Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Použití pásové struktury při předzpracování dat pro dekompozici binárních matic

2023

Klára Brázdilová

Vedoucí práce:
RNDr. Martin Trnečka, Ph.D.

Studijní program: Informatika,
Specializace: Programování a vývoj software

Bibliografické údaje

Autor:	Klára Brázdilová
Název práce:	Použití pásové struktury při předzpracování dat pro dekompozici binárních matic
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2023
Studijní program:	Informatika, Specializace: Programování a vývoj software
Vedoucí práce:	RNDr. Martin Trnečka, Ph.D.
Počet stran:	52
Přílohy:	elektronická data v úložišti katedry informatiky
Jazyk práce:	český

Bibliographic info

Author:	Klára Brázdilová
Title:	Banded structure in data preprocessing for binary matrix decomposition
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2023
Study program:	Computer Science, Specialization: Programming and Software Development
Supervisor:	RNDr. Martin Trnečka, Ph.D.
Page count:	52
Supplements:	electronic data in the storage of department of computer science
Thesis language:	Czech

Anotace

Výsledkem této bakalářské práce je implementace algoritmů pro převod binární matice do pásové struktury a následně porovnání několika základních filtrů z počítačové grafiky, pro odstranění nerelevantních dat. Práce vychází z výzkumu Banded Structure in Binary Matrices [1] týkající se hledání pásové struktury v binárních maticích.

Synopsis

The result of the work is the implementation of algorithms for conversion a binary matrix into a banded structure. Subsequently, a comparison of several basic filters from computer graphics, to remove irrelevant data. The work is based on the research Banded Structure in Binary Matrices [1] concerning the search for banded structure in binary matrixes.

Klíčová slova: pásová matice; binární data; dekompozice binárních matic; grafické filtry

Keywords: banded structure; binary data; binary matrix decomposition; graphic filters

Ráda bych poděkovala vedoucímu práce RNDr. Martinu Trnečkovi, Ph.D. za nespočet cenných rad a konzultací. Velké díky patří také Mgr. Markétě Trnečkové, Ph.D. za pomoc obzvláště v oblasti počítačové grafiky. Děkuji přátelům a rodině za morální podporu nejen při tvorbě této práce, ale také v průběhu celého studia.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

V každé vědní oblasti je střádána velká masa dat, kterou je třeba zpracovat, porovnat a případně se v ní snažíme hledat určité vzorce. Tato data mívají různou podobu. Některá z nich mají binární podobu a nebo je lze na binární převést. Tímto druhem dat se budeme zabývat.

Některá binární data si můžeme představit poměrně přirozeně jako matici (binární matici). Intuitivně můžeme chápat tuto matici jako zachycení vztahu, relace, mezi vlastností a objektem. Uveďme si příklad.

Představme si matici dat, kde řádky představují pacienty a sloupce představují příznaky, které se u těchto pacientů projevují. Každý pacient daný příznak buďto má (v matici se nachází na odpovídajícím řádku a sloupci 1) nebo nemá (v matici se v odpovídajícím řádku a sloupci nachází 0). Pacienti, kteří mají stejné příznaky by taktéž mohli mít stejnou nemoc. Při zjišťování, kteří pacienti mají stejné příznaky, hledáme souvislé obdélníky jedniček. Tyto obdélníky představují faktory, v našem případě nemoci, které pacienti mají. Binární matici si můžeme představit také jako černobílý obrázek, kde můžeme hledat obdélníky (faktory) i vizuálně. To se nám bude ale lépe dělat, když budeme permutovat řádky a sloupce, dokud obdélníky neuvidíme.

Pro objevení faktorů v datech existuje mnoho algoritmů. Faktory je možné získat pomocí dekompozice binárních matice (Binary matrix factorization). Jedná se o metodu, při které vstupní matici rozložíme na dvě výstupní matice, které popisují faktory. Pokud tyto výstupní matice vynásobíme vznikne (přibližně) původní matice.

V binárních maticích se přirozeně vyskytuje pásová struktura, kterou se budu snažit v této práci získat pomocí několika různých metod, které postupně představím, vysvětlím, provedu jejich implementaci a následně porovnám jejich výsledky. Pásovou strukturu budu získávat pouze pomocí permutování řádků a sloupců.

Dále na těchto datech budeme provádět úpravy. Za předpokladu, že se pásová struktura v datech nachází přirozeně, pak by bylo možné některé jedničky v datech vybočující z tohoto pásu odstranit a naopak doplnit jedničky tam, kde je mezera v jinak plném pásu. Těmito úpravami bychom mohli získat lepší výsledky při hledání faktorů (obdélníků).

Jak jsme si již řekli, binární matice je možné si představit jako obrázky, které jsou černobílé. Místo nuly máme barvu bílou a místo jedničky barvu černou. Díky této vizualizaci, jde přirozeně na data nahlížet a pracovat s nimi. Proto také lze pásovou strukturu z obrázku přímo vidět po permutování řádků a sloupců. Otevírá se možnost využití metod počítačové grafiky. Některé jedničky se nachází mimo pás, na což můžeme nahlížet jako na šum (nebo chybu), který můžeme odstranit. Naopak jedničky, které v pásu chybí mohou být pouze chybou a můžeme je na některá místa doplnit. V této práci představím několik základních filtrů se kterými budu experimentovat a porovnávat jejich výsledky.

Cílem této práce je vyzkoušet výše popsanou myšlenku. Na reálná data bu-

deme aplikovat metody pro získání pásové struktury, následně využijeme základních grafických filtrů a na takto upravená data aplikujeme dekompozici binárních matic. Následně porovnáme výsledky takto upravených dat s původními.

Struktura práce je následující: nejprve si vysvětlíme rozklad binárních matic, představíme si algoritmy pro získání pásové struktury a následně budeme experimentovat s několika grafickými filtry. Na závěr porovnáme jednotlivé metody použité v této práci.

2 Rozklad binárních matic

V této kapitole si představíme problém rozkladu binárních matic. Ukážeme si algoritmy GreConD a ASSO, které budeme následně používat.

2.1 Problém dekompozice binárních matic (BMF)

Faktorizace binárních matic je proces, při kterém objevujeme skryté vzorce v datech, neboli faktory. Matici chápeme jako objet-atributová data (řádky = objekty, sloupce = atributy). Faktor je popsán objekty, které ho ovlivňují a atributy, které jsou jeho projevem. Pojmy dekompozice, rozklad nebo faktorizace nebudeme v této práci rozlišovat.

Předpokládejme vstupní matici I o rozměrech $m \times n$, která je binární. Jednotlivé elementy matice budeme značit I_{ij}, kde i je index řádku a j je index sloupce. Pro libovolné i, j tedy platí, že $I_{ij} = 0$ nebo $I_{ij} = 1$.

Obecným cílem faktorizace matic je k vstupní matici $I \in \{0,1\}^{m \times n}$ vypočítat matice $A \in \{0,1\}^{m \times k}$ a $B \in \{0,1\}^{k \times n}$ takové, že

$$I \approx A \circ B,$$

kde $\circ$ označuje násobení binárních matic: $(A \circ B)_{ij} = max_{l=1}^{k} min(A_{il}, B_{lj})$, k je počet faktorů. Rozklad matic nemusí být nutně přesný. Někdy nám stačí jen rozklad přibližný a $\approx$ představuje přibližnou rovnost. Příkladem přibližného rozkladu matice I je následující rozklad:

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \circ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Například pro matici I bude vypadat dekompozice následovně:

$$I = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \circ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = A \circ B.$$

První matice odpovídá matici I a další dvě korespondují s maticemi A a B. V tomto případě je $k = 5$ a jedná se o přesný rozklad.

Faktorizace má několik praktických využitích, například jiný (úspornější) popis dat nebo také redukce dimensionality. V této práci se nebudeme zabývat dekompozicí jiných než binárních matic, jelikož jsou binární data lehce interpretovatelná a většinu nebinárních dat lze na binární převést pomocí škálování.

2.1.1 Chyba

Při dekompozici dat budeme připouštět chybu. Chybová funkce E pro vstupní matici $C \in \{0,1\}^{m \times n}$ a matici $D \in \{0,1\}^{m \times n}$ je definována následovně:

$$E(C, D) = ||C - D|| = \sum_{i,j=1}^{m,n} |C_{ij} - D_{ij}|.$$

Výsledná chyba má dvě komponenty:

- E_u – uncover error, chyba nepokrytí;
- E_o – overcover error, chyba překrytí.

Chybu nepokrytí můžeme intuitivně chápat tak, že v místě, kde se v matici I nachází 1, tak po vynásobení matice A a B se nachází na daném místě 0. Takže jsme tímto rozkladem nedokázali danou jedničku zachytit (pokrýt). Naopak u chyby překrytí máme v původní matici I hodnotu nula, ale u výsledku $A \circ B$ je na daném místě hodnota jedna.

Výše uvedené komponenty definujeme následovně:

$$E_u(I, A \circ B) = |\{\langle i, j \rangle ; I_{ij} = 1, (A \circ B)_{ij} = 0\}|;$$

$$E_o(I, A \circ B) = |\{\langle i, j \rangle ; I_{ij} = 0, (A \circ B)_{ij} = 1\}|.$$

Definice chyby pomocí komponent je tedy:

$$E(I, A \circ B) = E_u(I, A \circ B) + E_o(I, A \circ B).$$

Tyto komponenty nejsou symetrické. Při přidání faktoru E_u klesá, ale E_o roste. Následující dekompozice ukazuje chybu nepokrytí:

$$I = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \approx \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

Chyba překrytí může vypadá následovně:

$$I = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \approx \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

2.1.2 Základní pohledy na BMF

Na problém dekompozice binárních matic se můžeme dívat dvěma různými způsoby.

Prvním pohledem je Discrete Basis Problem (DBP, [2]), který pro matici $I \in \{0,1\}^{m \times n}$ a pozitivní celé číslo k, nalezne matice $A \in \{0,1\}^{m \times k}$ a $B \in \{0,1\}^{k \times n}$ minimalizující $||I - (A \circ B)||$. Zajímá nás tedy k nejdůležitějších faktorů matice I.

Druhým pohledem na BMF je Approximate Factorization Problem (AFP, [3]). Pro matici $I \in \{0,1\}^{m \times n}$ a danou chybu $\epsilon \geq 0$, najde matice $A \in \{0,1\}^{m \times k}$ a $B \in \{0,1\}^{k \times n}$ s k co nejmenším, tak aby $||I - (A \circ B)|| \leq \epsilon$. Hledáme tedy co nejmenší počet faktorů, které vysvětlují matici I s danou chybou.

V této práci budeme uvažovat, jak pohled AFP, tak DBP. K řešení AFP budeme používat algoritmus GreConD [4] a k řešení DBP algoritmus ASSO [2].

2.2 Algoritmus GreConD

GreConD provádí specifické hladové vyhledávání takzvaně „na vyžádání“ pro formální koncepty vstupní matice I a využívá tyto koncepty k vytvoření výstupních matic A a B. Vyhýbá se výpočtu všech formálních konceptů I, což vede k úspoře času při zachování kvality rozkladu [5].

2.2.1 Optimální dekompozice pomocí formálních konceptů

Formální konceptuální analýza [6] nám poskytuje framework pro práci s faktory. Formální kontext je trojice $\langle X, Y, I\rangle$, kde X je neprázdná množina objektů a Y je neprázdná množina atributů. Binární relace $I \subseteq X \times Y$ pro libovolné $y \in Y$ a libovolné $x \in X$ udává, zda-li $\langle x, y\rangle \in I$ nebo $\langle x, y\rangle \notin I$.

Každá binární matice $I \in \{0,1\}^{m\times n}$ nám indukuje operátory $^{\uparrow}$ a $^{\downarrow}$, $^{\uparrow} : 2^X \to 2^Y$ a $^{\downarrow} : 2^Y \to 2^X$ (ve formální konceptuální analýze známé jako šipkové operátory).

Množina $A^{\uparrow} \subseteq Y = \{1, \ldots, m\}$ a množina $B^{\downarrow} \subseteq X = \{1, \ldots, n\}$ jsou definovány následovně:

$$A^{\uparrow} = \{j \in Y \mid \text{pro každé } i \in A : I_{ij} = 1\},$$
$$B^{\downarrow} = \{i \in X \mid \text{pro každé } j \in B : I_{ij} = 1\}.$$

Hierarchicky uspořádanou množinu všech formálních konceptů $\mathcal{B}(I)$ tvoří úplný svaz, běžně označován jako formální konceptuální svaz.

$$\mathcal{B}(I) = \{\langle C, D\rangle \mid C \subseteq X, D \subseteq Y, C^{\uparrow} = D, D^{\downarrow} = C\}$$

Formální koncepty v binárních maticích můžeme vnímat jako maximální obdélníky plné jedniček, díky čemuž získáme optimální faktory matice I. [4]

Dekompozici pomocí formálních konceptů lze získat následovně. Pro množinu

$$\mathcal{F} = \{\langle C_1, D_1\rangle, \langle C_2, D_2\rangle, \ldots, \langle C_k, D_k\rangle\} \subseteq \mathcal{B}(I)$$

definujeme matici $A_{\mathcal{F}}$ o velikost $m\times k$ a matici $B_{\mathcal{F}}$ o velikosti $k\times n$ následovně:

$$(A_{\mathcal{F}})_{il} = (C_l)(i)$$
$$(B_{\mathcal{F}})_{lj} = (D_l)(j),$$

kde $l \in \{1, \ldots, k\}, i \in \{1, \ldots, m\}$ a $j \in \{1, \ldots, n\}$.

