Warszawa, 19 stycznia 2019

Metody ewolucyjne i uczenie się maszyn

Projekt 2
Dokumentacja końcowa

Na jednym z benchmarków dot. modelowania szeregów czasowych (np. NN3 lub M3) porównać metodę xgboost z modelem liniowym.

Prowadzący:
dr inż. Paweł Zawistowski

Wykonali:
Michał Kruszewski, Łukasz Blaszka

Spis treści

1. Cel

Celem projektu było praktyczne poznanie dowolnie wybranego modelu liniowego oraz metody xgboost, stosowanych do modelowania szeregów czasowych. Jako model liniowy wybrany został popularny model ARIMA (*ang. AutoRegressive Integrated Moving Average*). Realizacja celu oparta została na porównaniu obu metod na benchmark'u NN3.

2. Założenia

W specyfikacji wstępnej projektu znalazły się następujące założenia:

- Zadanie rozwiązane miało zostać w języku *R*.
- Jako benchmark testowy wybrany został nowszy benchmark NN3. Zawiera on mniej szeregów czasowych niż benchmark M3 (111 vs 3003), dlatego lepiej nadaje się do celów dydaktycznych (subiektywna ocena osób realizujących projekt).
- Jako model liniowy wybrany miał zostać popularny model ARIMA (*ang. AutoRegressive Integrated Moving Average*), uznawany za jedną z opcji do modelowania szeregów niestacjonarnych.
- W celu dopasowania najlepszego modelu ARIMA wykorzystana miała zostać metoda *auto.arima* pochodząca z pakietu *forecast*.
- W celu predykcji metodą xgboost wykorzystana miała zostać biblioteka *xgboost* (`https://github.com/dmlc/xgboost`).
- Ze względu na brak jakiegokolwiek doświadczenia z metodą xgboost, oraz liczbę jej parametrów, sposób optymalizacji określony miał zostać na etapie implementacji.

3. Krótki wstęp teoretyczny

3.1. ARIMA

Model ARIMA jest rozszerzeniem modelu ARMA, który poddawany jest różnicowaniu w celu usunięcia trendu. Model ARMA(p,q) dla stacjonarnego szeregu czasowego z_t ma postać:

$$z_t = \phi_1 z_{t-1} + ... + \phi_p z_{t-p} + w_t + \psi_1 w_{t-1} + ... + \psi_q w_{t-q} \tag{1}$$

gdzie z_t jest szeregiem stacjonarnym o zerowej średniej, $\phi_i, \psi_i \in \mathbb{R}(\phi_p \neq 0, \psi_q \neq 0)$, a $w_i \sim N(0, \sigma^2)$.

Model ARMA jest połączeniem dwóch modeli:

- Model AR(p) - proces autoregresyjny p rzędu określa wpływ p poprzednich obserwacji na wartość obecną z_t. Przy wyliczaniu modelu AR poszukiwane są wartości $(\phi_1, ..., \phi_p)$ by spełnić równanie[2]:

$$z_t = c + \phi_1 z_{t-1} + \phi_2 z_{t-2} + ... + \phi_p z_{t-p} + w \tag{2}$$

- Model MA(q) - średnia ruchoma q rzędu opisuje zależność obserwacji z_t od aktualnego zakłócenia w_t oraz q poprzednich. Poszukiwane są wartości $(\psi_1, ..., \psi_q)$ przy znanych wartościach zakłóceń $(w_1, ..., w_t)$. Wzór modelu MA[2]:

$$z_t = c + \psi_1 w_{t-1} + \psi_2 w_{t-2} + ... + \psi_q w_{t-q} + w_t \tag{3}$$

Model ARIMA do usuwania trendu wykonuje operację różnicowania szerego gdzie k to opóźnienie (lag). Wzór różnicowania szeregu[2]:

$$z'_t = z_t - z_{t-k} \tag{4}$$

Ostatecznie model ARIMA ma postac[2]:

$$z'_t = c + \phi_1 z'_{t-1} + ... + \phi_p z'_{t-p} + w_t + \psi_1 w_{t-1} + ... + \psi_q w_{t-q} \tag{5}$$

3.2. Metoda xgboost

Metoda xgboost (*Extreme Gradient Boosting*) wywodzi się z metody GBM (*ang. Gradient Boosting Machine*). Obie te metody opierają się o gradientowe wzmacnianie drzew (*ang. gradient boosted trees*). U ich podstaw znajduje się model zespołu drzew decyzyjnych (*ang. decision tree ensembles*). W zespole takim pojedyncze drzewo zazwyczaj słabo radzi sobie z klasyfikacją/regresją, dlatego do predykcji wykorzystywany jest cały zespół drzew. Idea wzmacniania gradientowego polega na tworzeniu ciągu prostych drzew, z których każde kolejne jest zbudowane do predykcji reszt generowanych przez poprzednie.

