



Politechnika Warszawska

Wydział Matematyki i Nauk Informacyjnych

Tunowalność hiperparametrów - Raport z projektu

Mateusz Brochocki
Aleksander Karch
Adrian Krzyżanowski

Warszawa, Polska

17 listopada 2025

1 Wstęp

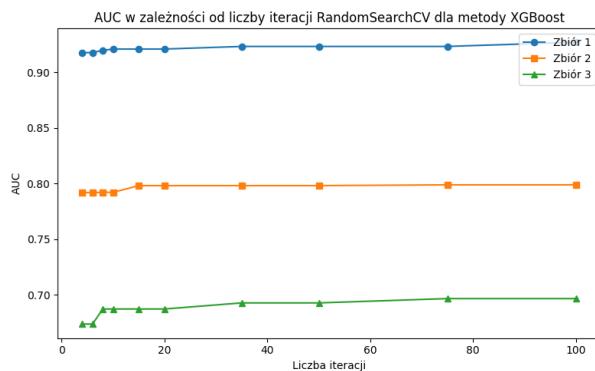
Celem niniejszej pracy jest przeanalizowanie tunowalności trzech algorytmów uczenia maszynowego, to jest regresji logistycznej z regularyzacją `elasticnet`, metody k-najbliższych sąsiadów oraz `XGboost'a` na piętnastu różnych średnie wielkości zbiorach danych, zawierających od kilkuset do kilku tysięcy obserwacji oraz mających od kilku do kilkudziesięciu zmiennych wyjaśniających. Do tunowania modeli została użyta metoda `RandomSearchCV` oraz optymalizacja bayesowska z pakietu `optuna`.

2 Liczba iteracji metody a stabilność optymalizacji

Na początku została przez nas sprawdzona stabilność optymalizacji ze względu na liczbę iteracji używanych metod. Z powodu ograniczonego miejsca w raporcie, przedstawimy wyniki uzyskane dla modelu `XGboost`, wyniki dla pozostałych modeli zostały uzyskane analogicznie.

Przeszukiwanie losowe

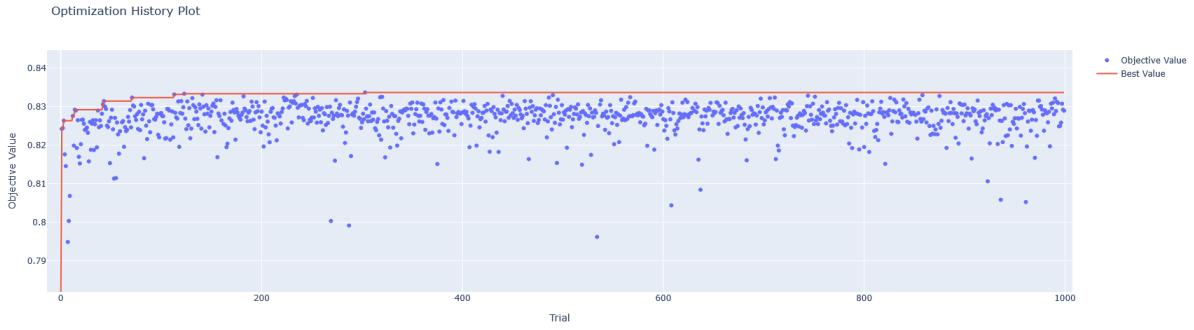
Na pierwszym wykresie możemy zauważyc, że stabilność optymalizacji uzyskujemy bardzo szybko - już po około 40-50 iteracjach nie obserwujemy poprawy jakości predykcyjnej modelu. W związku z tym w badanym dalej problemie dla każdego modelu używamy 50 iteracji `RandomSearchCV`.



Rysunek 1: Zależność uzyskanej miary AUC od ilości iteracji metody `RandomSearchCV`.