2.2.2 Průběh algoritmu

Pseudokód algoritmu GreConD je popsán v algoritmu 1. Klíčovou částí algoritmu je výběr konceptu $\langle C, D\rangle$, který maximalizuje $(C \times D) \cap \mathcal{U}$ (řádky 6–10). Přímočarým způsobem, jak provést výběr, je vypočítat všechny formální koncepty a poté při každé iteraci vybrat ten s největším $(C \times D) \cap U$. Tohoto způsobu využívá algoritmus GreCon [4]. GreConD místo procházení všech

formálních konceptů, konstruuje kandidáty postupným přidáváním „nadějných sloupců“. Tato myšlenka je založena na skutečnosti, že každý formální koncept $\langle C, D\rangle$ lze vyjádřit jako $D = \bigcup_{y\in D}\{y\}^{\downarrow\uparrow}$ a $C = D^{\downarrow}$.

Z pozorování vychází, že pokud $y \notin D$ pak $\left\langle (D \cup \{y\})^{\downarrow}, (D \cup \{y\})^{\downarrow\uparrow} \right\rangle$ je formální koncept s $D \subset (D \cup \{y\})^{\downarrow\uparrow}$. Proto lze sestavit libovolný formální koncept postupným přidáváním $\{y\}^{\downarrow\uparrow}$ do prázdné množiny atributů. GreConD vybere hladovým přístupem y, který maximalizuje

$$D \oplus y = ((D \cup \{y\})^{\downarrow} \times (D \cup \{y\})^{\downarrow\uparrow}) \cap \mathcal{U}.$$

GreConD prostě prochází sloupce, které maximalizují počet doposud nepokrytých jedniček matice I.

Algorithm 1: GreConD algorithm

Input: Binary matrix I
Output: Set of factors $\mathcal{F}$

$\mathcal{U} \leftarrow \{\langle i, j\rangle \,|I_{ij} = 1\}$
$\mathcal{F} \leftarrow \emptyset$
while $\mathcal{U} \neq \emptyset$ **do**
  $D \leftarrow \emptyset$
  $V \leftarrow 0$
  while *there is* $j \notin D$ *such that* $|D \oplus j| > V$ **do**
    select $j \notin D$ that maximizes $D \oplus j$
    $D \leftarrow (D \cup \{j\})^{\downarrow\uparrow}$
    $V \leftarrow \left|(D^{\downarrow} \times D) \cap \mathcal{U}\right|$
  end
  $C \leftarrow D^{\downarrow}$
  add $\langle C, D\rangle$ to $\mathcal{F}$
  foreach $\langle i, j\rangle \in C \times D$ **do**
    remove $\langle i, j\rangle$ from $\mathcal{U}$
  end
end

2.3 Algoritmus ASSO

Původní verze algoritmu ASSO byla představen v [2] Paulim Miettinenem. Dále se také vyvíjely další verze tohoto algoritmu, jako je například [7]. V této práci budeme uvažovat původní ASSO algoritmus.

2.3.1 Průběh algoritmu

Pseudokód algoritmu ASSO je popsán v algoritmu 2. Algoritmus přijímá na vstupu matici $C \in \{0,1\}^{m \times n}$, kladné celé číslo k, parametr $\tau \in [0,1]$ a reálné váhy w^+ a w^-. Vstupní hodnota k určuje počet faktorů, které má algoritmus vypočítat. Parametry τ, w^+ a w^- udávají míru tolerance chyby.

Algoritmus funguje následovně. Vypočítá čtvercovou matici A o velikost $m \times m$, kde $A_{ij} = 1$, pokud je hodnota confidence asociačního pravidla $\{i\} \to \{j\}$ minimálně τ.

Z matice A získáme dvě matice B a S takové, že $C \approx S \circ B$. Řádky matice A jsou kandidáty na řádky matice B. Aktuálních k řádků matice B jsou vybrané z řádků matice A hladovým způsobem za použití parametrů w^+ a w^- popsaných níže. V průběhu hladového výběru, k sloupců matice S je vybráno k řádků matice B.

ASSO je navržen pro DBP a běžně nevypočte přesný rozklad matice C.

Algorithm 2: ASSO algorithm

Input: A matrix $C \in \{0,1\}^{m \times n}$ for input data, a positive integer k, a threshold value $\tau \in [0,1]$, and real-valued weights w^+ and w^-.
Output: Matrices $S \in \{0,1\}^{m \times k}$ and $B \in \{0,1\}^{k \times n}$ for which $C \approx S \circ B$.

1 **for** $i = 1, \ldots, m$ **do**
2 **for** $j = 1, \ldots, m$ **do**
3 $A_{ij} \leftarrow 1$ **if** $c(\{i\} \to \{j\}, C) \geq \tau$
4 **end**
5 **end**
6 $B \leftarrow []$, $S \leftarrow []$
7 **for** $l = 1, \ldots, k$ **do**
8 $(A_{i\_}, s) \leftarrow \operatorname{argmax}_{A_{i\_} \in \{0,1\}^{1 \times n}, s \in \{0,1\}^{m \times 1}} \mathit{cover}(\begin{bmatrix} B \\ A_{i\_} \end{bmatrix}, [S\ s], C, w^+, w^-)$
9 $B \leftarrow \begin{bmatrix} B \\ A_{i\_} \end{bmatrix}$, $S \leftarrow [S\ s]$
10 **end**
11 **return** B and S

Značením $A_{i\_}$ je myšlen celý řádek i matice A. Funkce $\mathit{cover}(B, S, C, w^+, w^-)$ na řádku 8 je vypočítána jako

$$w^+ \left|\{\langle i,j\rangle \,|C_{ij} = 1, (S \circ B)_{ij} = 1\}\right| - w^- \left|\{\langle i,j\rangle \,|C_{ij} = 0, (S \circ B)_{ij} = 1\}\right|.$$

3 Pásová struktura

V této práci se budeme zabývat pásovou strukturou binárních matic. Binární matice je úplně pásová, pokud jak řádky, tak sloupce mohou být permutovány tak, že nenulové hodnoty budou vytvářet vzor schodů překrývajících se řádků.

Binární matice M je plně pásová, právě když existuje permutace řádků κ a permutace sloupců π taková, že:

1. každý řádek i v matici M_κ^π, kde horní index π značí permutaci sloupců a dolní index κ značí permutaci řádků, se prvky s 1 vyskytují v po sobě jsoucích indexech sloupců $\{a_i, a_i + 1, \ldots, b_i\}$;

2. tyto indexy splňují $a_i \leq a_{i+1}$ a $b_i \leq b_{i+1}$.

Příkladem plně pásové matice je následující matice M:

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Z definice pásové matice je zřejmé, že tato vlastnost platí i na jednotlivé podmatice. Taktéž platí, že pokud je matice M plně pásová, pak i transponovaná matice M^T je také plně pásová.

3.1 Přibližná pásová struktura

U binárních matic, která zachycují reálná (myšleno skutečná) data neočekáváme, že by byla plně pásová. Namísto hledání plně pásové struktury, která v reálných datech ani nemusí existovat, se budeme této plně pásové struktuře snažit přiblížit.

Odlišnost přibližné pásové struktury od plně pásové je možné měřit jako minimální počet prohození, které je třeba provést, abychom získali plně pásovou strukturu. Prohozením je myšleno přepsání hodnoty 0 na hodnotu 1 a naopak. Příslušný optimalizační problém se nazývá Bidirectional Minimum Banded Augmentation (Bidirectional MBA) [1].

Příkladem přibližné pásové struktury je následující matice M:

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

3.2 Algoritmy pro získání pásové struktury

Některé části této kapitoly, jsou volně přeloženy z [1].

Existuje několik algoritmů pro výpočet pásové struktury v binárních datech. Jak již bylo zmíněno, binární matice zachycující reálná data často není možné převést do plně pásové struktury. Na tato data používáme algoritmy, díky kterým se alespoň pásové struktuře přiblížíme. Mezi tyto algoritmy patří Fixed Permutation algorithm, Alternating method a Barycenter algorithm, které byly představeny v [1].

3.2.1 Bidirection Fixed Permutation

Tento algoritmus (algoritmus 3) využívá fixní permutaci sloupců π. Předpokládáme tedy, že správná permutace sloupců vstupní matice M, je již známa. Základní myšlenkou algoritmu Bidirection Fixed Permutation, řešící problém Bidirectional MBA, je pro každý řádek vyřešit problém maximálního podpole (Maximum subarray problem) a následně uspořádat řádky dle vypočítaných hodnot.

Algorithm 3: Bidirectional Fixed Permutation

Input: An $m \times n$ binary matrix M and a permutation π
Output: A permutation κ of rows

1 Fix the column permutation of M to be π
2 **for** *each row* i *in* M^π **do**
3 Let $(W^\pi)_{i\_}$ be the weight vector for row i of M
4 Let $[a, b]$ be the maximum consecutive subarray on $(W^\pi)_{i\_}$
5 Update $(M^\pi)_{i\_} = [a, b]$
6 **end**
7 **for** *each pair of rows* i, j *in* M^π **do**
8 **if** $(M^\pi)_{i\_} \subset (M^\pi)_{j\_}$ **then**
9 Let $A = (M^\pi)_{j\_} \setminus (M^\pi)_{i\_}$
10 Let W_A be the weight vector for A
11 Let $[a, b]$ be the solution of the maximum consecutive subarray of W_A
12 Update $(M^\pi)_{i\_}$ preserving $(M^\pi)_{j\_} \setminus (M^\pi)_{i\_} = [a, b]$
13 **end**
14 **end**
15 Sort rows $[a, b]$ of M^π in an ascending order of a, while deciding ties with the ascending order of their b.

Řádek 1 algoritmu 3 bude vysvětlen v další části textu.

Nyní popíšeme, jak algoritmus funguje. Uvažujeme matici W o rozměrech $m \times n$ definovanou následovně:

$$W_{ij} = \begin{cases} +1 & \text{pokud } M_{ij} = 1; \\ -1 & \text{pokud } M_{ij} = 0. \end{cases}$$

Problém hledání optimálního řešení s obousměrným přepsání hodnot pro $(M^\pi)_{i\_}$ odpovídá řešení problému maximálního podpole pro $(W^\pi)_{i\_}$. Řešením tohoto problému je najít souvislé podpole, jehož součet prvků je maximální. Toto lze provést v lineárním čase vzhledem k velikosti pole. Pokud tedy nalezneme řešení $[a, b]$ pro tento problém, kde a je počáteční sloupcový index a b koncový sloupcový index maximálního podpole, měli bychom aktualizovat $(M^\pi)_{i\_}$ tak, že mezi sloupci a a b budou 1 a všechny ostatní prvky budou 0. To odpovídá řádkům 2–6 v algoritmu 3.

V druhé fázi (řádky 7–14) algoritmus postupuje odstraněním řádků, které brání vytvoření plně pásové matice, podrobněji jsou tyto konflikty popsány v [1]. Užitečným technickým pozorováním je: Nechť $\hat{M}$ je binární matice M rozšířená o nové řádky $M_{ij} = M_j \setminus M_i$ pro každé dva řádky $M_i \subset M_j$. Pak platí, že matice M je plně pásová právě tehdy, když $\hat{M}$ splňuje vlastnost po sobě jdoucích jedniček.

Matice M bude plně pásová pokud její rozšířená matice $\hat{M}$ má po sobě jdoucí jedničky. Proto, aby se eliminovaly všechny možné konflikty mezi řádkovými intervaly M^π, stačí jednoduše projít všechny extra řádky popsané v $\hat{M}$ a zajistit, aby obsahovaly souvislé jedničky. Při projevu změn zpět do M^π bychom získali pásovou matici. Stejně jako výše můžeme využít problém maximálního podpole na extra řádcích $\hat{M}$ k přesnému řešení problému pro obousměrná prohození. Zbývá pouze aktualizovat řádky v M^π tak, aby byly v souladu s provedenými změnami v $\hat{M}$. Konečné získané řešení na M^π bude vždy pásové. V základu to odpovídá rozhodování o nejlepším počtu obousměrných prohozeních, které eliminují konflikty při párovém porovnání řádků v M^π. Konečná složitost algoritmu je $\mathcal{O}(n^2 m)$.

Algoritmus Bidirectional Fixed Permutation však není optimální pro problém Bidirectional MBA na M^π. Problém spočívá ve druhé fázi: řádky jsou porovnávány po párech a mohou být přehlédnuty obecně výhodná prohození.

Zásadní součástí algoritmu Bidirection Fixed Permutation je identifikovat dobrou permutaci sloupců, která bude od začátku zafixována (řádek 1). V následující části budou představeny algoritmy pro výběr této permutace. Intuice říká, že dobrá permutace bude mít tendenci umístit podobné sloupce dostatečně blízko k sobě. Pokud matice vykazuje pásovou strukturu, pořadí založené na podobnosti mezi sloupci by mělo tento pás zachovat.

