

Metodyka i praktyka filtracji opartej na ślepej separacji sygnałów

1. Wprowadzenie

Filtracja szeregów czasowych przy wykorzystaniu dekompozycji wielowymiarowych może być zastosowana w różnych obszarach analizy danych. Do najpopularniejszych można zaliczyć estymację trendów, eliminację szumów, a także agregację modeli. Czynnikiem łączącym poszczególne zagadnienia jest założenie, że przetwarzane dane są pewną mieszaniną lub kombinacją nieznanymi sygnałów (komponentów) źródłowych, które mogą być separowane. Założenie o nieznanym sposobie mieszania prowadzi nas do problemu ślepej separacji (ang. *Blind Signal Separation* lub *Blind Source Separation*, BSS) i metod tam stosowanych³. Stąd całą filtrację określać będziemy filtracją BSS⁴.

Proces filtracji BSS ustala szeroki schemat badawczy, mający wiele otwartych problemów⁵. W swojej aktualnej postaci daje on, z jednej strony, duże możliwości indywidualnego dopasowania rozwiązań do konkretnego zagadnienia, z drugiej zaś, wymaga wielu arbitralnych decyzji, mających zasadniczy wpływ na jakość uzyskiwanych wyników. W niniejszym opracowaniu przedstawimy kwestie związane z metodyką oraz praktyką filtracji BSS. Dla czytelności rozważań przedstawiona zostanie cała koncepcja wraz z typowymi algorytmami, zaliczanymi do rozwiązań ślepej separacji. Algorytmy te zostaną zaprezentowane w swoich podstawowych postaciach. O ile bowiem z teoretycznego punktu widzenia, w danej klasie problemów, koncentrujemy się zwykle na najsilniejszych

¹ Szkoła Główna Handlowa w Warszawie.

² Szkoła Główna Handlowa w Warszawie.

³ P. Comon, Ch. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, 2010, s. 7.

⁴ R. Szupiluk, *Dekompozycje wielowymiarowe w agregacji predykcyjnych modeli data mining*, Oficyna Wydawnicza SGH w Warszawie, Warszawa 2013.

⁵ R. Szupiluk, P. Rubach, *Extreme Value Model for Volatility Measure in Machine Learning Ensemble, Artificial Intelligence and Soft Computing*, ICAISC 2018, „Lecture Notes in Computer Science” 2018, vol. 10841.

i najefektywniejszych rozwiązaniach, to z punktu widzenia praktycznych zastosowań prostota implementacji algorytmu może mieć istotne znaczenie. Filtracja BSS zostanie zaprezentowana w kontekście problemu poprawy wyników predykcji.

2. Filtracja BSS

Proces filtracji BSS zostanie zaprezentowany w swej podstawowej koncepcji. Zestaw danych wejściowych (np. wyniki predykcji), traktowanych jako jedna zmienna wielowymiarowa, poddajemy określonej transformacji, w wyniku czego otrzymujemy zbiór ukrytych komponentów bazowych. Wśród komponentów ukrytych identyfikujemy komponenty niepożądane, które eliminujemy. Pozostałe komponenty poddajemy transformacji odwrotnej do wcześniej użytej. W efekcie otrzymujemy wyniki będące „oczyszczoną” wersją danych wejściowych. W zapisie formalnym cała koncepcja wygląda jak niżej.

Przyjmijmy istnienie zbioru m szeregów czasowych x_i , $i = 1, \dots, m$ reprezentujących wyniki predykcji. Szeregi te zbierzemy w jednej wielowymiarowej zmiennej $x = [x_1, \dots, x_m]^T$. Przyjmijmy dalej, że obserwowane wartości zawierają komponenty związane z rzeczywistymi wartościami, jak również komponenty związane z błędami predykcji, takie jak szумы i zakłócenia (losowe lub nie). Inaczej mówiąc, możemy przyjąć, że dany rezultat jest kombinacją istotnych dla nas komponentów konstruktywnych \hat{s}_j , $j = 1, \dots, p$ związanych z rzeczywistymi wartościami predykcji oraz zakłócających \tilde{s}_l , $l = 1, \dots, q$. Wszystkie te składniki związane ze specyfiką danego modelu potraktujemy jako ukryte komponenty bazowe zawarte (zmieszane) w wielowymiarowej zmiennej x . W przypadku liniowego sposobu (systemu) mieszania można to zapisać jako