Różnica w idei działania xgboost a GDM jest niewielka. xgboost używa bardziej regularnej formalizacji modelu w celu kontroli przeuczenia (*over-fitting*). Dodatkowo xgboost jest mocno zoptymalizowany w obszarach szybkości obliczeń oraz zużycia pamięci.

Trzeba dodać informację co jest x a co jest y podczas predykcji w XGBoost.

4. Zbiór danych

Liczba szeregów czasowych znajdujących się w benchmark'u NN3 wynosi 111. Długość wektora testowego dla każdego z szeregów wynosi 18. Zbiór danych jest zaczerpnięty z jednorodnej populacji empirycznych biznesowych szeregów czasowych. Wśród danych dominują szeregi, które wydają się być mocno zaszumione (rys. 1a, 1b). Znaleźć tam również można szeregi czasowe z wyraźnym trendem (rys. 1c) lub wahaniami okresowymi (rys. 1d).

Szeregi czasowe w benchmark'u NN3 są stosunkowo krótkie (min 68, max 144), a sam benchmark uznawany jest przez ekspertów z dzedziny modelowania szeregów czasowych jako dość „prosty".

5. Implementacja

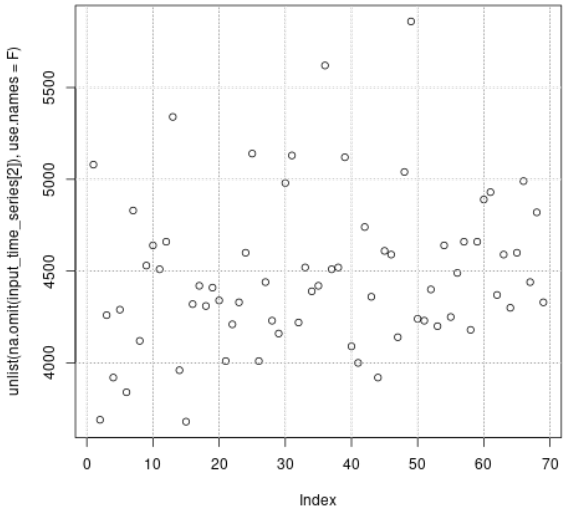
Szeregi czasowe z benchmark'u NN3 zostały znormalizowane do przedziału $< 1 : 2 >$ w celu umożliwienia porównywania błędów bezwzględnych. Wadą normalizacji może być przeskalowanie większości z wartości szeregu do wąskiego przedziału w przypadku, kiedy w danych znajduje się co najmniej 1 mocno odstający punkt [1]. Przesunięcie zakresu z przedziału $< 0 : 1 >$ do $< 1 : 2 >$ pozwala również na obliczanie błędów względnych bez obawy dzielenia przez 0.

Predykcja wartości odbywa się z krokiem o długości 1. Oznacza to, iż przy predykcji wartości dla chwil czasowych $t + n$ wykorzystywane są rzeczywiste wartości szeregu z chwil czasowych wcześniejszych od $t + n$.

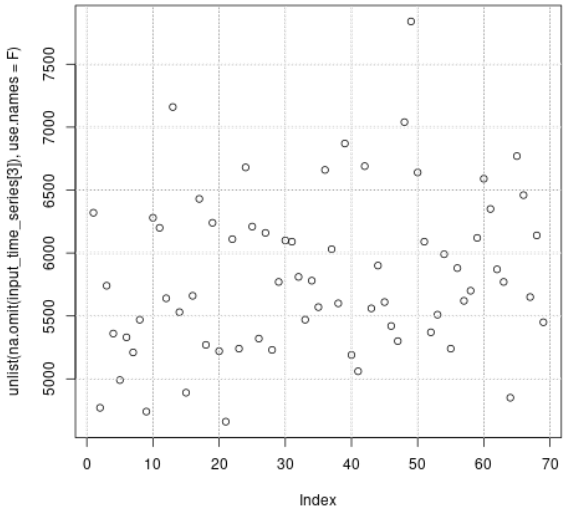
5.1. ARIMA

Predykcje z krokiem o długości 1 zrealizowano tworząc oddzielny model dla każdej wartości szeregu testowego. Szereg uczący w każdym kroku był zwiększany o jedną wartość z szeregu testowego.