3.2.1.1 Spectral Ordering

Spectral Ordering je jedna z metod pro výběr permutace sloupců (řádek 1 algoritmu 3). Zakládá se na výpočtu podobnosti jednotlivých sloupců a následného sestupného seřazení podle těch podobností. Výsledek Spectral Ordering bude následně použit jaké vstupní permutace sloupců pro algoritmus 3.

Jedna z možných měr podobnosti mezi dvěma sloupci je podobnost korelace (z angl. correlation similatiry). Pro dva sloupce a a b je korelační podobnost definována jako:

$$CorrS(M\_a, M\_b) = (1 + \rho_{ab})/2,$$

kde ρ_{ab} představuje Pearsonův koeficient mezi sloupci. Hodnoty se pohybují od 1 (identické sloupce) po 0 (úplně odlišné sloupce).

K nalezení dobrého pořadí sloupců budeme využívat spektrální řazení (Spectral Ordering) představené v [8].

Pseudokód algoritmu je popsán v algoritmu 4.

Algorithm 4: Spectral Ordering

Input: An $m \times n$ binary matrix M
Output: Banded matrix M

```
S ← m × m array
for i = 0, ..., m do
    for j=i, ..., m do
        S_{i,j} = S_{j,i} = Copute coeficient value for M_i and M_j
    end
end
L = Laplacian matrix of S
D, V = Eigenvalues and Eigenvectors of L
Find second smallest Eigenvalue and use corresponding Eigenvector
  (also known as Fiedler's vector) as permutation of columns M.
return M
```

3.2.2 Alternating method

Metoda Alternating vychází z tvrzení, že pokud je matice M plně pásová, pak i transponovaná matice M^T je také plně pásová. To znamená, že v praxi můžeme řešit problém Bidirectional MBA buď nad maticí M nebo nad její transponovanou verzí M^T. Tato vlastnost naznačuje následující střídavý přístup: řešení problému nalezení dobré permutace řádků vzhledem k aktuální permutaci sloupců (s využitím Bidirectional Fixed Permutation) a poté transponovat matici, abychom pracovali s aktuální permutací, dokud nedojde ke konvergenci nebo nedosáhneme určitého počtu iterací.

Algorithm 5: The Alternating algorithm for Bidirectional MBA

Input: An $m \times n$ binary matrix M, a number of iterations t
Output: Permuted matrix M
Initialize π with a random permutation of the columns
Let $A = M^{\pi}$
while *reaching t iterations* **do**
    Apply an algorithm 3 on A
    Transpose the current matrix $A = A^T$
end
return Matrix A with the best band found.

Alternating method ne nutně konverguje k řešení u všech matic, jak je vysvětleno v [1].

3.2.3 Barycenter algorithm

Další metodou je Barycenter algorithm. Algoritmus byl původně představen v [9]. Pro hledání dobré permutace řádků a sloupců využívá míry barycentra (průměrné polohy jedniček v řádku). Barycentrum řádku i matice M je definováno následovně:

$$Barycenter(i) = \frac{\sum_{j=1}^{m} j \cdot M_{ij}}{\sum_{j=1}^{m} M_{ij}}.$$

Barycenter algorithm nejprve vypočítá barycentra pro všechny řádky, poté řádky seřadí od nejmenšího po největší podle hodnot barycentra (stabilní třídění) a nakonec transponuje matici M, aby iterativně postupoval podle stejné strategie až konverguje k řešení.

Algorithm 6: Barycenter algorithm

Input: An $m \times n$ binary matrix M
Output: Banded matrix M
while *matrix M is changing* **do**
    $cost = []$
    for i=0, ..., n **do**
        $cost[i]$ = Barycenter value for i
    end
    π = permutation of sorted *cost*
    M^{π}
    $M = M^T$
end
return M

3.2.4 Další varianty

Nabízí se samozřejmě další možnosti získání pásové matice. V této práci prozkoumáme ještě dvě kombinace metod již zmíněných.

Vzhledem k povaze algoritmů se nabízí použít jako vstupní permutaci pro algoritmus 3 výstup algoritmu 6. Dále je možné ještě na tento výsledek algoritmu 3 aplikovat metodu 5. Výsledky těchto variant se budeme zabývat v experimentální části této práce.

4 Grafické filtry

Díky tomu, že binární matice můžeme chápat jako černobílé obrázky, je možné s nimi pracovat pomocí nástrojů počítačové grafiky. Aby bylo možné s daty takto pracovat, je nejprve nutné je převést do přibližné pásové struktury. Odlišnosti od plně pásové struktury lze chápat jako šum, který se budeme snažit odstranit. Na odstranění tohoto šumu budeme používat filtry z počítačové grafiky. Tyto filtry si postupně představíme.

4.1 Deleted band

Hlavním důvodem použití filtrů je přiblížení se k pásové struktuře, proto se přirozeně nabízí filtr DELETED BAND. Tento filtr pracuje na základě odstranění jedniček nacházejících se mimo pás.

Pro vstupní matici M o velikosti $m \times n$ a přirozené číslo k vytvoří matici B o velikost $m \times n$, která je přibližně pásová. Hodnota k, nám v procentech udávat část matice, která má být odstraněna. Pás matice B bude mít šířku 100 % $-k$ %. Následně se provede porovnání matic M a B. Pro každý řádkový index i a sloupcový index j platí, že pokud

$$M_{ij} = 1 \text{ a zároveň } B_{ij} = 0, \text{ pak } M_{ij} = 0.$$

Jedná se tedy o logický součin. Jinými slovy, všechny hodnoty z M, které jsou mimo pás vytvořený v matici B se odstraní.

Možné by bylo vstupní hodnotu k, která nám udává míru odstranění hodnot, stanovit absolutně, ale mnohem výhodnější v případě tohoto filtru bude hodnotu volit relativně vzhledem k podobě dat. Relativní šířka pásu nám dává více možností v experimentální části.

4.2 Square filter

Jako další přirozený filtr se nabízí SQUARE THRESHOLD FILTER (dále jen SQUARE FILTER). Filtr pracuje na základě okolí bodu. Pro každý bod vstupní matice M vypočítá průměr jeho okolí. Pokud průměr překročí určitý práh, pak hodnota bodu bude 1, jinak 0.

Velikost okolí bodu, stejně jako práh je možné volit jako vstupní parametr filtru. Pro okolí bodu je standardní velikost 3. Hodnota prahu je ovšem plně volitelná a je možné s ní více experimentovat, jelikož u každého datasetu může být vhodná jiná hodnota. Více se volbě hodnoty prahu budeme věnovat v experimentální části práce.

4.3 Dilatace a eroze

Jednou ze základních morfologických operací na binárním obraze je DILATACE. DILATACE matice A strukturálním elementem B je definována následovně:

$$A \oplus B = \{z | B_z \cap A \subseteq A\},$$

kde B_z značí posunutí obrazu B podle bodu z. Jinak řečeno se ptáme každého bodu obrazu matice A, pokud umístíme střed strukturálního elementu B na toto políčko, zdali se překrývají na některém bodu obrazu jedničky. Pokud ano, výsledek bude jedna, jinak nula.

Druhou ze základních morfologických operací je EROZE. Jde o duální morfologickou transformaci k dilataci. EROZE binární matice A strukturálním elementem B je definována:

$$A \ominus B = \{z | B_z \subseteq A\},$$

kde B_z značí posunutí obrazu B vektorem z, neboli $B_z = \{c | c = b + z \text{ pro } b \in B\}$. [10]

Princip je stejný jako u DILATACE, ale požadujeme aby na všech místech, kde je v matici B jednička byla na těchto místech jednička i v matici A, pak bude výsledek 1, jinak 0.

Důležitou roli tohoto filtru hraje podoba strukturálního elementu B, jehož volbě se budeme věnovat v experimentální části.

5 Experimentální část

V této části práce se budeme věnovat implementaci již zmíněných algoritmů a následně je budeme testovat na reálných datech. Taktéž se budeme věnovat vlivu obrazových filtrů na dekompozici.

5.1 Použité technologie

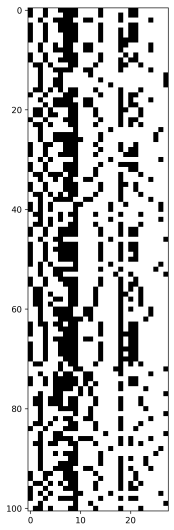
Všechny algoritmy pro tuto práci byly implementovány v jazyce Python. Použita byla knihovna NumPy [11] a knihovna Matplotlib [12]. Veškeré implementace včetně skriptů pro spuštění následujících experimentů jsou součástí elektronické přílohy této práce.

5.2 Datasety

V následující části textu budeme pracovat s několika datasety, na kterých budeme algoritmy testovat a porovnávat. Jedná se o konkrétně o datasety `zoo`, `paleo`, `mushroom` a `healthcare`. Všechny tyto datasety jsou binární a standardně používané jako testovací data v oblasti dekompozic binárních matic.

5.2.1 Dataset zoo

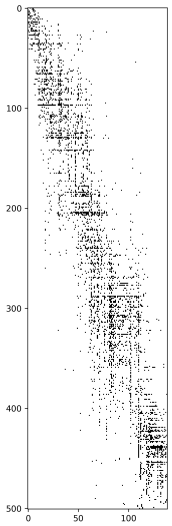
Dataset `zoo` zachycuje u každého zvířete (objekty), zdali danou vlastnost (atribut) má či ne. Obsahuje informace o 101 objektech a jejich 16 atributech. Obrázek 1 ukazuje tento dataset. [13]



Obrázek 1: Dataset `zoo`

5.2.2 Dataset paleo

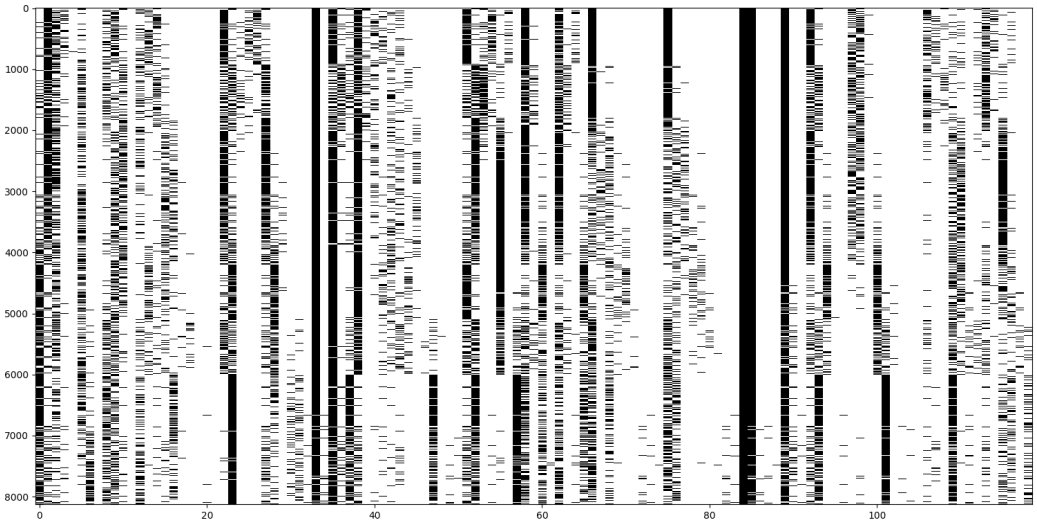
Tento dataset popisuje fosilní záznamy podle umístění (objekty) a pomocí různých vlastností (atributy). Velikost datasetu je 501 objektů a 139 atributů. Obrázek 2 ukazuje tento dataset. [14]



Obrázek 2: Dataset `paleo`

5.2.3 Dataset mushroom

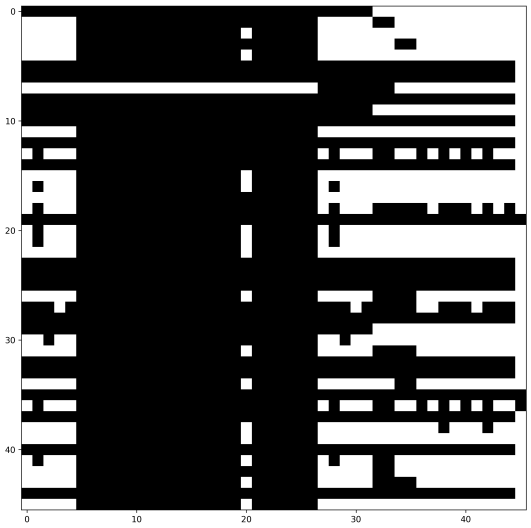
Dataset `mushroom` popisuje vzorky odpovídající 23 druhům žábrových hub z čeledi Agaricus a Lepiota. Velikost datasetu je 8124 objektů a 119 atributů. Obrázek 3 zobrazuje tento dataset. [15]



Obrázek 3: Dataset `mushroom`

5.2.4 Dataset healthcare

Dataset `healthcare` [16] představuje pravidla pro řízení přístupu k síti používaná ve společnosti Hewlett Packard pro řízení konektivity externích obchodních partnerů. Obsahuje 46 objektů a 46 atributů. Obrázek 4 zobrazuje tento dataset.



