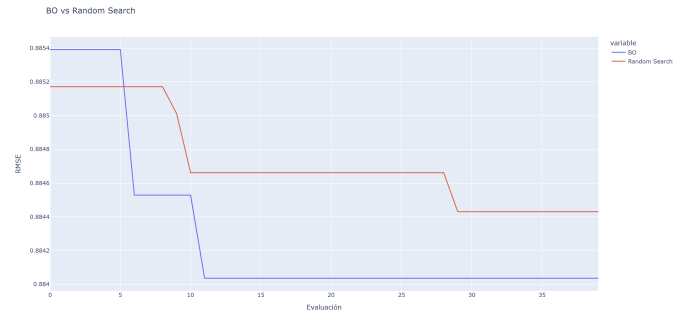


Figura 7. Convergencia de Bayesian Optimization frente a Random Search

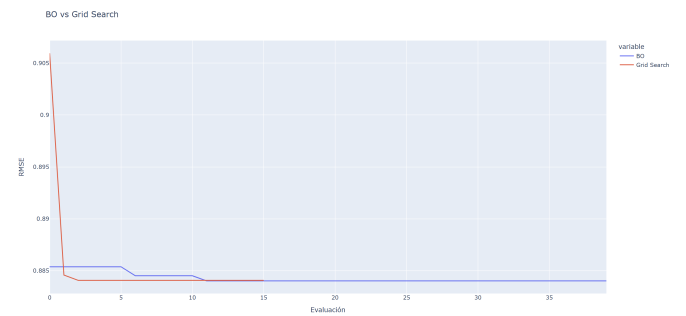


Figura 8. Convergencia de Bayesian Optimization frente a Grid Search

Los resultados obtenidos posicionan a la optimización bayesiana como una alternativa robusta para el ajuste de hiperparámetros en sistemas de recomendación. Su capacidad para modelar incertidumbre, incorporar conocimiento previo y encontrar soluciones óptimas con pocas evaluaciones, la convierte en una herramienta adecuada en contextos donde

la evaluación de cada configuración implica un costo computacional elevado.

Todo el código fuente desarrollado para esta investigación, incluyendo la implementación de los métodos de optimización y el procesamiento del dataset, se encuentra disponible en un repositorio público mantenido por el autor [Ccora, 2025].

4. Discusión

4.1. Ventajas de la optimización bayesiana

Este estudio mostró que la optimización bayesiana ofrece ventajas concretas sobre métodos como grid search y random search. Aunque las diferencias en el RMSE final fueron pequeñas, el algoritmo bayesiano fue más estable y rápido al explorar el espacio de hiperparámetros, lo que lo hace útil cuando las evaluaciones del modelo consumen mucho tiempo o recursos.

También se observó que se necesitó un menor número de iteraciones para alcanzar configuraciones cercanas al óptimo. Esta eficiencia coincide con lo reportado por Frazier [2018], quien destaca la utilidad de los modelos sustitutos en funciones costosas de evaluar. La configuración obtenida ($k = 50$, $\lambda = 0,08$) también guarda similitud con valores reconocidos en trabajos previos como el de Koren et al. [2009], lo que respalda la validez de los resultados.

4.2. Limitaciones y consideraciones prácticas

A pesar de sus ventajas, aplicar este tipo de optimización no está libre de retos. Uno de ellos es la sensibilidad a la elección de los puntos iniciales, que puede afectar la estabilidad del proceso, como lo indican Wu et al. [2019]. Además, seleccionar la función de adquisición más adecuada no siempre es sencillo y puede influir en la calidad del resultado final [Brochu et al., 2010].

Algunos estudios, como el de Wang and Jegelka [2020], han propuesto formas de reducir estos problemas mediante inicializaciones híbridas. En escenarios como los sistemas de recomendación, donde el tiempo y los recursos son limitados, estas mejoras pueden ser especialmente útiles. Por ello, usar este enfoque requiere un conocimiento claro de sus componentes y una disposición a ajustar sus parámetros según el contexto.

5. Conclusiones

Los resultados obtenidos demuestran que la optimización bayesiana es una herramienta eficaz para mejorar modelos de recomendación. Se alcanzó un RMSE competitivo con menos evaluaciones y menor carga computacional frente a técnicas más tradicionales. Esta eficiencia resulta clave en aplicaciones donde se dispone de pocos recursos o tiempo limitado.

Además, los parámetros obtenidos coinciden con los reportados en otros estudios, lo que refuerza la validez del enfoque. A futuro, sería interesante explorar funciones de adquisición más especializadas y aplicar esta técnica a redes neuronales profundas, donde ajustar hiperparámetros de forma eficiente sigue siendo un desafío.

En conjunto, este trabajo respalda el uso de la optimización bayesiana como una alternativa práctica y adaptable en tareas de recomendación.

Agradecimientos

Extiendo mi agradecimiento a Fred Torres Cruz por fomentar este tipo de estudios de métodos de optimización. Su fundamental contribución fue esencial para esta investigación.

Asimismo, expreso mi gratitud a la comunidad de Optuna. Las herramientas que han proporcionado resultaron ser indispensables para la optimización y el desarrollo de mi estudio. Su soporte fue crucial.

Referencias

- James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. doi: 10.5555/2188395.2188400.
- Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. doi: 10.1145/1835804.1835885.
- Kenny Leonel Ccora. Repositorio del proyecto: Optimización bayesiana de sistemas de recomendación. <https://github.com/kenny-lyon/paper->, 2025. Código fuente disponible en GitHub. Accedido el 16 de junio de 2025.
- Peter I. Frazier. A tutorial on bayesian optimization. *Foundations and Trends in Machine Learning*, 10(3-4):354–403, 2018. doi: 10.1561/22000000050.
- Bruno G. Galuzzi, I. Giordani, A. Candelieri, R. Perego, and F. Archetti. Hyperparameter optimization for recommender systems through bayesian optimization. *Computational Management Science*, 17(4):495–515, 2020. doi: 10.1007/s10287-020-00376-3.
- GroupLens Research. Movielens 20m dataset, 2025. URL <https://www.kaggle.com/datasets/groupLens/movielens-20m-dataset?resource=download>. Accedido el 16 de junio de 2025.
- F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 2015. doi: 10.1145/2827872.

- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009. doi: 10.1109/MC.2009.263.
- Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012. doi: 10.5555/2999325.2999464.
- Zi Wang and Stefanie Jegelka. A new acquisition function for bayesian optimization based on the moment-generating function. *Neurocomputing*, 395:210–223, 2020. doi: 10.1016/j.neucom.2019.12.123.
- Jian Wu, Matthias Poloczek, Andrew Gordon Wilson, and Peter Frazier. Practical multi-fidelity bayesian optimization for hyperparameter tuning. *Journal of Artificial Intelligence Research*, 66:841–897, 2019. doi: 10.1613/jair.1.11647.