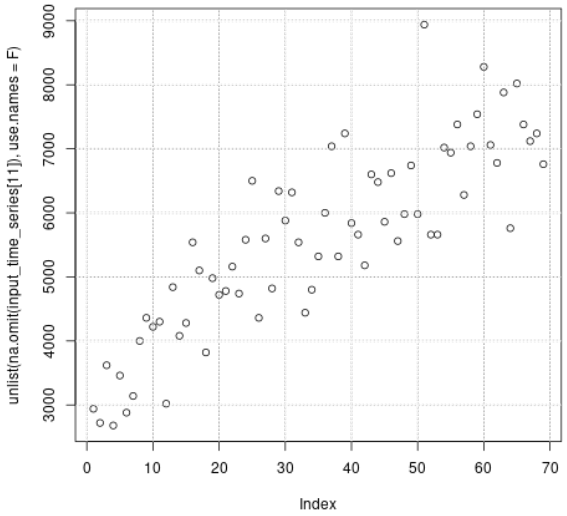
```
learn_data_length = length( tmp_ts_learn )
for (j in 1:TEST_DATA_LENGTH) {
```



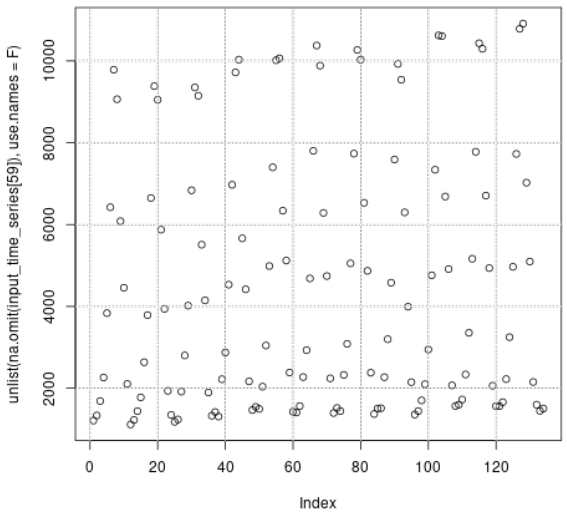
(a) Dane zaszumione



(b) Dane zaszumione



(c) Trend



(d) Wahania okresowe

Rysunek 1: Przykładowe szeregi czasowe z benchmark'u NN3.

```
    ts_end = learn_data_length + j
    ts < ts(tmp_ts[1:ts_end])
    arima_model = auto.arima(ts)
    arima_forecast_all = forecast(arima_model, h = 1)
    arima_forecast = as.data.frame(arima_forecast_all)$'Point_Forecast'
    arima_forecast_oneAhead[j] = arima_forecast
}
```

5.2. XGBoost

Do uzupełnienia.

6. Prezentacja wyników

Porównanie modelu ARIMA z metodą xgboost dokonane zostanie na podstawie przebiegu funkcji gęstości prawdopodobieństwa względnych błędów. Wartości błędów nie będą uśredniane dla poszczególnych szeregów, co oznacza, że długość wektora błędów dla którego zostanie przedstawiony przebieg funkcji gęstości prawdopodobieństwa wynosić będzie $111 \times 18 = 1998$.