Obrázek 4: Dataset `healthcare`

5.3 Získání pásové struktury

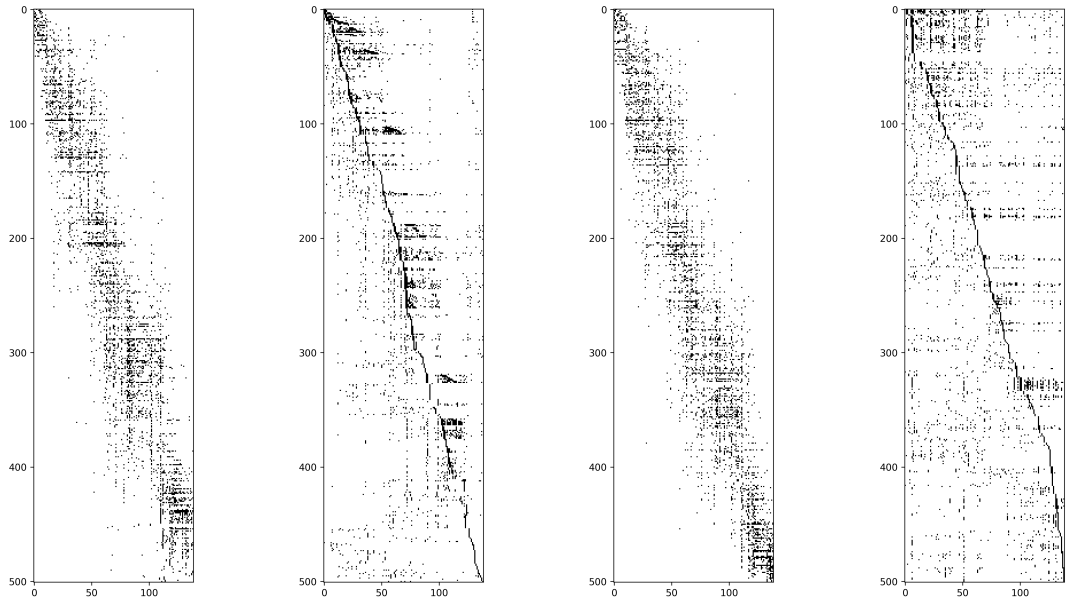
V kapitole 3.2 jsme si představili algoritmy 3, 5 a 6 pro získání pásové struktury. Všechny tyto metody jsem pro účely této práce otestovala na výše zmíněných datasetech. Kompletní výsledky jsou dostupné v elektronické příloze této práce. Kvůli množství výsledků jsou v tomto textu ukázány pouze reprezentativní výsledky, které zachycují trend.

Porovnání výsledků na datasetu `paleo` je na obrázku 5. Výsledky pro dataset `zoo` vidíme na obrázku 6.

Z obrázků 5 a 6 je zřejmé, že výsledky metody Alternating jsou poměrně blízko pásové struktuře. Podobně je na tom o metoda Spectral Ordering BFP. U metody Barycenter jsou sice data v okolí pásu, ale netvoří přímo viditelný pás. Proto jsme následně experimentovali také s možností kombinací těchto algoritmů, jak je popsáno v sekci 3.2.4. Výsledky pro dataset `paleo` vidíme na obrázku 7 a výsledky pro dataset `zoo` vidíme na obrázku 8.

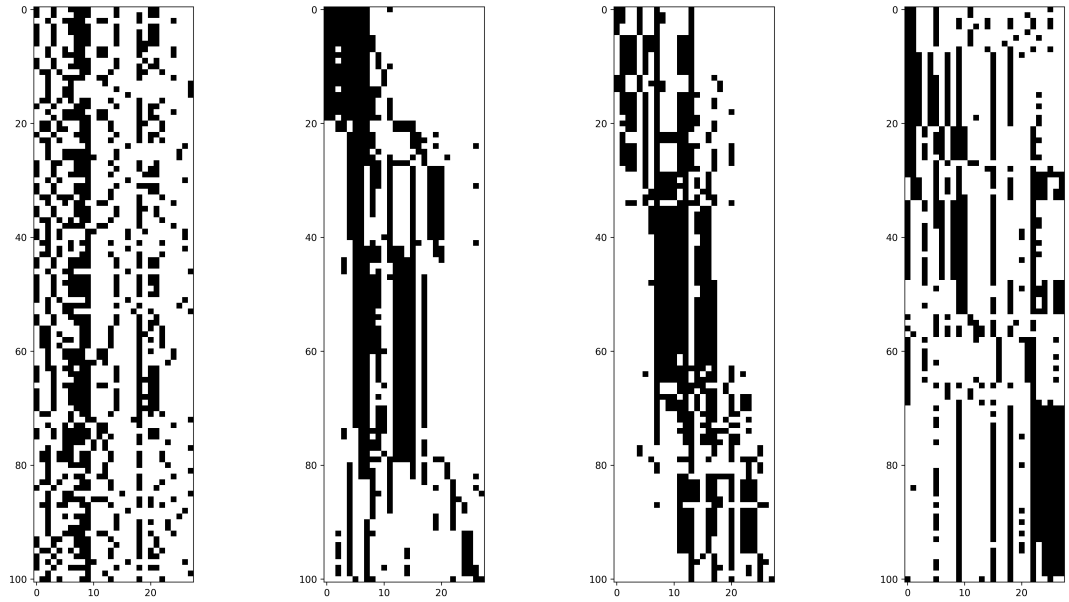
Výsledky kombinace metod Barycenter BFP se blíží pásové struktuře. U kombinace metod Barycenter BFP Alternating nám už vznikají sloupcovité artefakty, které nejsou úplně žádoucí.

Obecně se výsledky lišily podle datasetu, ale lze pozorovat, že nejlepší výsledky dává metoda Alternating a varianta Barycenter BFP.



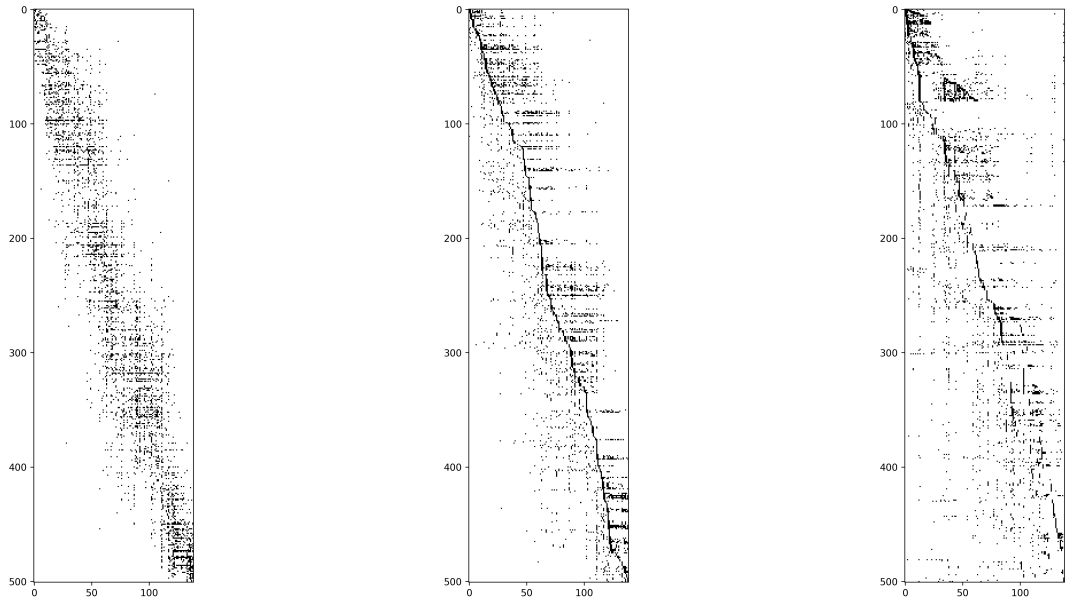
(a) (b) (c) (d)

Obrázek 5: Obrázek 5a ukazuje originální data před permutací. Obrázek 5b ukazuje výsledek metody Alternating, obrázek 5c ukazuje výsledek metody Barycenter a obrázek 5d ukazuje výsledek metody Spectral Ordering BFP.



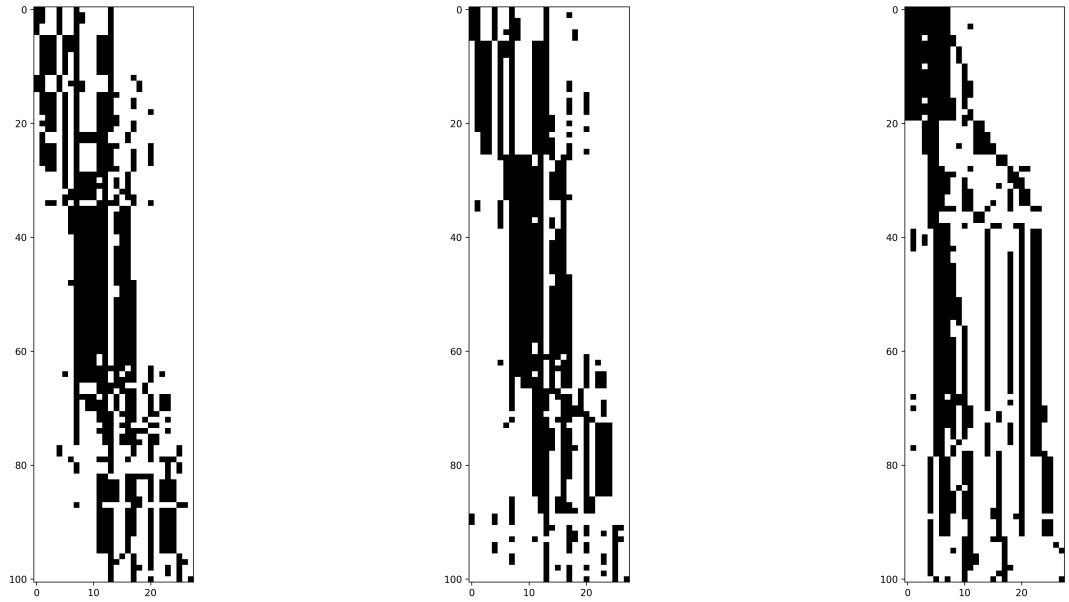
(a) (b) (c) (d)

Obrázek 6: Obrázek 6a ukazuje originální data před permutací. Obrázek 6b ukazuje výsledek metody Alternating, obrázek 6c ukazuje výsledek metody Barycenter a obrázek 6d ukazuje výsledek metody Spectral Ordering BFP.



(a) **(b)** **(c)**

Obrázek 7: Obrázky 7 pracují s datasetem `paleo`. Na obrázku 7a vidíme výsledek pouze po aplikaci metody Barycenter. Obrázek 7b zobrazuje kombinaci metod Barycenter BFP. Obrázek 7c zobrazuje kombinaci Barycenter BFP Alternating.



(a) **(b)** **(c)**

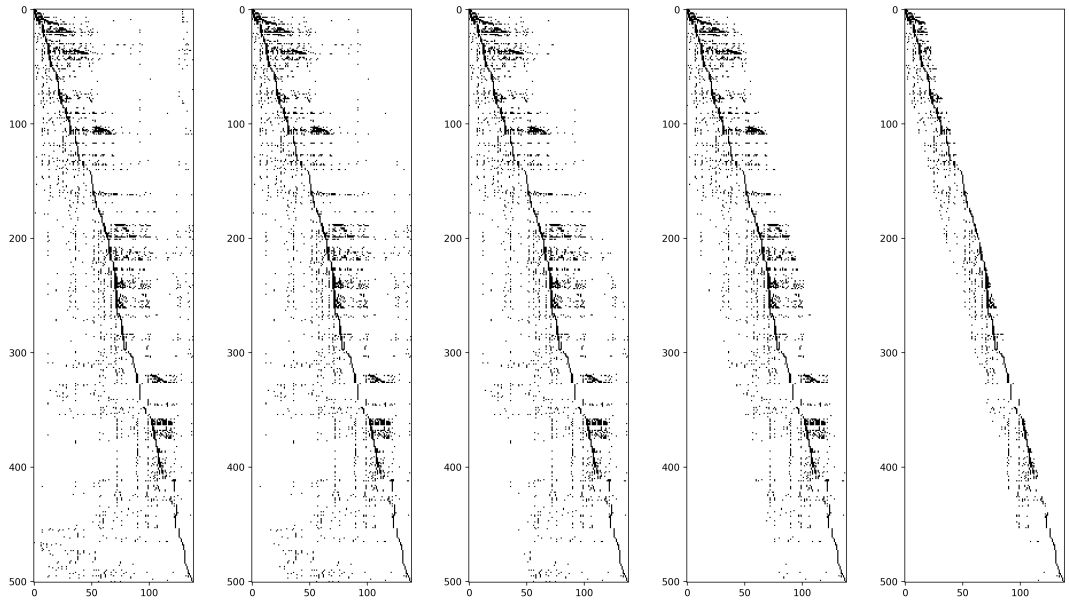
Obrázek 8: Na obrázku 8a vidíme dataset `zoo` pouze po metodě Barycenter. Obrázek 8b zobrazuje kombinaci metody Barycenter a BFP. Obrázek 8c zobrazuje kombinaci metod Barycenter, BFP a Alternating také na datasetu `zoo`.