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k), \quad (1)$$

gdzie k oznacza numer obserwacji lub indeks czasu, macierz $\mathbf{A} = [a_{ij}]$ o wymiarach $m \times n$ reprezentuje system mieszający zaś macierz $\mathbf{s}(k) = [\hat{s}_1(k), \dots, \hat{s}_p(k), \tilde{s}_{p+1}(k), \dots, \tilde{s}_{p+q}(k)]^T$ reprezentuje zbiór komponentów bazowych, gdzie $n = p + q$. Dla uproszczenia przyjmijmy, że $m = n$, macierz \mathbf{A} jest nieosobliwa, zaś wartość oczekiwana $E\{\mathbf{s}\} = \mathbf{0}$. W niektórych sytuacjach wygodnie jest stosować pełny zapis macierzowy postaci:

$$\mathbf{X} = \mathbf{A}\mathbf{s}, \quad (2)$$

gdzie $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(N)]$ zawiera wszystkie N wektorów obserwacji, analogicznie $\mathbf{S} = [\mathbf{s}(1), \mathbf{s}(2), \dots, \mathbf{s}(N)]$

Zauważmy dalej, że identyfikując system mieszający \mathbf{A} oraz komponenty bazowe \mathbf{s} oraz eliminując komponenty zakłócające (stawiając odpowiednio $\tilde{s}_i = 0$), otrzymamy:

$$\hat{\mathbf{x}}(k) = \mathbf{A} [\hat{s}_1(k), \dots, \hat{s}_p(k), \mathbf{0}_{p+1}(k), \dots, \mathbf{0}_n(k)]^T, \quad (3)$$

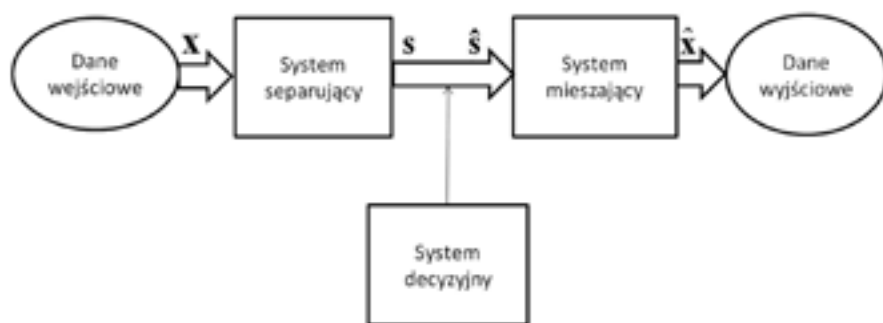
gdzie $\hat{\mathbf{x}} = [\hat{x}_1, \dots, \hat{x}_m]^T$ jest „poprawianą” wersją pierwotnych rezultatów \mathbf{x} . Można to także przedstawić w analogicznej postaci, wstawiając w macierzy $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ zera w kolumnie odpowiadającej eliminowanemu sygnałowi z macierzy \mathbf{S} . W efekcie dla $\hat{\mathbf{A}} = [\mathbf{a}_1, \dots, \mathbf{a}_p, \mathbf{0}_{p+1}, \mathbf{0}_{p+2}, \dots, \mathbf{0}_n]$ otrzymamy:

$$\hat{\mathbf{X}} = [\mathbf{a}_1, \dots, \mathbf{a}_p, \mathbf{0}_{p+1}, \mathbf{0}_{p+2}, \dots, \mathbf{0}_n] \mathbf{s} = \hat{\mathbf{A}} \mathbf{s}. \quad (4)$$

Cały proces filtracji można przedstawić jako:

$$\hat{\mathbf{x}} = \hat{\mathbf{A}} \mathbf{S} = \hat{\mathbf{A}} \mathbf{A}^{-1} \mathbf{x}. \quad (5)$$

Kluczowym zagadnieniem w rozważanej procedurze jest znalezienie takiej transformacji, która rzeczywiście prowadzi do dekompozycji danych \mathbf{x} na interesujące nas komponenty wspierające i zakłócające. Cały proces przedstawiono na rysunku 1.



Rysunek 1. Proces filtracji szeregów czasowych

Źródło: opracowanie własne.