Optymalizacja bayesowska

Do badania stabilności optymalizacji wykorzystaliśmy wykresy generowane przez pakiet `optuna`. Możemy zauważyc na poniższym przykładzie (zbiór 2), że stabilność uzyskujemy wolniej niż w przypadku przeszukiwania losowego - w ciągu około 100-120 iteracji. Dalej jednak obserwujemy delikatną poprawę mocy predykcyjnej modelu - w przypadku `XGboost'a` następuje nawet po 300 iteracjach. W związku z tym w dalszych rozważaniach zdecydowaliśmy się na kompromis pomiędzy mocą predykcyjną modeli, względem których badamy tunowalność, a czasem i rozsądkiem. Z tego powodu dla regresji logistycznej będziemy wykonywali 200 iteracji, dla metody k-najbliższych sąsiadów 300, a dla `XGboost'a` - 400 iteracji.



Rysunek 2: Linia na wykresie obrazuje zależność uzyskanej miary AUC względem ilości wykonanych prób optymalizacji.

3 Zakresy hiperparametrów

Tabela 1: W tabeli zostały przedstawione zakresy badanych hiperparametrów użytych podczas tunowania poszczególnych algorytmów z pominięciem dyskretnego hiperparametru *weights* dla *knn*. W dwóch ostatnich kolumnach są optymalne zakresy do tunowania wyznaczone za pomocą dwóch rozważanych metod samplingu.

Algorytm	Hiperparametr	Typ	Rozważany zakres	Optymalny zakres przeszukiwanie losowe	Optymalny zakres opt. bayesowska
<i>elasticnet</i>	C	float	[0, 10]	[0.19, 5.79]	[0.01, 5.5]
	l1_ratio	float	[0, 1]	[0.07, 0.93]	[0, 0.98]
<i>knn</i>	n_neighbors	integer	[1, 40]	[14, 40]	[17.1, 38.3]
	p	float	[1, 6]	[1.02, 5.4]	[1, 4.14]
<i>xgboost</i>	learning_rate	float	[0, 1]	[0.07, 0.76]	[0.03, 0.88]
	n_estimators	integer	[10, 5000]	[828, 4225]	[410, 4837]
	subsample	float	[0, 1]	[0.24, 0.82]	[0.19, 0.95]
	gamma	float	[0, 1]	[0.06, 0.87]	[0.04, 0.88]
	reg_alpha	float	[0, 10]	[0.78, 8.44]	[0.01, 8.09]
	reg_lambda	float	[0, 40]	[4.8, 38.3]	[0.75, 36.2]
	max_depth	integer	[1, 20]	[1, 17.6]	[1.7, 19.3]

Optymalne zakresy hiperparametrów zostały wyznaczone na podstawie dolnego (5%) i górnego (95%) kwantyla rozkładu wcześniej wyznaczonych najlepszych wartości hiperparametrów dla każdego ze zbiorów. Na podstawie wyników można wywnioskować, że zakres hiperparametru C z regresji logistycznej można by skrócić do 5.8 (lub 5.5), gdyż znaczna większość najlepszych wartości hiperparametrów mieściła się w przedziale do tej wartości. Podobnie nie opłaca się rozważać liczb siedmiu mniejszej niż 14 (lub 17) w *knn*. Dalej, przedział *n_estimators* mógłby zaczynać się od większej wartości. Co do zakresu pozostałych hiperparametrów, to można nałożyć drobne poprawki.

4 Tunowalność algorytmów

W kolejnej części zbadaliśmy tunowalność każdego z trzech algorytmów. W tym celu dla każdego z piętnastu wybranych zbiorów wyznaczyliśmy najlepsze możliwe wartości parametrów z