5.4 Obrázkové filtry

V kapitole 4 jsme si představily několik filtrů, které můžeme aplikovat na matice převedené do přibližné pásové struktury. Jedná se o filtry DELETED BAND, SQUARE FILTER, DILATACE a EROZE. Připomeňme si, že binární matice můžeme chápat jako černobílé obrázky. Díky tomu můžeme odlišnosti od plně pásové struktury chápat jako šum, který se budeme snažit odstranit. Nyní si chování jednotlivých filtrů ukážeme na reálných datech.

5.4.1 Deleted band

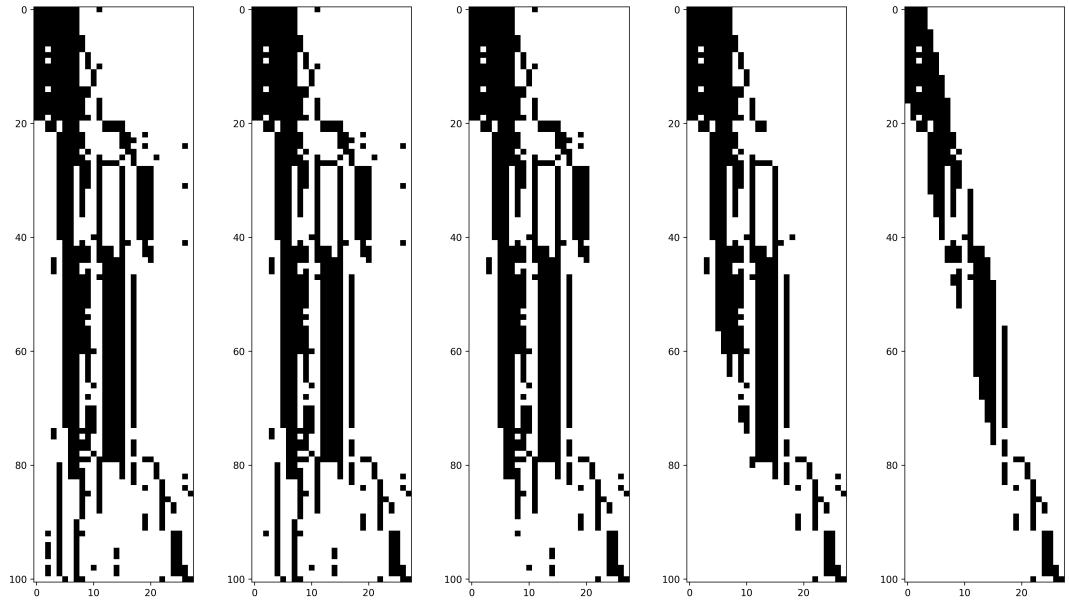
Filtr DELETED BAND jsme si představili v kapitole 4.1. Hlavní experimentální částí u tohoto filtru je volba šířky pásu. Pro účely této práce byly zvoleny šířky pásu 30 %, 50 %, 70 % a 90 %. Chování tohoto filtru se lišilo podle datasetu. K nejlepším výsledkům docházelo u datasetu PALEO. Na obrázcích 9 a 10 vidíme příklady chování.



(a) **(b)** **(c)** **(d)** **(e)**

Obrázek 9: Na obrázku vidíme chování filtru deleted band na datasetu `paleo` po metodě ALTERNATING. Data po metodě ALTERNATING vidíme na 9a. Následně aplikování filtru deleted band s šířkou mazaného pásu 30 % – obrázek 9b, 50 % – obrázek 9c, 70 % – obrázek 9d a 90 % – obrázek 9e.

Z tabulky 1 můžeme vidět míru podobnosti matic před a po aplikování filtru. Podobnost je uvedena v procentech a vždy je počítána z matice po permutaci a matice po permutaci a aplikování filtru. Při hodnotách 70 % a 90 % docházelo k velkému pozměnění originálních dat, ovšem jen u některých datasetů. Výsledky tohoto filtru nebyly příliš příznivé.



Obrázek 10: Na obrázku vidíme chování filtru DELETED BAND na datasetu ZOO po metodě ALTERNATING. Uspořádání hodnot šířek pásu je stejné jako na obrázku 9.

5.4.2 Square filter

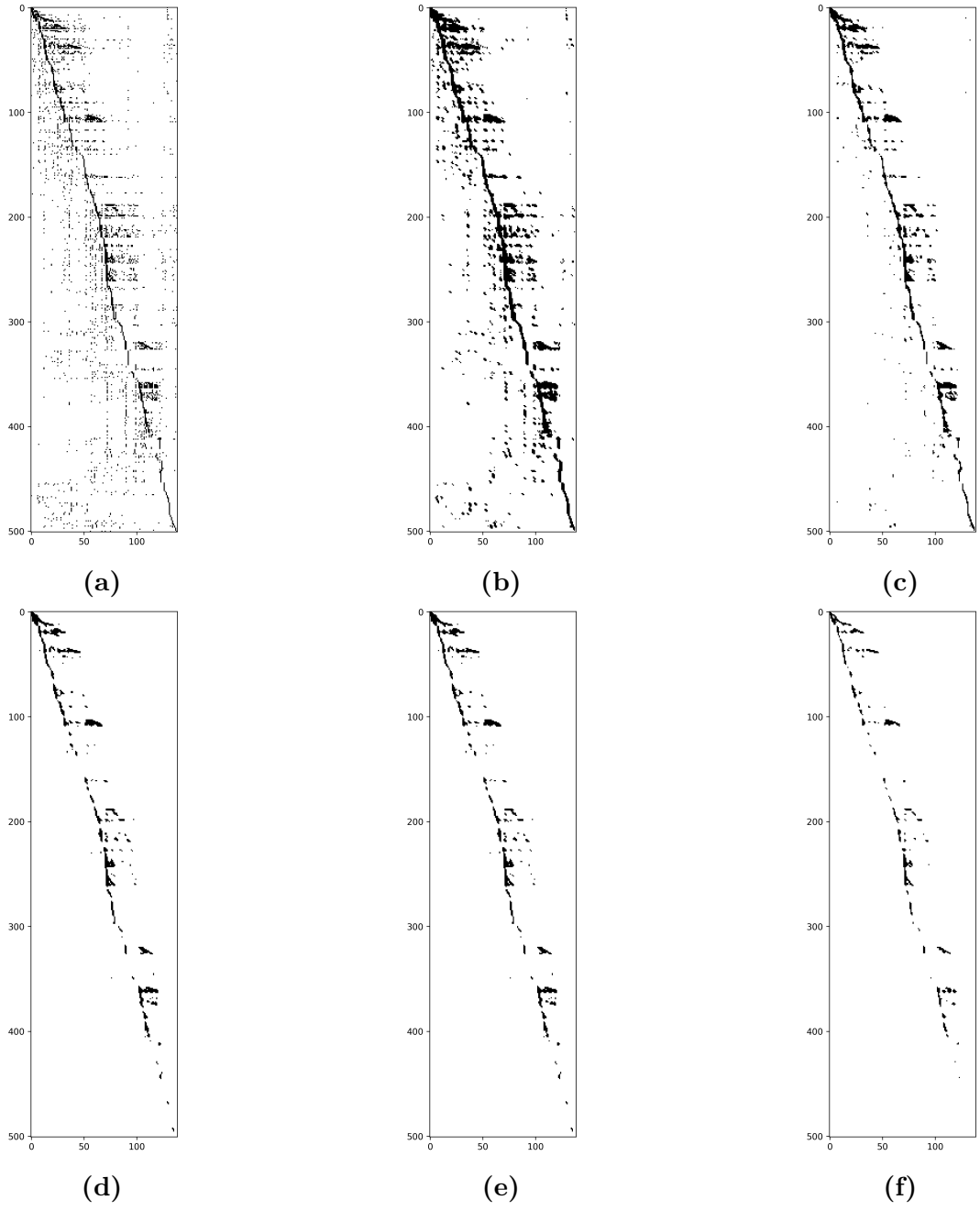
Tento filtr jsme si představili v kapitole 4.2. SQUARE FILTER pracuje na základě okolí bodu, které bylo pro účely této práce zvoleno velikosti 3. Pro hodnotu prahu byly voleny hodnoty $0,2, 0,3, 0,35, 0,4$ a $0,5$. Na obrázcích 11 a 12 vidíme příklad chování filtru s různými hodnoty prahu (hranice).

Tabulka 2 popisuje podobnost matice před a po použití SQUARE FILTER s různými hranicemi.

Z obrázků 11, 12 a tabulky 2 lze jasně vidět, že při hranici 0,35 až 0,5 docházelo k velkému pozměnění dat. Výsledky tohoto filtru nebyly obecně dobré.

dataset a daná metoda	šířka pásu			
	30%	50%	70%	90%
paleo				
spectral-ordering-pearson-bfp	99,82	99,46	98,61	96,55
barycenter-bfp	100,00	99,99	99,66	97,43
alternating	99,90	99,74	99,36	97,39
barycenter	100,00	100,00	99,93	98,18
barycenter-bfp-alternating	99,96	99,87	99,48	97,28
zoo				
spectral-ordering-pearson-bfp	99,65	96,75	91,02	80,87
barycenter-bfp	99,93	99,33	96,04	84,55
alternating	99,82	98,41	92,89	83,77
barycenter	100,00	99,82	95,83	84,87
barycenter-bfp-alternating	100,00	98,62	91,51	79,56
healthcare				
spectral-ordering-pearson-bfp	96,36	86,06	66,68	44,33
barycenter-bfp	96,88	89,93	76,65	52,74
alternating	97,54	87,76	70,84	47,35
barycenter	96,88	90,08	77,50	52,17
barycenter-bfp-alternating	96,88	89,89	77,46	51,98
mushroom				
spectral-ordering-pearson-bfp	98,34	95,40	90,81	84,51
barycenter-bfp	99,77	98,72	94,40	86,53
alternating	99,03	96,25	91,40	84,83
barycenter	99,91	99,08	95,02	87,10
barycenter-bfp-alternating	98,16	95,19	89,91	84,00

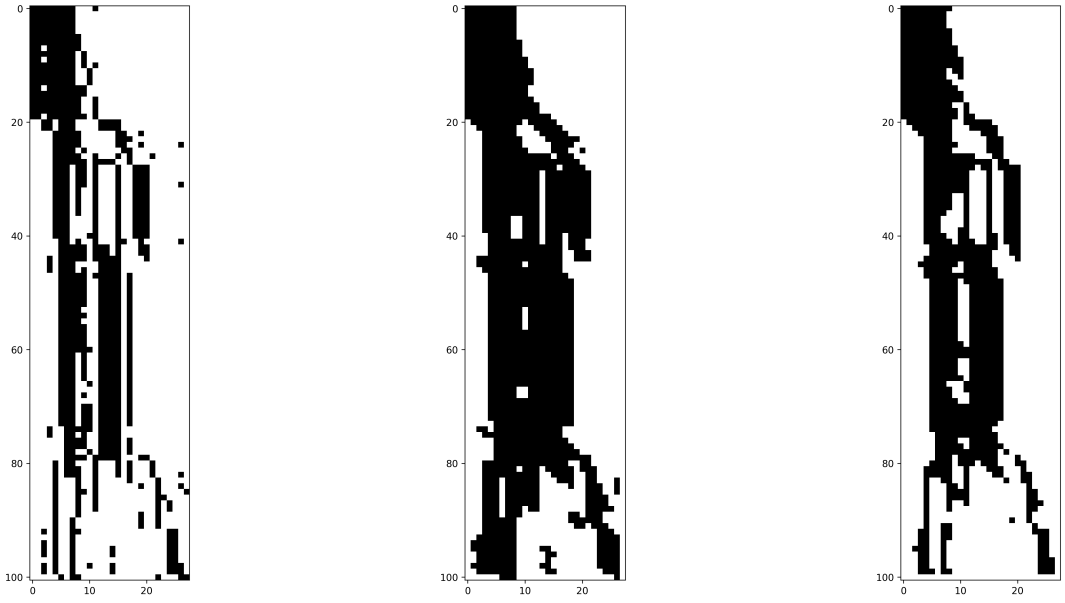
Tabulka 1: Podobnost matic po použití filtru DELETED BAND