Przedstawiona wyżej metoda wymaga od użytkownika dokonania wyborów, które mogą w istotny sposób wpływać na otrzymane wyniki. Pierwszym silnym założeniem leżącym u podstaw całej metody jest założenie o liniowym

i statycznym zmieszaniu komponentów bazowych. W istocie owo zmieszanie może być znacząco odmienne. Rzeczywiste mieszanie może się realizować w układzie nieliniowym statycznym, dynamicznym liniowym lub nieliniowym dynamicznym. Przyjęcie liniowego modelu jest jednak typowe dla problemów BSS i nie oznacza przesądzenia o naturze zjawiska. Jest to wyłącznie pewien model roboczy, pozwalający na wyprowadzenie wysoce efektywnych, ale jednocześnie stosunkowo prostych w implementacji algorytmów separacji.

3. Metody ślepej separacji

Punktem wyjścia w poszukiwaniu komponentów bazowych \mathbf{s} może być oczekiwanie, że charakteryzują się one pewnymi odrębnymi „fizycznymi” czynnikami determinującymi proces modelowania. Na cały proces możemy także spojrzeć jak na próbę separacji nieznanymi komponentów bazowych zmieszanych w nieznanym systemie. Prowadzi to wprost do tzw. problemu ślepej separacji oraz rozwiązań w nim stosowanych. Rozwiązania tu spotykane realizowane są zwykle w jednym z dwóch podejść. W pierwszym nieznanne wielkości estymowane są wprost z zależności (1). W drugim przypadku poszukiwana jest transformacja odwrotna do (1), co można przedstawić jako:

$$\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k) \approx \mathbf{s}(k) \quad (6)$$

gdzie macierz $\mathbf{W} = \mathbf{A}^{-1}$ odgrywa rolę systemu separującego odwrotnego do \mathbf{A} . W zależności od wybranej metody możemy mieć do czynienia jeszcze z dodatkowymi założeniami. Metodami dekompozycji, jakie zestawimy w ramach niniejszego opracowania, będą: analiza składowych niezależnych oraz algorytm wieloetapowej dekokorelacji AMUSE.

3.1. Analiza składowych niezależnych

Analiza składowych niezależnych (ang. *Independent Component Analysis*, ICA) należy do podstawowych i najstarszych metod ślepej separacji⁶. Stąd w początkowej fazie rozwoju problematyki ślepej separacji często problem i metodę jego rozwiązania utożsamiano ze sobą. Opracowanie metod separacji

⁶ A. Hyvärinen, J. Karhunen, E. Oja, *Independent Component Analysis*, Wiley, New York 2001.

opartych na innych niż wzajemna niezależność kryteriach wiązało się z oddzieleniem problemu (BSS) od metody jego rozwiązania (ICA), niemniej nadal ten związek jest silnie zaznaczony. Wśród wielu algorytmów analizy składowych niezależnych jedno z głównych miejsc zajmuje algorytm typu Natural Gradient⁷. Algorytm ten można przedstawić w postaci

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu(k) [\mathbf{I} - E\{f(y)y^T\}] \mathbf{W}(k). \quad (7)$$

gdzie $f(\cdot)$ jest funkcją nieliniową, której optymalna postać jest następująca:

$$f_i(y_i) = -\frac{\partial \log(p_i(y_i))}{\partial y_i} \quad (8)$$

zaś $p_i(y_i)$ oznacza funkcję gęstości prawdopodobieństwa y_i .

Można przyjąć, że algorytm (7) jest bezpośrednim rozwinięciem pierwszego algorytmu ślepej separacji autorstwa J. Héroult i Ch. Juttena⁸. Doczekał się on licznych rozwinięć i uogólnień⁹. Mimo swej literaturowej popularności, w tym stosunkowo obszernej teorii objaśniającej jego właściwości, oraz licznych implementacji potwierdzających skuteczności działania – jego wykorzystanie bezpośrednio z zadanej formuły może sprawić wiele trudności.

Algorytm ten, zaliczany do klasy algorytmów neuronowych, zawiera zmienny w czasie współczynnik uczenia $\mu(k)$, którego właściwy dobór warunkuje skuteczność separacji. W odróżnieniu od wielu systemów neuronowych stosunkowo odpornych na dobór współczynnika uczenia, w przypadku ww. algorytmu jest on bardzo wrażliwy na charakterystykę przetwarzanych danych. Jego efektywna wersja może mieć postać złożonego systemu filtrów, co stawia pod znakiem zapytania szybkość jego implementację przez użytkownika niewtajemniczonego w jego niuanse.

