Байесовские методы подбора гиперпараметров

Фетисов Григорий

4 июня 2026 г.

План Работы

1. Постановка задачи
2. Используемые данные и модели
3. Используемые методы
4. Результаты

Постановка задачи

Постановка задачи

Цель: Автоматический подбор гиперпараметров моделей машинного обучения с помощью байесовской оптимизации.

Проблема:

- Ручной подбор гиперпараметров – трудоёмкий и неоптимальный процесс.
- Полный перебор (Grid Search) требует огромных вычислительных ресурсов.
- Случайный поиск не учитывает результаты предыдущих итераций.

Зачем нужно исследование:

- Найти лучшие гиперпараметры за меньшее число итераций.
- Сравнить эффективность разных методов байесовской оптимизации (GP, TPE).
- Проверить, на каких типах данных и моделях байесовские методы дают наибольший выигрыш.

Используемые данные и модели

Используемые датасеты

Датасет	Задача	Объектов	Признаков	Типы признаков
Adult	Классиф.	48 842	14	Числ. + катег.
Bank Marketing	Классиф.	45 211	16	Числ. + катег.
Spambase	Классиф.	4 601	57	Числовые
California Housing	Регрессия	20 640	11	Числ. + катег.
Superconductivity	Регрессия	21 263	81	Числовые

Почему такой выбор:

- Разные размеры данных (от 4.6K до 48.8K объектов).
- Два типа задач: классификация (3) и регрессия (2).
- Различная природа признаков: смешанные, чисто числовые, с выбросами.
- Наличие пропусков требует предобработки.

Используемые модели

Тип	Классификация	Регрессия	Параметров
Линейные	LogisticRegression	Ridge, Lasso, ElasticNet	2–5
Деревья	RandomForestClassifier	RandomForestRegressor	5
Бустинг	CatBoostClassifier	CatBoostRegressor	6–8

Почему эти модели:

- **Разнообразие** Рассматриваем различные семейства моделей.
- **Быстрый процесс обучения** Позволяет делать эксперименты.
- **Универсальность** Несложно сделать единый пайплайн .

Метрики: ROC-AUC (классификация), MSE (регрессия).

Используемые методы

Гауссовский процесс — определение

Определение: Набор случайных величин $\{f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ называется **гауссовским процессом**, если для любого конечного набора точек $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ случайный вектор

$$\mathbf{f} = \big(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\big)^\top$$

имеет многомерное нормальное распределение:

$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}), \qquad \mu_i = m(\mathbf{x}_i), \quad K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j).$$

Обозначение:

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\cdot),\, k(\cdot,\cdot)\big)$$

где:

- $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ — **функция среднего** (часто $m \equiv 0$);
- $k(\mathbf{x}, \mathbf{x}')$ — **ядро**, выбранная функция, задающая $\mathrm{Cov}\left(f(\mathbf{x}), f(\mathbf{x}')\right) = k(\mathbf{x}, \mathbf{x}')$.

Априорное распределение и зашумлённые наблюдения

Априорное распределение: Ещё не видя данных, для любого конечного множества $\mathbf{X}$ имеем:

$$p(\mathbf{f} \mid \mathbf{X}) = \mathcal{N}\big(\mathbf{f} \mid \mathbf{m}, \mathbf{K}_{XX}\big)$$

где $\mathbf{m} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n))^\top, \quad (\mathbf{K}_{XX})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Наблюдения с шумом: В задаче подбора гиперпараметров мы видим зашумлённые значения целевой функции:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$$

Тогда ковариационная матрица наблюдаемых данных:

$$\mathrm{Cov}(y_i, y_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 \delta_{ij} \quad \Longleftrightarrow \quad \mathbf{K}_y = \mathbf{K}_{XX} + \sigma_n^2 \mathbf{I}$$

σ_n^2 — дисперсия шума наблюдений (настраиваемый гиперпараметр ГП).

Совместное распределение и апостериорный вывод

Задача: даны обучающие точки $(\mathbf{X}, \mathbf{y})$, нужно предсказать $\mathbf{f}_*$ в новых точках $\mathbf{X}_*$.

Совместное априорное распределение:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{m}_* \end{pmatrix}, \begin{pmatrix} \mathbf{K}_{XX} + \sigma_n^2 \mathbf{I} & \mathbf{K}_{X*} \\ \mathbf{K}_{*X} & \mathbf{K}_{**} \end{pmatrix} \right)$$

Апостериорное распределение (обусловливание многомерного нормального):

$$\mathbf{f}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

$$\boldsymbol{\mu}_* = \mathbf{m}_* + \mathbf{K}_{*X}(\mathbf{K}_{XX} + \sigma_n^2 \mathbf{I})^{-1}(\mathbf{y} - \mathbf{m})$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_{*X}(\mathbf{K}_{XX} + \sigma_n^2 \mathbf{I})^{-1}\mathbf{K}_{X*}$$

Вычислительная сложность: $O(n^3)$ из-за обращения матрицы $n \times n$.