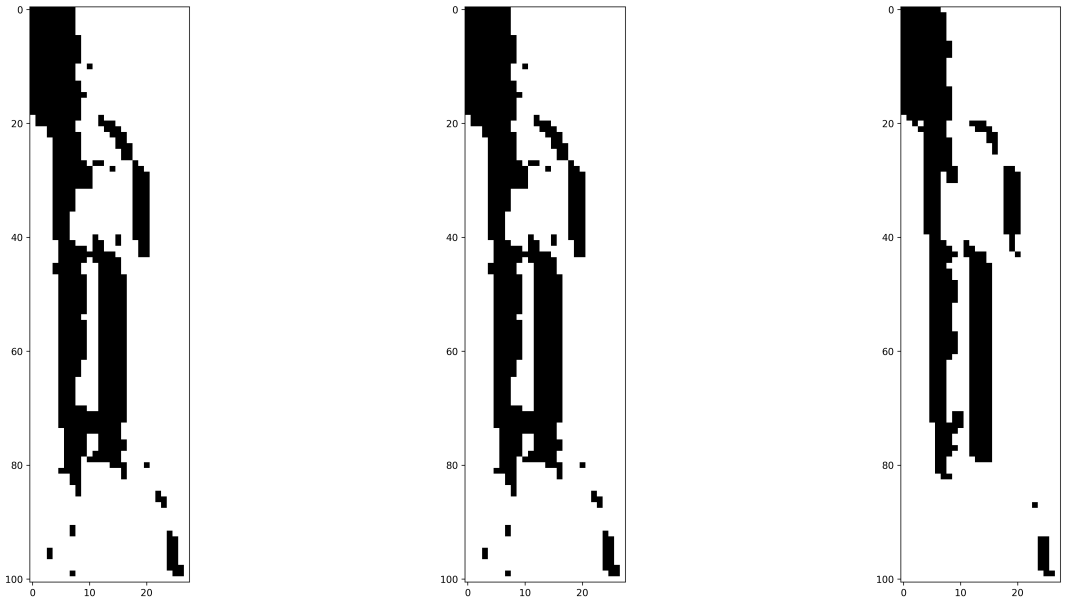
(a) **(b)** **(c)**

(d) **(e)** **(f)**

Obrázek 11: Na obrázku vidíme použití SQUARE FILTER na datasetu `paleo`. Obrázek 11a ukazuje data po ALTERNATING METHOD, obrázek 11b výsledek SQUARE FILTER s hranicí 0,2, obrázek 11c s hranicí 0,3, obrázek 11d s hranicí 0,35, obrázek 11e s hranicí 0,4 a obrázek 11f s hranicí 0,5.



(a) (b) (c)



(d) (e) (f)

Obrázek 12: Na obrázku vidíme použití SQUARE FILTER na datasetu ZOO. Obrázek 12a ukazuje data po ALTERNATING METHOD, obrázek 12b výsledek SQUARE FILTER s hranicí 0,2, obrázek 12c s hranicí 0,3, obrázek 12d s hranicí 0,35, obrázek 12e s hranicí 0,4 a obrázek 12f s hranicí 0,5.

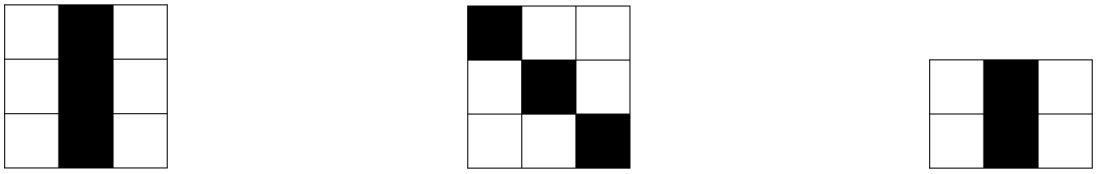
dataset a daná metoda	hodnota prahu				
	0,2	0,3	0,35	0,4	0,5
paleo					
spectral-ordering-pearson-bfp	93,41	95,89	95,70	95,45	95,70
barycenter-bfp	93,63	96,10	96,07	95,69	96,07
alternating	93,93	96,25	96,22	95,93	96,22
barycenter	92,90	95,31	95,41	95,11	95,41
barycenter-bfp-alternating	93,78	96,16	96,06	95,76	96,06
zoo					
spectral-ordering-pearson-bfp	68,21	87,20	80,66	81,61	80,66
barycenter-bfp	79,46	90,70	89,14	90,31	89,14
alternating	80,20	91,27	89,29	90,38	89,29
barycenter	78,64	88,83	87,91	88,26	87,91
barycenter-bfp-alternating	76,27	89,99	86,63	88,44	86,63
healthcare					
spectral-ordering-pearson-bfp	90,50	94,99	93,29	95,32	93,29
barycenter-bfp	95,70	97,02	96,83	97,31	96,83
alternating	92,16	96,36	95,37	95,98	95,37
barycenter	95,18	96,83	96,22	96,88	96,22
barycenter-bfp-alternating	95,65	97,78	97,50	98,02	97,50
mushroom					
spectral-ordering-pearson-bfp	69,28	88,45	82,17	83,29	82,17
barycenter-bfp	80,96	92,61	89,21	90,39	89,21
alternating	81,11	93,41	88,61	90,76	88,61
barycenter	80,45	89,10	88,30	87,75	88,30
barycenter-bfp-alternating	81,89	94,33	90,48	91,38	90,48

Tabulka 2: Podobnost matic po použití SQUARE FILTER

5.4.3 Dilatace a eroze

Filtry DILATACE a EROZE jsme si představili v kapitole 4.3. Jak jsme si již zmínili, důležitou roli tohoto filtru hraje podoba strukturálního elementu B. Standardně se používá velikost 3×3. Tuto velikost budeme používat také a to s jednotkovou maticí (obrázek 13b) a druhá varianta bude s jedničky v prostředním sloupci (obrázek 13a). Vzhledem k tvaru pásové matice, kterého se snažíme dosáhnout, se nabízí matice 3×2 s jedničkami v prostředním sloupci (obrázek 13c).

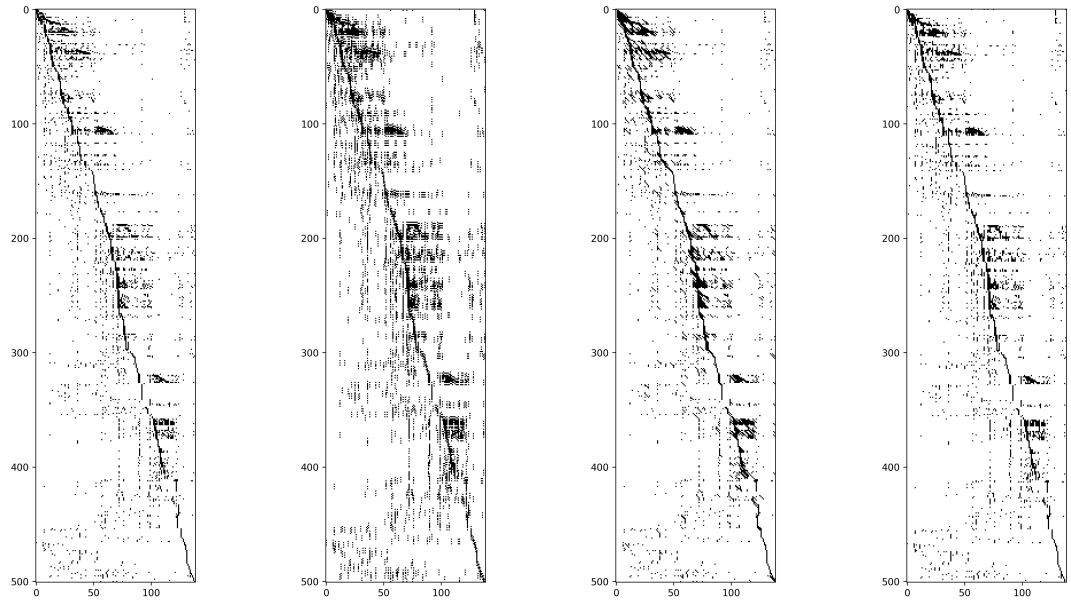
Možné by byly také další varianty strukturálního elementu B, ale výše tři zmíněné se jeví ideální vzhledem k podobě pásové matice.



(a) **(b)** **(c)**

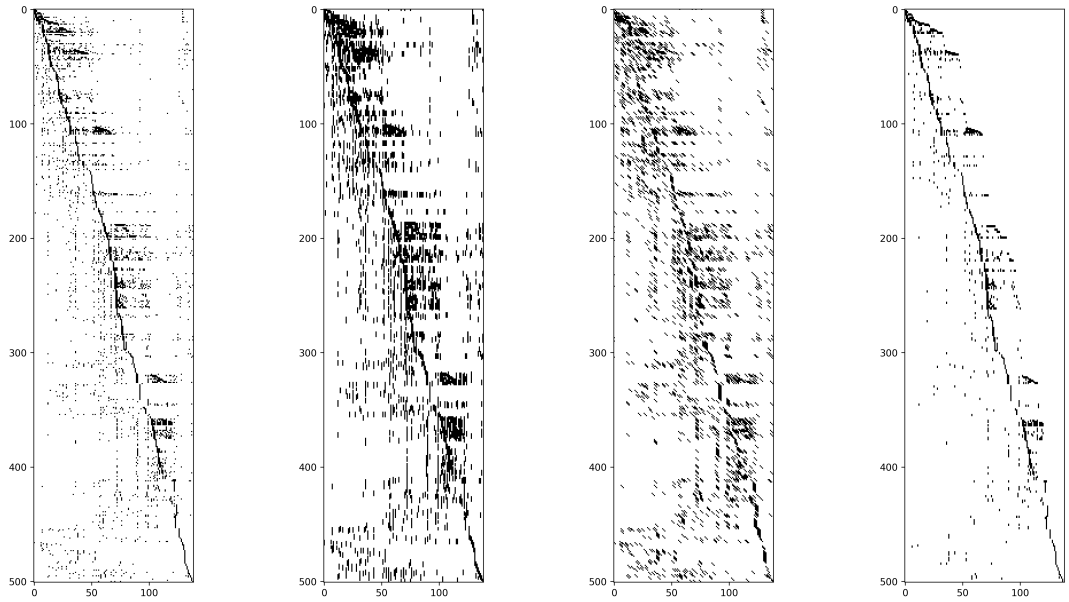
Obrázek 13: Strukturální element pro DILATACI a EROZI

Tyto filtry vzhledem k tomu, že jsou prakticky svými protějšky je vhodné je používat v kombinaci. Vhodné jsou obě dvě pořadí jak DILATACE–EROZE (obrázek 14), tak EROZE–DILATACE (obrázek 15). Na obrázcích 14 a 15 postupně vidíme dataset `paleo` po ALTERNATING METHOD a následně po aplikaci filtru s maskou 13a, 13b a 13c. Na obrázcích 17 a 16 vidíme totéž, ale pro dataset `zoo`.



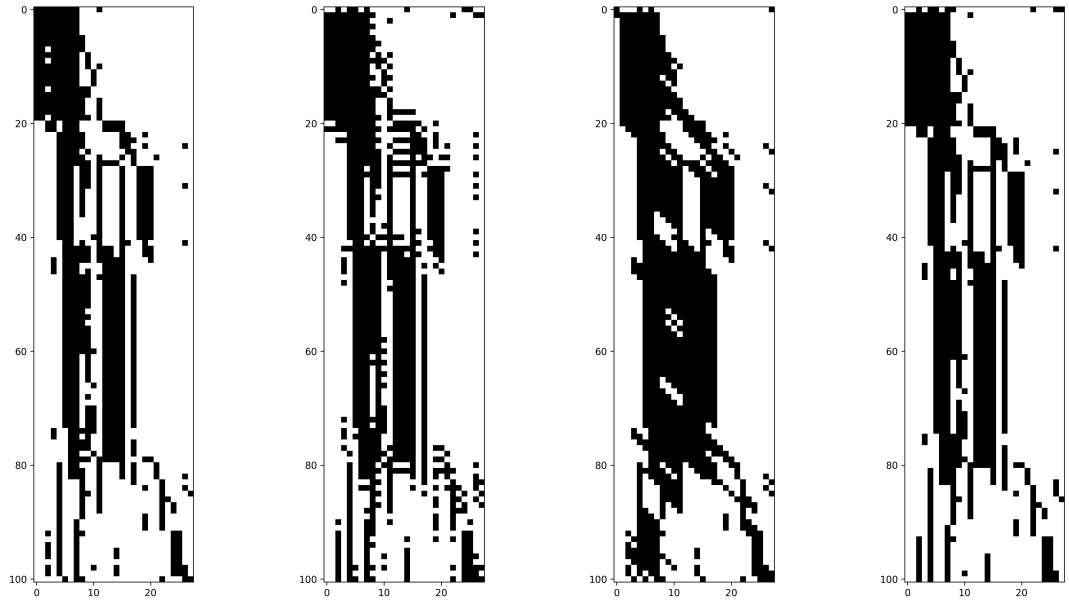
(a) **(b)** **(c)** **(d)**

Obrázek 14: Použití DILATACE–EROZE na dataset `paleo`



(a) (b) (c) (d)