Kolejną kwestią jest dobór nieliniowości, których optymalna postać (8) zakłada znajomość docelowej postaci estymowanego sygnału, który dopiero jest poszukiwany. W efekcie oznacza to konieczność stosowania różnego rodzaju aproksymacji lub podejść czysto heurystycznych. Należy mieć także na uwadze, że algorytmy ICA nie separują sygnałów gaussowskich, co oznacza, że tylko jeden ze zmieszanych sygnałów może pochodzić z tego rozkładu.

⁷ S. Amari, A. Cichocki, H.H. Yang, *A New Learning Algorithm for Blind Signal Separation*, Advances in Neural Information Processing Systems, NIPS-1995, vol. 8, MIT Press, Cambridge, MA 1996, s. 757–763.

⁸ J. Héroult, Ch. Jutten, *Space or Time Adaptive Signal Processing by Neural Network Models*, International Conference on Neural Networks for Computing, Snowbird 1986.

⁹ A. Cichocki, S. Amari, *Adaptive Blind Signal and Image Processing*, Wiley, Chichester 2002.

Jednak, mimo powyższych ograniczeń oraz trudności, algorytm ICA Natural Gradient w wielu sytuacjach staje się w zasadzie niezastąpiony. Algorytm ten może być stosowany zarówno dla danych posiadających strukturę czasową, jak i dla danych bez tej struktury. Alternatywne algorytmy ICA, jak np. JADE, działają skutecznie, jednak ze względu na swoją złożoność obliczeniową adekwatne są zasadniczo do niewielkiej liczby sygnałów¹⁰. Z kolei algorytmy FASTICA są w zasadzie tożsame z (7), są jednak zorientowane na ekstrakcję pojedynczych sygnałów, w przypadku jednoczesnej separacji wielu sygnałów mogą wymagać dodatkowej operacji ortogonalizacji¹¹. Ponieważ wartość oczekiwaną występującą w (7) można zastąpić wartościami chwilowymi, algorytm jest adekwatny dla dużych i niestacjonarnych zbiorów danych, umożliwiając pracę online.

3.2. Algorytm AMUSE

Algorytm AMUSE jest jednym z podstawowych reprezentantów metod separacji bazujących na statystyce drugiego rzędu¹². W swojej podstawowej, a jednocześnie najprostszej postaci jest zaprezentowany poniżej¹³.

1. Przyjmuje się: $\mathbf{z}(t) = \mathbf{x}(t)$, $p = 0$.
2. Obliczana jest macierz:

$$\mathbf{R}_{zz}(p) = E\{\mathbf{z}(t)\mathbf{z}^T(t-p)\}. \quad (9)$$

3. Dokonywany jest rozkład względem wartości osobliwych (SVD) macierzy $\mathbf{R}_{zz}(p)$:

$$\mathbf{R}_{zz}(p) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (10)$$

gdzie $\mathbf{U} \in \mathfrak{R}^{m \times m}$ oraz $\mathbf{V} \in \mathfrak{R}^{m \times m}$, są macierzami ortogonalnymi, zaś $\mathbf{\Sigma} = \text{diag}\{\sigma_1, \dots, \sigma_m\}$, jest diagonalną macierzą wartości osobliwych.

4. Wykonywana jest transformacja przy danym opóźnieniu:

$$\mathbf{y}(t) = \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\mathbf{z}(t). \quad (11)$$

5. Przyjmując, $\mathbf{z}(t) = \mathbf{y}(t)$, $p = 1$ i powtarzane są punkty 2–4.

¹⁰ J.F. Cardoso, *High-order Contrasts for Independent Component Analysis*, „Neural Computation” 1999, vol. 11, no. 1, s. 157–192.

¹¹ A. Hyvärinen, J. Karhunen, E. Oja, op. cit.

¹² L. Tong, V. Soon, Y.F. Huang, R. Liu, *Indeterminacy and identifiability of blind identification*, „IEEE Transactions on Circuits and Systems” 1991, vol. 38, s. 499–509.

¹³ R. Szupiluk, A. Cichocki, *Ślepa separacja sygnałów przy wykorzystaniu statystyk drugiego rzędu*, XXIV IC-SPETO, Ustroń 2001, s. 485–488.