Апостериорное предсказание для одной точки

Для одной тестовой точки $\mathbf{x}_*$:

Предсказанное среднее (наилучшая оценка $f(\mathbf{x}_*)$):

$$\mu(\mathbf{x}_*) = m(\mathbf{x}_*) + \mathbf{k}_*^\top (\mathbf{K}_{XX} + \sigma_n^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{m})$$

Предсказанная дисперсия (неопределённость прогноза):

$$\sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (\mathbf{K}_{XX} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*$$

где $\mathbf{k}_* = \big(k(\mathbf{x}_1, \mathbf{x}_*), \ldots, k(\mathbf{x}_n, \mathbf{x}_*)\big)^\top$.

Интерпретация:

- $\mu(\mathbf{x}_*)$ — наша лучшая догадка о значении функции в новой точке;
- $\sigma^2(\mathbf{x}_*)$ **убывает** вблизи точек наблюдений, **растёт** вдали от них.

Ядро (ковариационная функция) и его гиперпараметры

RBF (гауссово) ядро — бесконечно гладкие функции:

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right)$$

Matérn-ядро ($\nu = 5/2$) — функции конечной гладкости:

$$k_{\text{Matérn}}(r) = \sigma_f^2 \left(1 + \frac{\sqrt{5}\, r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}\, r}{\ell}\right), \quad r = \|\mathbf{x} - \mathbf{x}'\|$$

Гиперпараметры ядра (настраиваются по данным):

- ℓ — **длина шкалы**: на каком расстоянии точки «влияют» друг на друга;
- σ_f^2 — **сигнальная дисперсия**: масштаб вариации функции;

TPE в Optuna

Идея: В отличие от GP, TPE не строит суррогатную модель целевой функции, а **моделирует плотности распределения параметров**.

Процедура:

1. Фиксируется порог $\gamma \in (0, 1)$ (обычно $\gamma = 0.25$).
2. Наблюдения разделяются на две группы:
   - **«Хорошие»** (D_{good}) – лучшие $100\gamma\%$ результатов.
   - **«Плохие»** (D_{bad}) – остальные.
3. Для каждой группы строятся оценки плотности параметров:

$$p(\mathbf{x} \mid y > y^*) \approx \ell(\mathbf{x}), \qquad p(\mathbf{x} \mid y \leq y^*) \approx g(\mathbf{x})$$

Интуиция: TPE ищет области, где **много хороших** точек (ℓ велико) и **мало плохих** (g мало).

Оценка плотностей: Parzen-окна

Плотности $\ell(\mathbf{x})$ и $g(\mathbf{x})$ оцениваются с помощью **одномерных ядерных оценок** (Parzen-окна) для каждого параметра независимо.

Для параметра x_i с наблюдениями $\{x_i^{(1)}, \ldots, x_i^{(n)}\}$:

$$\hat{p}(x_i) = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{x_i - x_i^{(j)}}{h}\right)$$

где K – ядро (обычно гауссово), h – ширина окна.

Для непрерывных параметров используется усечённое гауссово ядро. **Для дискретных** – сглаженное категориальное распределение.

Это позволяет TPE естественно работать со **смешанными** и **условными** пространствами гиперпараметров.

Оценка плотностей: Parzen-окна

Плотности $\ell(\mathbf{x})$ и $g(\mathbf{x})$ оцениваются с помощью **одномерных ядерных оценок** (Parzen-окна) для каждого параметра независимо.

Для параметра x_i с наблюдениями $\{x_i^{(1)}, \ldots, x_i^{(n)}\}$:

$$\hat{p}(x_i) = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{x_i - x_i^{(j)}}{h}\right)$$

где K – ядро (обычно гауссово), h – ширина окна.

Для непрерывных параметров используется усечённое гауссово ядро. **Для дискретных** – сглаженное категориальное распределение.

Это позволяет TPE естественно работать со **смешанными** и **условными** пространствами гиперпараметров.

Результаты

Результаты классификации (ROC-AUC)

Модель	Датасет	GP	TPE	Random	GP vs Rand	TPE vs Rand
CatBoost	Adult	0.9290	0.9292	0.9286	**+0.05%**	**+0.07%**
LogisticRegression	Adult	0.9057	0.9063	0.9061	*-0.05%*	**+0.02%**
RandomForest	Adult	0.9184	0.9185	0.9184	0.00%	**+0.01%**
CatBoost	Bank	0.8070	0.8071	0.8065	**+0.06%**	**+0.07%**
LogisticRegression	Bank	0.7713	0.7712	0.7712	**+0.02%**	0.00%
RandomForest	Bank	0.7963	0.7966	0.7965	*-0.02%*	**+0.02%**
CatBoost	Spam	0.9887	0.9883	0.9880	**+0.07%**	**+0.03%**
LogisticRegression	Spam	0.9723	0.9724	0.9723	0.00%	0.00%
RandomForest	Spam	0.9852	0.9842	0.9838	**+0.15%**	**+0.04%**

Таблица: Сравнение методов оптимизации на задачах классификации.
Жирным отмечен положительный прирост над случайным поиском, *курсивом* — отрицательный.