Obrázek 15: Použití EROZE–DILATACE na dataset `paleo`

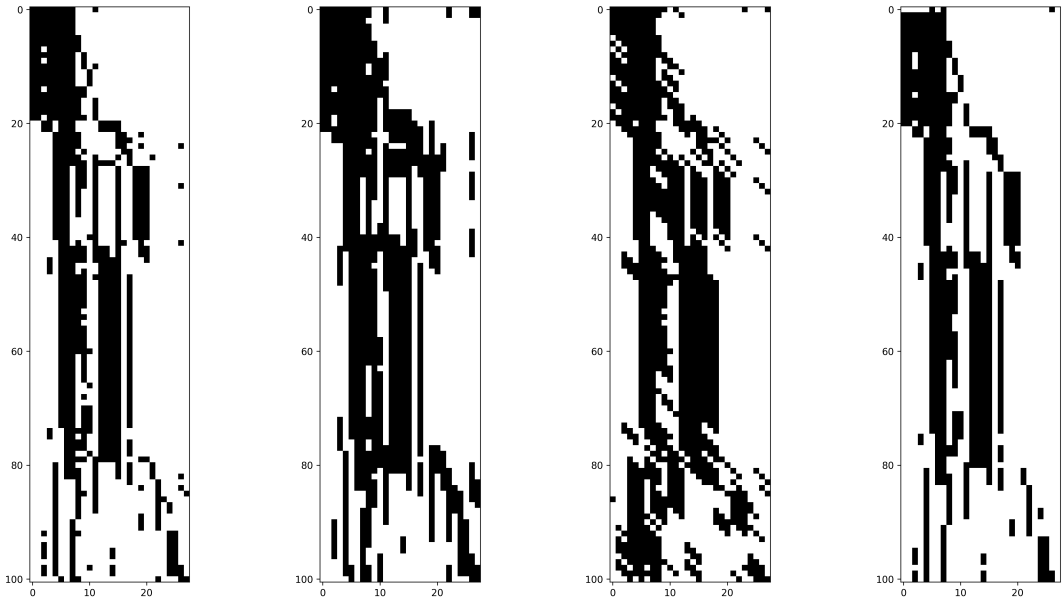


(a) (b) (c) (d)

Obrázek 16: Použití DILATACE–EROZE na dataset `zoo`

Přehled podobnosti matice před a po aplikování filtru DILATACE–EROZE vidíme v tabulce 3 a pro filtr EROZE–DILATACE vidíme v tabulce 4.

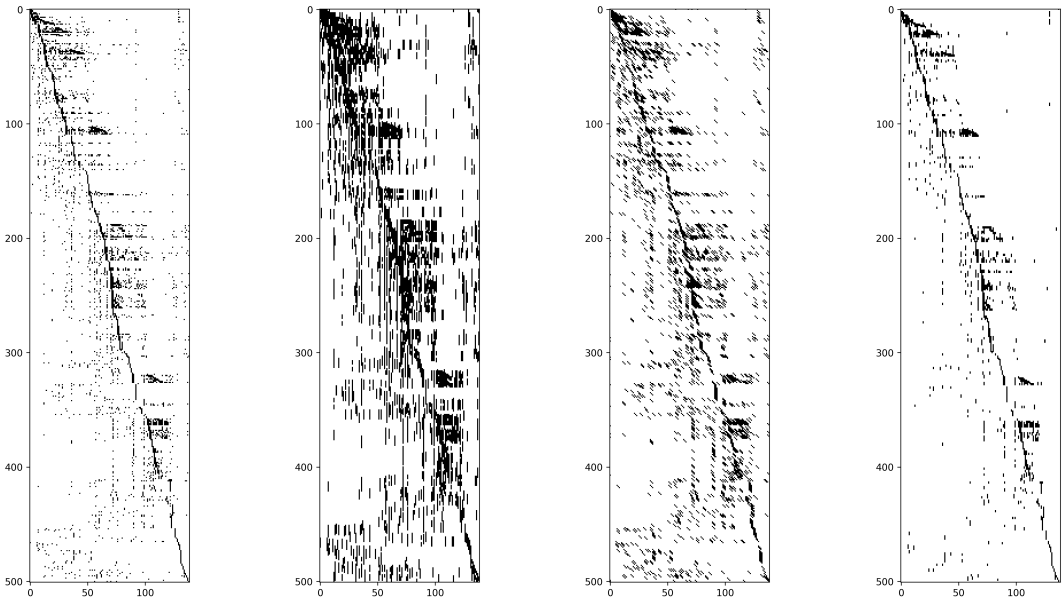
Taktéž jsme experimentovali s kombinací DILATACE–EROZE–EROZE–DILATACE a EROZE–DILATACE–DILATACE–EROZE. Výsledky pro dataset `paleo` pro filtr DILATACE–EROZE–EROZE–DILATACE vidíme na obrázku 18 a pro filtr EROZE–



(a) **(b)** **(c)** **(d)**

Obrázek 17: Použití EROZE–DILATACE na dataset `zoo`

DILATACE–DILATACE–EROZE vidíme na obrázku 19. V tabulkách 5 a 6 pak vidíme podobnost matic před a po aplikaci těchto kombinací filtrů.



(a) **(b)** **(c)** **(d)**

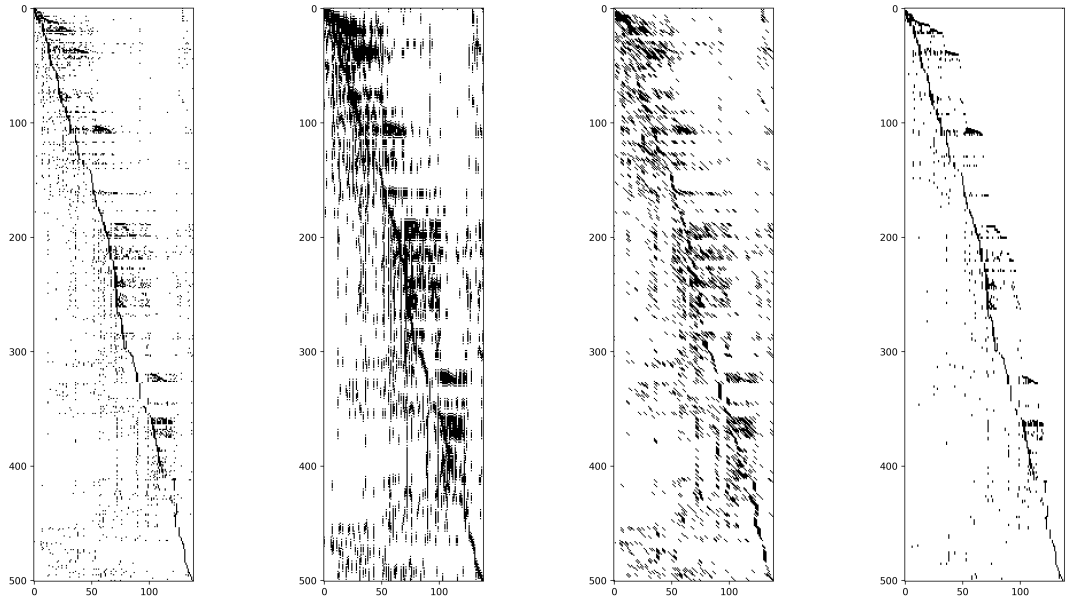
Obrázek 18: Použití DILATACE–EROZE–EROZE–DILATACE na dataset `paleo`

	1	2	3
paleo			
spectral-ordering-pearson-bfp	89,38	96,07	91,94
barycenter-bfp	94,55	94,08	98,85
alternating	94,83	94,43	98,96
barycenter	92,20	95,33	88,03
barycenter-bfp-alternating	92,81	96,06	89,53
zoo			
spectral-ordering-pearson-bfp	85,64	91,97	68,56
barycenter-bfp	94,38	93,88	90,52
alternating	93,85	93,49	89,67
barycenter	83,38	91,34	86,60
barycenter-bfp-alternating	78,04	93,67	88,40
healthcare			
spectral-ordering-pearson-bfp	89,18	91,12	94,80
barycenter-bfp	94,05	94,23	94,85
alternating	94,38	93,71	95,94
barycenter	96,12	93,29	92,44
barycenter-bfp-alternating	96,88	93,34	92,91
mushroom			
spectral-ordering-pearson-bfp	86,57	92,20	69,63
barycenter-bfp	96,50	96,06	92,50
alternating	97,19	96,72	92,55
barycenter	82,20	92,26	86,74
barycenter-bfp-alternating	83,36	97,02	95,28

Tabulka 3: Podobnost matic po použití filtru DILATACE–EROZE. Sloupec 1 značí strukturální element col-matrix 3×3, sloupec 2 značí col-matrix 3×2 a sloupec 3 značí unit-matrix 3×3.

	1	2	3
paleo			
spectral-ordering-pearson-bfp	94,52	94,17	98,95
barycenter-bfp	89,25	96,03	92,89
alternating	89,92	96,13	93,12
barycenter	98,31	93,00	93,87
barycenter-bfp-alternating	98,94	94,20	94,58
zoo			
spectral-ordering-pearson-bfp	91,80	91,02	86,49
barycenter-bfp	90,31	94,06	84,34
alternating	89,25	94,02	85,33
barycenter	89,14	91,09	92,47
barycenter-bfp-alternating	91,55	93,07	93,95
healthcare			
spectral-ordering-pearson-bfp	93,29	92,20	94,94
barycenter-bfp	92,91	93,71	96,12
alternating	91,45	92,67	95,42
barycenter	93,81	93,76	94,05
barycenter-bfp-alternating	94,52	93,86	94,42
mushroom			
spectral-ordering-pearson-bfp	92,21	90,91	92,45
barycenter-bfp	94,27	96,34	82,80
alternating	95,03	97,12	81,70
barycenter	92,06	91,10	92,33
barycenter-bfp-alternating	94,37	96,85	97,25

Tabulka 4: Podobnost matic po použití filtru EROZE–DILATACE. Sloupec 1 značí strukturální element col-matrix 3×3, sloupec 2 značí col-matrix 3×2 a sloupec 3 značí unit-matrix 3×3.



(a) **(b)** **(c)** **(d)**

Obrázek 19: Použití EROZE–DILATACE–DILATACE–EROZE na dataset `paleo`

	1	2	3
paleo			
spectral-ordering-pearson-bfp	81,76	94,59	93,01
barycenter-bfp	82,01	94,48	94,14
alternating	83,26	94,65	94,07
barycenter	93,80	93,86	82,70
barycenter-bfp-alternating	93,82	94,54	82,35
zoo			
spectral-ordering-pearson-bfp	77,19	88,19	74,54
barycenter-bfp	83,52	90,56	85,61
alternating	81,47	91,27	86,78
barycenter	85,08	87,23	79,28
barycenter-bfp-alternating	83,73	89,92	79,60
healthcare			
spectral-ordering-pearson-bfp	82,04	87,76	93,29
barycenter-bfp	83,32	91,07	92,01
alternating	85,78	90,74	93,24
barycenter	91,92	90,97	84,59
barycenter-bfp-alternating	92,44	91,07	84,40
mushroom			
spectral-ordering-pearson-bfp	83,92	90,51	75,44
barycenter-bfp	93,14	94,90	87,27
alternating	94,15	95,91	85,61
barycenter	86,74	90,33	83,60
barycenter-bfp-alternating	87,05	95,69	94,35

Tabulka 5: Podobnost matic při použití filtru DILATACE–EROZE–EROZE–DILATACE. Sloupec 1 značí strukturální element col-matrix 3×3, sloupec 2 značí col-matrix 3×2 a sloupec 3 značí unit-matrix 3×3.

	1	2	3
paleo			
spectral-ordering-pearson-bfp	85,61	94,72	91,35
barycenter-bfp	85,66	94,68	92,34
alternating	86,54	94,75	92,60
barycenter	91,36	94,11	85,02
barycenter-bfp-alternating	92,30	94,71	85,92
zoo			
spectral-ordering-pearson-bfp	81,22	87,80	62,55
barycenter-bfp	86,63	90,74	79,24
alternating	85,01	90,91	81,19
barycenter	79,60	87,31	82,71
barycenter-bfp-alternating	74,05	89,96	83,73
healthcare			
spectral-ordering-pearson-bfp	84,97	87,76	92,20
barycenter-bfp	87,52	91,07	91,49
alternating	87,90	90,36	91,97
barycenter	91,68	90,97	87,71
barycenter-bfp-alternating	92,25	91,07	87,90
mushroom			
spectral-ordering-pearson-bfp	84,62	90,69	65,10
barycenter-bfp	93,50	94,99	80,23
alternating	94,35	96,01	78,83
barycenter	79,44	90,52	84,58
barycenter-bfp-alternating	80,55	95,73	94,61

Tabulka 6: Podobnost matic při použití filtru EROZE–DILATACE–DILATACE–EROZE. Sloupec 1 značí strukturální element col-matrix 3×3, sloupec 2 značí col-matrix 3×2 a sloupec 3 značí unit-matrix 3×3.

5.5 Výsledky faktorizace

V předchozí kapitole jsme pomocí grafických filtrů pozměnili původní data. Z tohoto důvodu nás teď bude zajímat, jaký vliv měly tyto změny na faktorizaci.