Algorytm ten cechuje się wieloma zaletami, które mogą przemawiać za jego wykorzystaniem w przypadku samodzielnych implementacji. Przede wszystkim dzięki swojej stosunkowej prostocie jest szybki w implementacji oraz działaniu. Cechuje się wysoką efektywnością, a jednocześnie nie jest zależny (nie posiada żadnych dodatkowych parametrów). Pewnym mankamentem jest jego stosunkowo duża wrażliwość na założoną postać modelu generującego (1). Jednak podstawowym ograniczeniem jest stosowalność wyłącznie do danych posiadających strukturę czasową.

3.3. Nieujemna faktoryzacja macierzy

Nieujemna faktoryzacja macierzy (ang. *Non-negative Matrix Factorization*, NMF)¹⁴ jest kolejną metodą, która może być zaliczona do kategorii metod ślepej separacji. Podstawową ideę NMF można dla modelu (2) sprowadzić do faktoryzacji danej macierzy \mathbf{X} jako iloczynu dwóch macierzy nieujemnych \mathbf{A} i \mathbf{S} ¹⁵. Przez nieujemność macierzy \mathbf{A} rozumie się nieujemność jej elementów. W praktyce \mathbf{X} oznacza nieujemną macierz danych obserwowanych, \mathbf{A} oraz \mathbf{S} zaś są nieujemnymi macierzami faktoryzującymi, reprezentującymi odpowiednio system mieszający oraz sygnały źródłowe. Poszukiwanie macierzy \mathbf{A} oraz \mathbf{S} wiąże się z minimalizacją określonej funkcji celu, które często oparte są na funkcjach dywergencji¹⁶. Jednym z pierwszych algorytmów, który określił sposób znajdowanych rozwiązań NMF, jest algorytm ISRA¹⁷:

$$\mathbf{A} \leftarrow \mathbf{A} \cdot \mathbf{X} \mathbf{S}^T / \mathbf{A} \mathbf{S} \mathbf{S}^T \quad (12)$$

$$\mathbf{S} \leftarrow \mathbf{S} \cdot \mathbf{A}^T \mathbf{X} / \mathbf{A}^T \mathbf{A} \mathbf{S}, \quad (13)$$

gdzie \cdot oznacza mnożenie tablicowe element na element, podobna notacja (matlabowska) odnosi się do dzielenia i potęgowania.

Algorytmy NMF przeżyły w ostatniej dekadzie okres bujnego rozwoju. Należy jednak zaznaczyć, że o ile z punktu widzenia formalnego operacje nieujemnej

¹⁴ A. Cichocki, R. Zdunek, A.-H. Phan, S. Amari, *Nonnegative Matrix and Tensor Factorizations*, Applications to Exploratory Multi-way Data Analysis 2009, John Wiley.

¹⁵ Ibidem.

¹⁶ I. Csiszar, *Axiomatic Characterizations of Information Measures*, „Entropy” 2008, vol. 10, s. 261–273.

¹⁷ D.D. Lee, H.S. Seung, *Learning the Parts of Objects with Nonnegative Matrix Factorization*, „Nature” 1999, no. 401, s. 788–791.

faktoryzacji SA z reguły realizowane są bardzo efektywnie, to jako narzędzie separacji rzeczywistych sygnałów poszczególne problemy NMF wymagają z reguły dedykowanych rozwiązań, często o stosunkowo złożonej postaci. Prezentowany algorytm ISRA jest prosty i łatwy w implementacji, jednak można zaryzykować stwierdzenie, że są to jego podstawowe zalety. Jego tendencja do zbiegania do lokalnych minimów w praktyce oznacza trudności separacji nawet w najprostszych przypadkach.

4. Eksperyment praktyczny

W niniejszym eksperymencie pokażemy skuteczność filtracji szeregów czasowych wykorzystującej ślepą separację sygnałów oraz dokonamy porównania trzech wspomnianych wcześniej algorytmów: Nieujemnej Faktoryzacji Macierzy (NMF), JADE oraz AMUSE. W badaniu próbowano dokonać poprawy predykcji krótkoterminowego zużycia energii elektrycznej. Do tego celu wykorzystano zbiór 2000 obserwacji ogólnego dziennego zużycia energii w Polsce. Dla wybranego okresu wykonano prognozę zużycia energii za pomocą sześciu modeli sieci neuronowych typu MLP optymalizowanych algorytmem Levenberga-Marquardta. W efekcie uzyskano prognozy wielkości błędu MAPE oraz MSE (tabela 1).