Краткий вывод

- Все три метода дают очень близкие результаты; прирост минимален (до $+0.15\%$).

Результаты регрессии (neg MSE)

Модель	Датасет	GP	TPE	Random	GP vs Rand	TPE vs Rand
California Housing (MSE $\times 10^9$)						
CatBoost	California	-1.99	-2.01	-2.06	**+3.06%**	**+2.11%**
ElasticNet	California	-4.77	-4.77	-4.77	**+0.01%**	**+0.01%**
Lasso	California	-4.77	-4.77	-4.77	0.00%	0.00%
RandomForest	California	-2.42	-2.44	-2.52	**+3.71%**	**+3.02%**
Ridge	California	-4.77	-4.77	-4.77	*−0.00%*	0.00%
Superconductivity (MSE)						
CatBoost	Supercond.	-89.68	-90.87	-91.42	**+1.91%**	**+0.61%**
ElasticNet	Supercond.	-312.77	-313.21	-313.70	**+0.30%**	**+0.16%**
Lasso	Supercond.	-311.94	-311.96	-311.96	**+0.01%**	0.00%
RandomForest	Supercond.	-90.30	-92.12	-93.62	**+3.54%**	**+1.60%**
Ridge	Supercond.	-311.49	-311.47	-311.47	*−0.00%*	0.00%

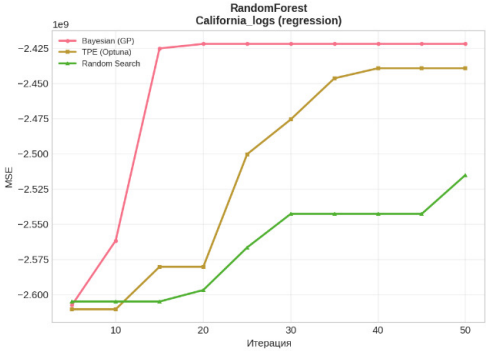
Таблица: Сравнение методов оптимизации на задачах регрессии.
Жирным отмечен положительный прирост над случайным поиском, *курсивом* — отрицательный.
Приведенные результаты получены после **50** итераций подбора параметров

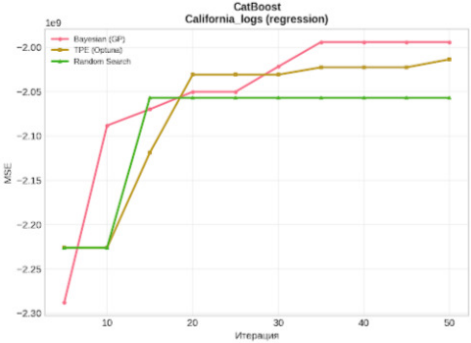
.

Краткие выводы для регрессии

- **Значимый выигрыш** (до +3.71%) только для ансамблевых моделей (RandomForest, CatBoost).
  - *Причина:* сложный многомерный ландшафт гиперпараметров (глубина, число деревьев, регуляризация) требует интеллектуального исследования.
- **GP уверенно опережает TPE** (напр., RandomForest на California: +3.71% против +3.02%).
  - *Причина:* гауссовский процесс строит более точную глобальную суррогатную модель, лучше улавливая взаимодействия параметров. TPE с одномерными оценками плотности сходится медленнее при ограниченном бюджете итераций.
- **Линейные модели** (Ridge, Lasso, ElasticNet) практически не улучшаются.
  - *Причина:* их пространство гиперпараметров практически одномерно (α), ландшафт «плоский» вблизи оптимума. Даже случайный поиск быстро его находит.

Графики сходимости





Только дискретные параметры (RandomForestRegressor, California)

Метод	Лучший neg_MSE	Прирост над Random
GP (EI)	-2.530×10^9	+0.20%
TPE	$\mathbf{-2.518 \times 10^9}$	**+0.68%**
Random	-2.535×10^9	—

Таблица: Пространство: `n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_features`.

Интерпретация

- **TPE вышел на первое место** с заметным отрывом (+0.68% над Random).
- GP показал скромное улучшение над случайным поиском (+0.20%).
- При чисто дискретном пространстве преимущество гауссовского процесса исчезает — TPE работает точнее.

beginframeЭволюция лидеров при сужении пространства

Тип пространства	Лучший метод
Полное (непрерывные + дискретные)	GP
Фиксированы непрерывные (оставлены дискретные)	TPE

Причины смены лидера

- **Гауссовский процесс** силён в непрерывных областях — он строит гладкую суррогатную модель и точно предсказывает улучшения.
- Когда остаются только дискретные параметры, гладкая модель GP становится неточной, и оптимизация сводится к исследованию небольшого числа комбинаций.
- **TPE**, напротив, изначально использует одномерные оценки плотности и лучше адаптирован к дискретным и условным переменным.

Исходный код проекта

Репозиторий на GitHub

https://github.com/Fetis789/OptimizationProjectBayes

В репозитории: ноутбуки с полным пайплайном, логи всех экспериментов,
графики динамики оптимизации и итоговые таблицы.