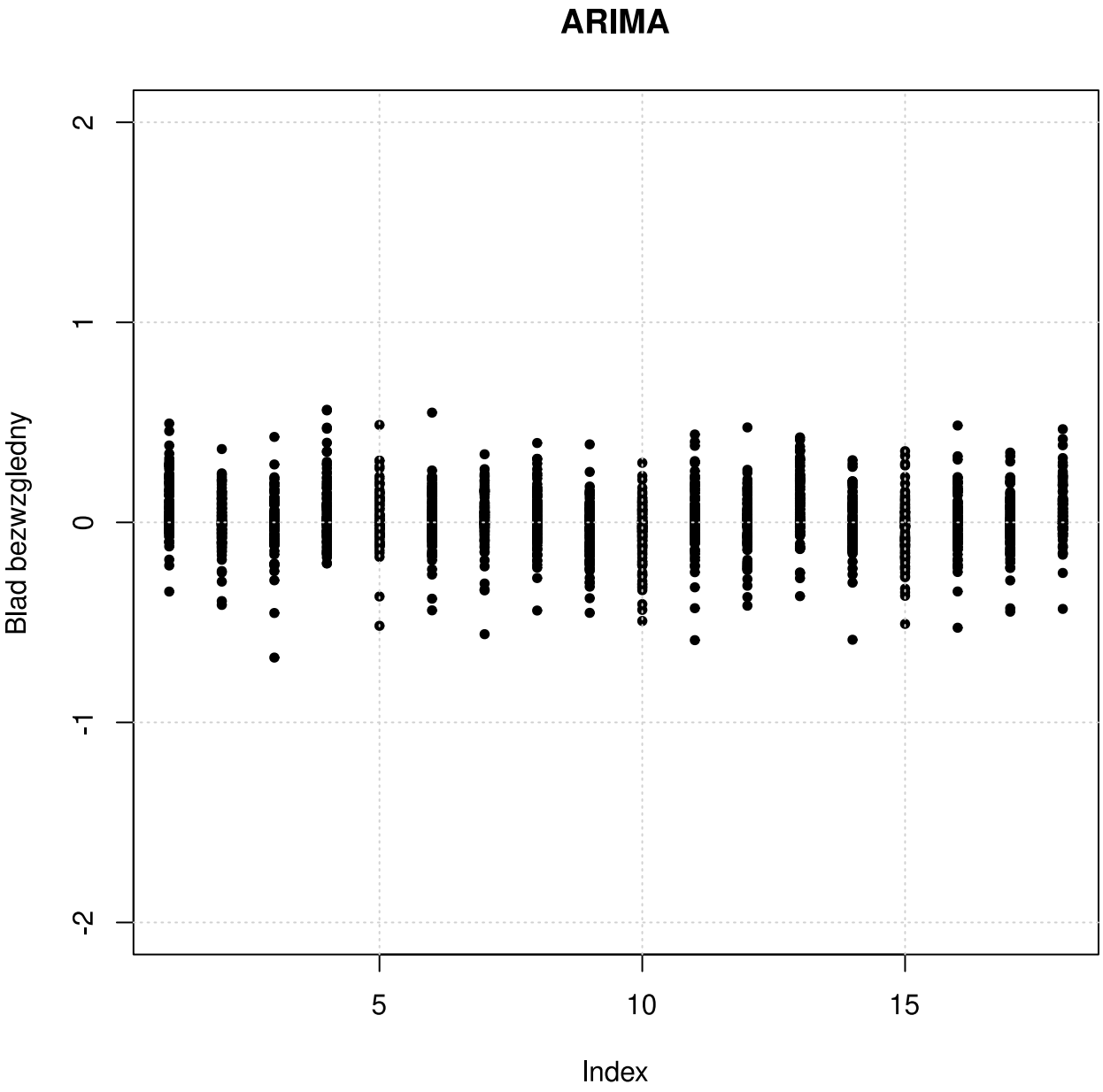
Zagregowana funkcja gęstości prawdopodobieństwa błędów może ukrywać niektóre właściwości stosowanych modeli, dlatego zostanie ukazany również, na wspólnym wykresie, przebieg błędów dla wszystkich szeregów czasowych.

Wyniki strojenia parametrów metody xgboost zaprezentowane zostaną w analogiczny sposób. Dodatkowo zamieszczone mogą zostać fragmenty wydruków stosowanych funkcji, niosące wartościowe informacje.

Bibliografia

[1] Standardization vs. normalization | Data Mining Blog - www.dataminingblog.com. http://www.dataminingblog.com/standardization-vs-normalization/.

[2] Sebastian Zając. Modelowanie szeregów czasowych za pomocą procesówarma i arima. http://sebastianzajac.pl/work/BSc.pdf.


ARIMA
Blad bezwzgledny
2
1
0
-1
-2
5
10
15
Index