Tabela 1. Błędy prognozy dla poszczególnych modeli sieci neuronowych

	x_1	x_2	x_3	x_4	x_5	x_6	avg x
MAPE (10^{-2})	2,670	2,678	2,581	2,660	2,479	2,634	2,617
MSE (10^{-3})	1,022	1,021	1,023	0,978	0,856	1,031	0,989

Źródło: opracowanie własne.

Kolejne kroki wykonano dla każdego z trzech wymienionych powyżej algorytmów. Zgodnie z przedstawionym na rysunku 1 procesem składały się one z następujących etapów:

- 1) dekompozycji prognoz na komponenty bazowe $s_1 \dots s_6$,
- 2) podjęcia decyzji o zastąpieniu wybranych komponentów bazowych zerami – w tym badaniu przeprowadzono testy dla wszystkich możliwych dwuelementowych kombinacji i wybrano dla każdego algorytmu tę kombinację dwóch komponentów bazowych, których usunięcie dawało najmniejszy średni błąd prognozy mierzony za pomocą metody MAPE,
- 3) ponownego mieszania,

- 4) obliczenia błędu prognozy za pomocą MSE oraz MAPE dla skorygowanego modelu.

Wyniki przedstawiono w tabelach 2, 3 i 4 oraz na rysunkach 2, 3 i 4.

Tabela 2. Błędy prognozy dla skorygowanych modeli – algorytm JADE – 5. i 6. komponent bazowy zastąpiony zerami

	\hat{x}_1	\hat{x}_2	\hat{x}_3	\hat{x}_4	\hat{x}_5	\hat{x}_6	avg \hat{x}
MAPE (10^{-2})	2,566	2,653	2,562	2,604	2,465	2,484	2,556
MSE (10^{-3})	0,937	0,994	0,997	0,966	0,870	0,877	0,940

Źródło: opracowanie własne.

Tabela 3. Błędy prognozy dla skorygowanych modeli – algorytm AMUSE – 2. i 5. komponent bazowy zastąpiony zerami

	\hat{x}_1	\hat{x}_2	\hat{x}_3	\hat{x}_4	\hat{x}_5	\hat{x}_6	avg \hat{x}
MAPE (10^{-2})	2,521	2,553	2,488	2,541	2,459	2,604	2,528
MSE (10^{-3})	0,893	0,937	0,937	0,884	0,868	0,985	0,917

Źródło: opracowanie własne.

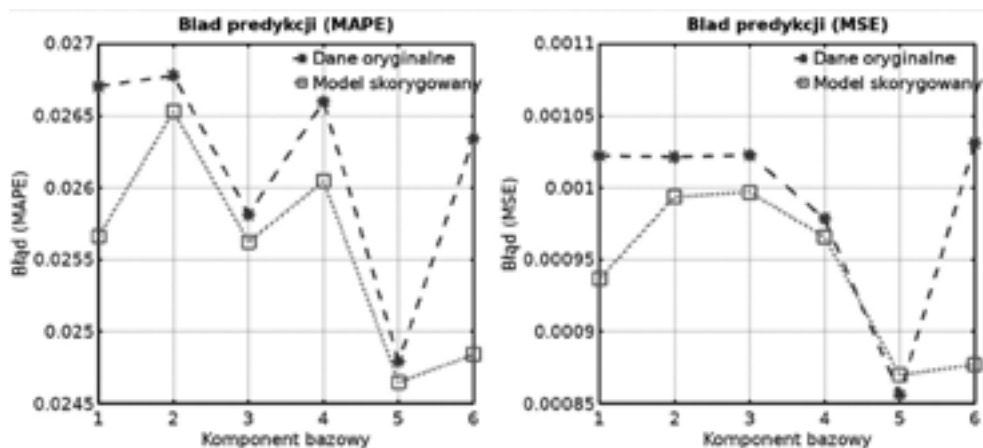
Tabela 4. Błędy prognozy dla skorygowanych modeli – algorytm NMF – 4. i 5. komponent bazowy zastąpiony zerami

	\hat{x}_1	\hat{x}_2	\hat{x}_3	\hat{x}_4	\hat{x}_5	\hat{x}_6	avg \hat{x}
MAPE	473,4	49,7	38,9	111,8	65,1	191,1	155
MSE	719892	331293	99714	99710	143237	25,2	232312

Źródło: opracowanie własne.