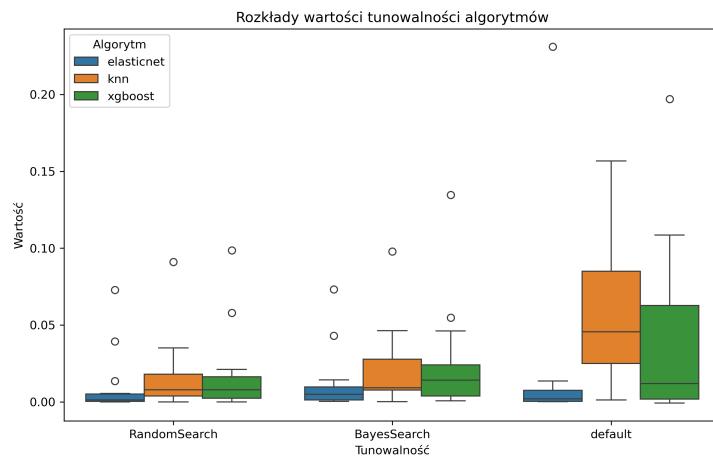
zakresu podanego w rozdziale 3. Do losowania punktów użyliśmy metody przeszukiwania losowego - `RandomSearchCV` oraz optymalizacji Bayesowskiej. Jako metrykę przyjęliśmy miarę AUC. Po otrzymaniu wyników policzyliśmy tunowalność algorytmów na każdym ze zbiorów, czyli różnicę między miarą AUC dla najlepszych hiperparametrów na danym zbiorze, a miarą AUC dla wyznaczonych domyślnych najlepszych hiperparametrów dla wszystkich zbiorów. Oprócz tunowalności otrzymanej dwiema różnymi metodami wyznaczania najlepszych hiperparametrów policzyliśmy również tunowalność odnoszącą się do hiperparametrów otrzymanych techniką przeszukiwania losowego dla danego zbioru, a domyślnych systemowych hiperparametrów. Uśrednione wyniki z odchyleniem standardowym dla średniej ($\pm SEM$) dla każdego z algorytmów zostały przedstawione w poniższej tabeli.

Tabela 2: *Tunowalność algorytmów. ostatnia kolumna przedstawia różnicę między miarą AUC dla najlepszych hiperparametrów wyznaczonych za pomocą RandomSearchCV, a miarą AUC dla domyślnych, systemowych hiperparametrów.*

Algorytm	Tunowalność <code>RandomSearch</code>	Tunowalność opt. bayesówka	Tunowalność default
elasticnet	0.00955 ± 0.00523	0.012 ± 0.00517	0.0189 ± 0.0152
knn	0.0162 ± 0.00611	0.0205 ± 0.00656	0.0601 ± 0.0119
xgboost	0.0169 ± 0.00695	0.0246 ± 0.00887	0.042 ± 0.0146

Na podstawie tabeli można wywnioskować, że algorytmy `knn` oraz `XGboost` są bardziej tunowalne niż regresja logistyczna z `elasticnet`. Oznacza to, że dla tych algorytmów bardziej opłaca się dostrajać hiperparametry, a nie korzystać z najlepszych domyślnych w celu poprawienia miary AUC. Ponadto widać różnicę pomiędzy tunowalnością w przypadku domyślnych hiperparametrów (default), a najlepszych wyznaczonych (`RandomSearchCV`). Sugeruje to, że korzystanie z najlepszych wyznaczonych hiperparametrów dla wszystkich zbiorów może być całkiem niezłą alternatywą dla każdorazowego tunowania dla nowego rozważanego zbioru. Miara AUC nie poprawi się aż tak bardzo jak w przypadku tunowania względem domyślnych systemowych hiperparametrów, a można zaoszczędzić na czasie.