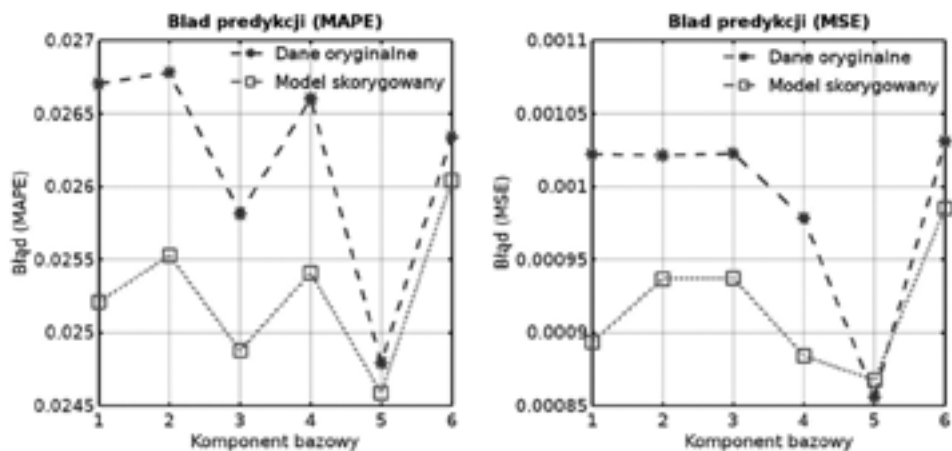
Uwagę zwraca fakt, iż algorytm NMF zupełnie nie poradził sobie z postawionym zadaniem – w tym przypadku wartości średnich błędów prognozy są o kilka rzędów większe niż w oryginalnych modelach predykcyjnych.

Jeśli zaś chodzi o pozostałe dwa algorytmy: JADE oraz AMUSE w przypadku użycia obydwu algorytmów udało się uzyskać zauważalnie niższe błędy prognozy niż dla modeli oryginalnych. Wyniki dla obydwu są zbliżone, choć algorytm AMUSE uzyskał większe niż JADE procentowe obniżenie błędu predykcji. Wynosi ono odpowiednio dla AMUSE: 3,4% mierzone za pomocą MAPE oraz 7,2% za pomocą MSE, a dla JADE 2,3% mierzone za pomocą MAPE oraz 4,9% za pomocą MSE.



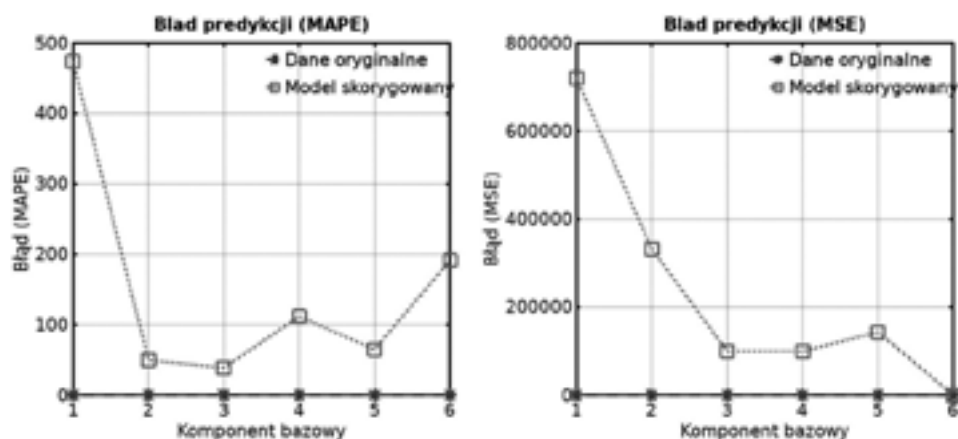
Rysunek 2. Błędy predykcji dla oryginalnego modelu oraz skorygowanego za pomocą algorytmu JADE – 5. i 6. komponent bazy zastąpiony zerami

Źródło: opracowanie własne.



Rysunek 3. Błędy predykcji dla oryginalnego modelu oraz skorygowanego za pomocą algorytmu AMUSE – 2. i 5. komponent bazy zastąpiony zerami

Źródło: opracowanie własne.



Rysunek 4. Błędy predykcji dla oryginalnego modelu oraz skorygowanego za pomocą algorytmu NMF – 3. i 6. komponent bazowy zastąpiony zerami

Źródło: opracowanie własne.

5. Podsumowanie

Teoretyczne ujęcia metod analizy danych często pomijają praktyczne aspekty implementacji. Może się to okazać znaczącym problemem w sytuacji rzeczywistej implementacji danego rozwiązania w warunkach ograniczonego czasu i zasobów. Rozwiązania, które wydają się sprawdzonym standardem, mogą w praktyce wymagać długotrwałego odkrywania wydawałoby się „łatwych” parametrów lub ich ustawień. Wcześniejsza znajomość potencjalnych trudności, które staraliśmy się zasygnalizować, może znacząco poprawić efektywność całego procesu rzeczywistej analizy danych.

Dla pełni wywodu przedstawiona została cała metoda filtracji BSS wraz ze standardowymi algorytmami ślepej separacji. Dla opisanych algorytmów przeprowadziliśmy także badania komputerowe, mające pokazać zalety, ale również słabości poszczególnych algorytmów separacji. Nie jest naszym zamiarem przesądzać o ogólnych możliwościach badanych rozwiązań. Przede wszystkim chcieliśmy zwrócić uwagę, iż praktyka zastosowań może być dość odmienna od potencjalnych teoretycznych możliwości poszczególnych metod.

Bibliografia

- Amari S., Cichocki A., Yang H.H., *A New Learning Algorithm for Blind Signal Separation*, Advances in Neural Information Processing Systems, NIPS-1995, vol. 8, MIT Press, Cambridge, MA 1996, s. 757–763.
- Cardoso J.-F., *High-order Contrasts for Independent Component Analysis*, „Neural Computation” 1999, vol. 11, no. 1, s. 157–192.
- Cichocki A., Amari S., *Adaptive Blind Signal and Image Processing*, Wiley, Chichester 2002.
- Cichocki A., Amari S., *Families of Alpha-Beta-and Gamma-Divergences: Flexible and Robust Measures of Similarities*, „Entropy” 2010, vol. 12, 1532–1568.
- Cichocki A., Zdunek R., Phan A.-H., Amari S., *Nonnegative Matrix and Tensor Factorizations*, Applications to Exploratory Multi-way Data Analysis 2009, John Wiley.
- Comon P., Jutten Ch., *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, 2010, s. 7.
- Csiszar I., *Axiomatic Characterizations of Information Measures*, „Entropy” 2008, vol. 10, s. 261–273.
- Hérault J., Jutten Ch., *Space or time Adaptive Signal Processing by Neural Network Models*, International Conference on Neural Networks for Computing, Snowbird 1986.
- Hyvärinen A., Karhunen J., Oja E., *Independent Component Analysis*, Wiley, New York 2001.
- Lee D.D., Seung H.S., *Learning the Parts of Objects with Nonnegative Matrix Factorization*, „Nature” 1999, no. 401, s. 788–791.
- Szupiluk R., *Dekompozycje wielowymiarowe w agregacji predykcyjnych modeli data mining*, Oficyna Wydawnicza SGH w Warszawie, Warszawa 2013.
- Szupiluk R., Cichocki A., *Ślepa separacja sygnałów przy wykorzystaniu statystyk drugiego rzędu*, XXIV IC-SPETO, Ustroń 2001, s. 485–488.
- Szupiluk R., Rubach P., *Extreme Value Model for Volatility Measure in Machine Learning Ensemble*, Artificial Intelligence and Soft Computing. ICAISC 2018, „Lecture Notes in Computer Science” 2018, vol. 10841.
- Tong L., Soon V., Huang Y.F., Liu R., *Indeterminacy and Identifiability of Blind Identification*, „IEEE Transactions on Circuits and Systems” 1991, vol. 38, s. 499–509.

* * *

Methodology and practical aspects of signal filtration based on Blind Source Separation

Abstract

This paper presents the methodological and practical aspects of signal filtration using Blind Signal Separation. The paper describes the whole process and discusses various methods and algorithms used in signal filtration: ICA FASTICA, JADE, AMUSE and Nonnegative Matrix Factorization (NMF). A practical experiment has been conducted to demonstrate the overall filtration framework and show the effectiveness of three chosen algorithms: JADE, AMUSE and NMF. The practical goal of this experiment was to improve the prediction of short-term energy consumption. While using NMF does not provide an improvement, the other two algorithms perform well and lead to a significant reduction in the prediction error.

Keywords: Time-series Filtration, Non-negative Matrix Factorization, Blind Signal Separation, Blind Source Separation, AMUSE, JADE, ICA