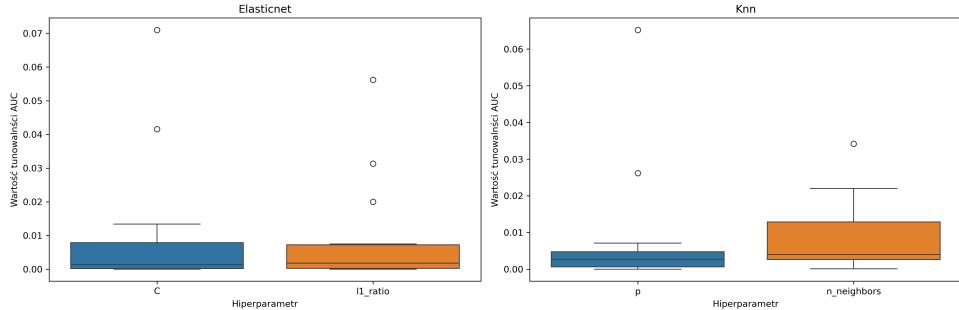
Rozkłady wartości tunowalności można zaobserwować na rysunku (3). Można zauważyć, że pojawia się nieco outlierów, więc mediana wyników lepiej sprawdziłaby się niż średnia, jednak wnioski pozostają te same, to znaczy regresja logistyczna z `elasticnet` cechuje się najmniejszą tunowalnością, a rozkład wartości tunowalności w przypadku defaultowym jest bardziej rozrzucony od pozostałych przypadków.



Rysunek 3: Boxploty rozkładów wartości każdej tunowalności dla poszczególnych algorytmów.

5 Tunowalność poszczególnych hiperparametrów

W kolejnej części policzyliśmy tunowalność poszczególnych hiperparametrów. Z powodu ograniczonego miejsca zbadaliśmy tylko dwa algorytmy - regresję logistyczną z `elasticnet` i `knn`. Wyjątkowo ograniczyliśmy się też tylko do metody `RandomSearchCV`. Rozkłady wartości tunowalności względem miary AUC zostały umieszczone na rysunku (4).



Rysunek 4: Boxploty rozkładów wartości tunowalności dla poszczególnych hiperparametrów.

W przypadku algorytmu `knn` można zauważyc, że tunując hiperparametr `n_neighbors` możemy uzyskać lepsze rezultaty jeśli chodzi o wartość miary AUC niż tunując hiperparametr `p`. Ponadto tunowalność `p` jest niewielka, więc wartość `p` ma niewielki wpływ na poprawę miary AUC (pomijając dwa zbiory). Natomiast w przypadku regresji logistycznej z `elasticnet` rozkład hiperparametru `C` jest skupiony blisko zera (mała mediana), więc tunowanie `C` nie wpłynie znacząco na poprawę AUC, choć mogą zdarzyć się przypadki, kiedy wpływ będzie zauważalny. Dla `l1_ratio` sytuacja wygląda podobnie, jednak tutaj rozrzut wartości jest trochę mniejszy. Jest to spójne z naszym poprzednim wnioskiem, że `elasticnet` cechuje najmniejszą tunowalność.

6 Wpływ techniki losowania wartości hiperparametrów na tunowalność algorytmów

Postanowiliśmy sprawdzić w sposób formalny, czy technika losowania wartości hiperparametrów wpływa istotnie na samą tunowalność. Niech X, Y oznaczają wektor tunowalności algorytmu względem odpowiednio optymalizacji bayesowskiej oraz przeszukiwania losowego. Zakładamy, że obydwie próbki pochodzą z nieznanych nam rozkładów o medianach M_X, M_Y . Interesuje nas weryfikacja hipotezy

$$\begin{cases} \mathcal{H}_0 : M_X = M_Y, \\ \mathcal{H}_1 : M_X > M_Y. \end{cases}$$

W tym celu posłużyliśmy się testem znakowanych rang Wilcooxona. Wyniki prezentujemy w poniższej tabelce:

Rozważany model	Regresja logistyczna	k-najbliższych sąsiadów	XGBoost
p-wartość testu	0.00054697	0.00084814	0.00194024

Tabela 3: Tabela przedstawiająca wyniki testu Wilcooxona dla poszczególnych modeli

Widzimy w szczególności, że p-wartości te są bardzo małe, co skłania nas do odrzucenia hipotezy zerowej i wyciągnięcia wniosku, że optymalizacja bayesowska daje nam lepszą możliwość tunowania hiperparametrów od przeszukiwania losowego.