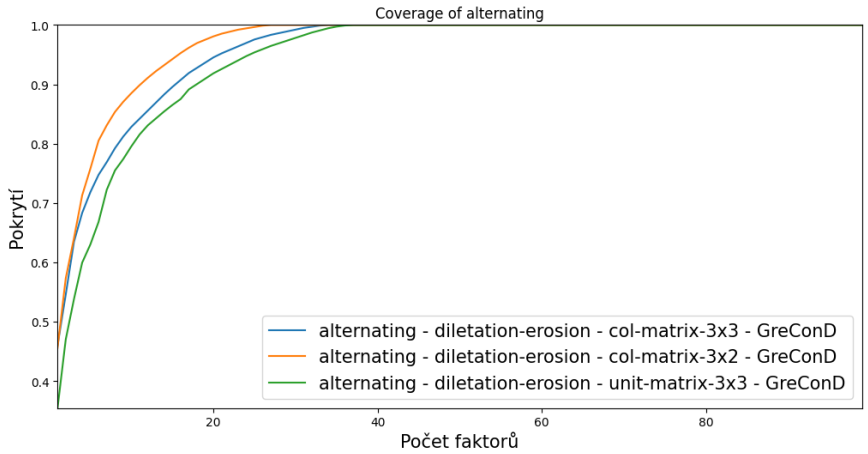
5.5.1 GreConD faktorizace

Při dekompozici algoritmem GreConD jsme porovnávali výsledek nad upravenými daty s výsledky nad původními daty. Upravená data se od původních příliš neliší. Určeme míru pokrytí výsledných matic. Pokrytí matice I maticí $A \circ B$ získáme:

$$1 - E(I, A \circ B)/||I||,$$

kde $E(I, A \circ B)$ značí chybu z kapitoly 2.1.1.

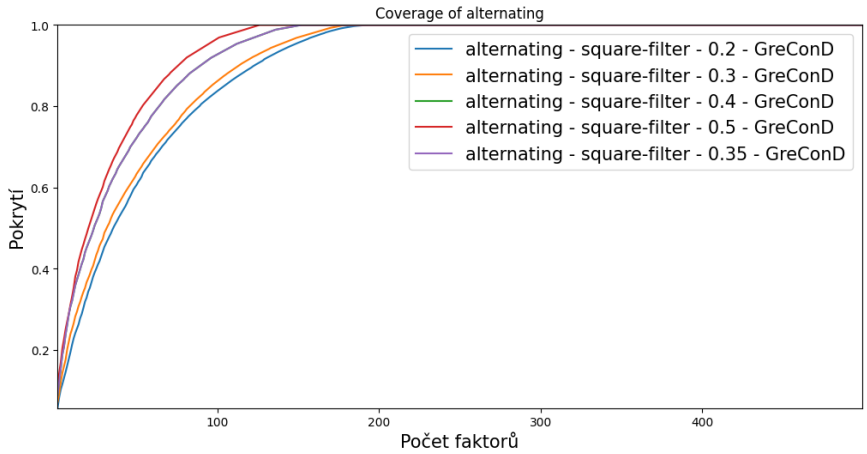
Kompletní pokrytí pro jednotlivé filtry a metody nalezneme v příloze. Pro metodu alternating a filtr dilatace–eroze na datasetu zoo vidíme graf pokrytí na obrázku 20. Z grafu vidíme, že pokrytí je velmi vysoké.



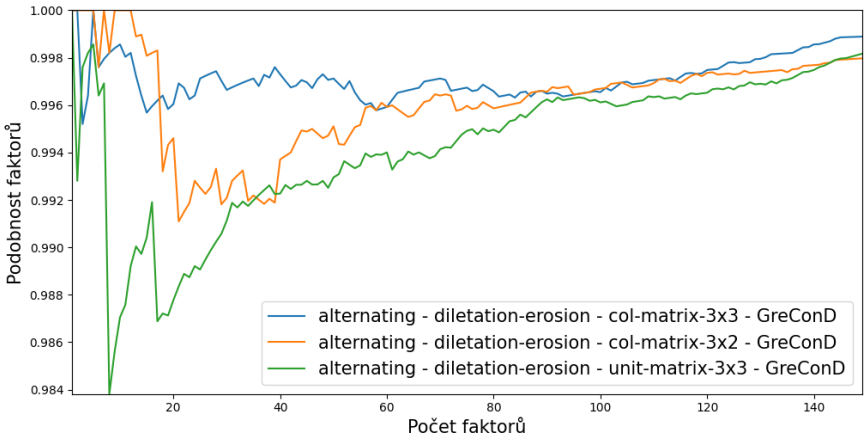
Obrázek 20: Pokrytí alternating metody s filtrem dilatace–eroze datasetu zoo při dekompozici GreConD algoritmem

Důležitou součástí je podobnost jednotlivých faktorů, které vypočítal GreConD před a po aplikaci filtru. Pro porovnání podobnosti faktorů jsme využili Simple matching coefficient (SMC). Z výsledků vyplývá, že podobnost faktorů je obecně velmi vysoká, jak můžeme vidět z ukázkových grafů 22 a 23.

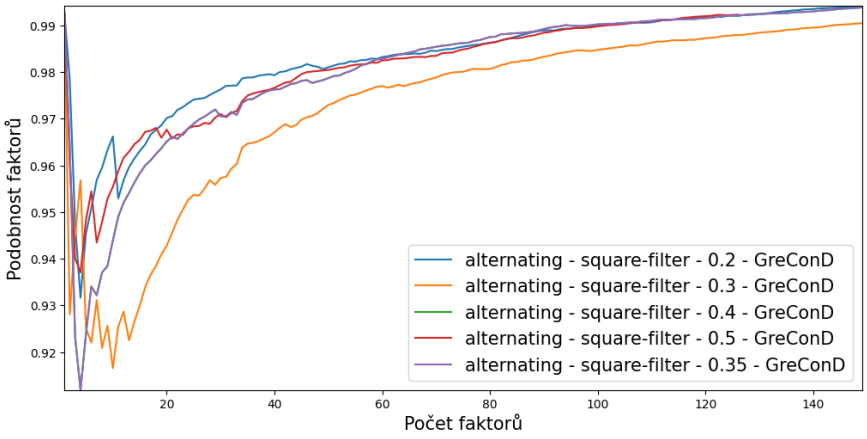
Zajímavé chování ukazuje počet faktorů, který algoritmu GreConD při faktorizaci vypočetl. Ukázalo se, že se velmi často počet faktorů při úpravě dat zvětší. Přehled můžeme vidět v tabulce 7. Pro jednotlivé datasety a na ně použité metody pro získání pásové struktury vidíme počet faktorů u jednotlivých filtrů a jejich argumentů.



Obrázek 21: Pokrytí ALTERNATING metody s filtrem SQUARE FILTER datasetu `zoo` při dekompozici GRECOND algoritmem



Obrázek 22: Podobnost faktorů získaných dekompozicí algoritmem GRECOND matic upravených filtrem DILATACE–EROZE na datasetu `paleo` po metodě ALTERNATING



Obrázek 23: Podobnost faktorů získaných dekompozicí algoritmem GRECOND matic upravených filtrem SQUARE FILTER na datasetu `paleo` po metodě ALTERNATING

	Deleted band				square filter					E-D[a]			D-E			D-E-E-D			E-D-D-E		
	30%	50%	70%	90%	0.2	0.3	0.35	0.4	0.5	1[b]	2[c]	3[d]	1	2	3	1	2	3	1	2	3
zoo																					
alternating	29	30	30	29	38	34	31	31	25	33	26	40	35	28	38	38	30	39	34	26	39
barycenter	30	30	31	33	36	34	36	36	28	39	30	32	38	27	38	38	28	32	39	24	35
barycenter-bfp	31	32	28	33	39	31	34	34	28	36	29	38	37	29	37	35	28	40	36	26	38
bary.-bfp-alter.	31	31	30	26	38	35	32	32	28	39	30	31	38	29	32	38	28	34	39	27	33
spectral-ordering	31	29	27	25	44	33	28	28	21	33	31	37	35	26	39	34	28	39	34	30	41
paleo																					
alternating	149	148	148	141	194	182	153	153	127	174	155	142	156	155	151	185	160	145	176	156	142
barycenter	150	150	150	154	195	176	131	131	62	175	144	144	149	151	168	145	145	206	150	132	191
barycenter-bfp	150	150	150	146	195	176	162	162	126	175	144	144	157	152	149	196	154	156	180	145	143
bary.-bfp-alter.	151	150	146	132	194	180	152	152	122	157	151	158	142	147	175	145	149	183	144	144	180
spectral-ordering	149	147	148	137	184	163	125	125	88	155	154	146	169	149	140	187	157	140	176	146	141
healthcare																					
alternating	23	40	43	39	20	15	12	12	11	16	12	29	19	13	23	12	10	30	13	11	35
barycenter	26	33	41	52	23	19	19	19	17	27	14	18	33	12	18	27	11	17	35	11	16
barycenter-bfp	26	32	40	52	25	18	19	19	17	15	13	36	18	14	23	14	12	30	17	12	35
bary.-bfp-alter.	25	34	41	52	22	19	19	17	16	24	13	16	36	12	17	30	10	12	37	10	14
spectral-ordering	26	39	47	39	19	20	17	17	14	15	13	26	18	12	31	10	10	29	11	10	33
mushroom																					
alternating	123	127	117	91	165	155	143	143	129	164	118	160	155	123	155	161	123	163	162	117	158
barycenter	114	128	124	134	166	156	137	137	104	156	135	154	149	116	160	157	130	174	153	114	167
barycenter-bfp	115	132	122	115	175	156	151	151	127	158	127	162	150	142	158	165	147	167	162	129	169
bary.-bfp-alter.	120	124	118	90	176	157	137	137	130	156	120	153	163	119	155	164	123	158	159	116	159
spectral-ordering	125	128	109	95	144	139	39	39	30	152	138	144	162	118	155	170	127	151	168	125	156

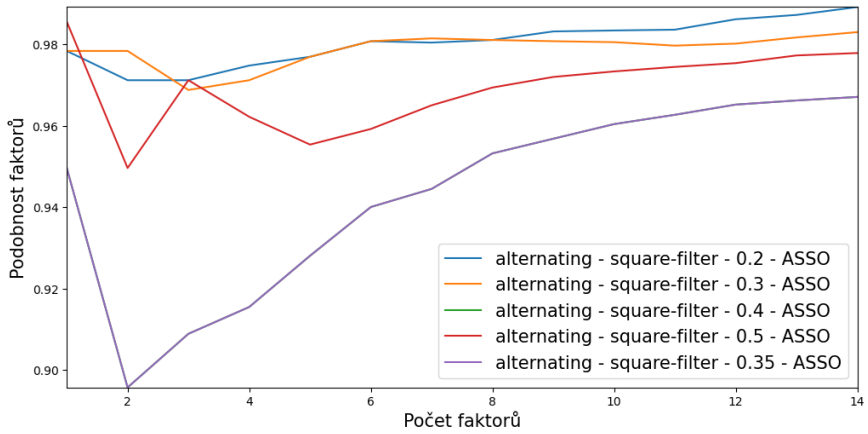
Tabulka 7: Přehled počtu faktorů vypočtených pomocí GreConD

[a] E = eroze, D = dilatace
[b] 3×3 col matrix
[c] 3×2 col matrix
[d] 3×3 unit matrix

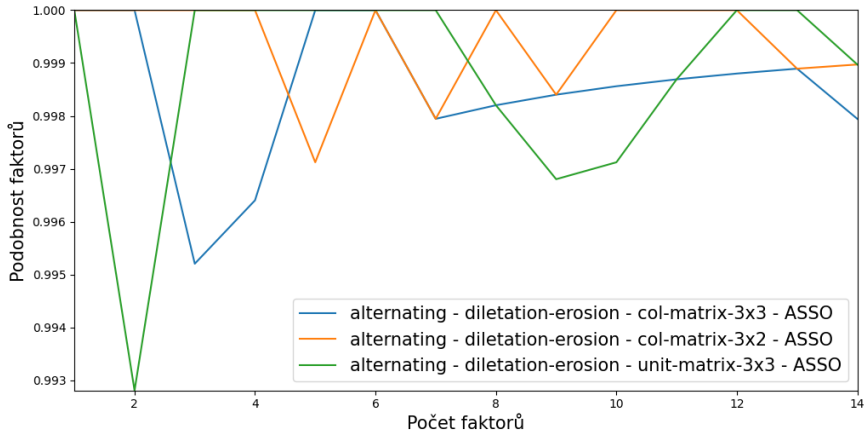
5.5.2 ASSO faktorizace

Abychom zjistili jaký vliv měly změny dat na podobu faktorů, použila jsem ještě algoritmus ASSO na upravené matice. Algoritmus jsem spouštěla s hodnotou $\tau = 0,9$ a $w^+ = 1, w^- = 1$. Počty faktorů jsem zvolila 5, 10 a 15. Stejně jako u algoritmu GreConD jsem vypočítala grafy pokrytí, které naleznete v příloze. Příklad takového grafu je na obrázku 26, kde vidíme pokrytí pro dataset `paleo` s metodou ALTERNATING a filtrem DILATACE–EROZE pro 5 (obrázek 26a), 10 (obrázek 26b) a 15 (obrázek 26c) faktorů. Z grafů 26 jde vidět, že pokrytí rychle stoupá.

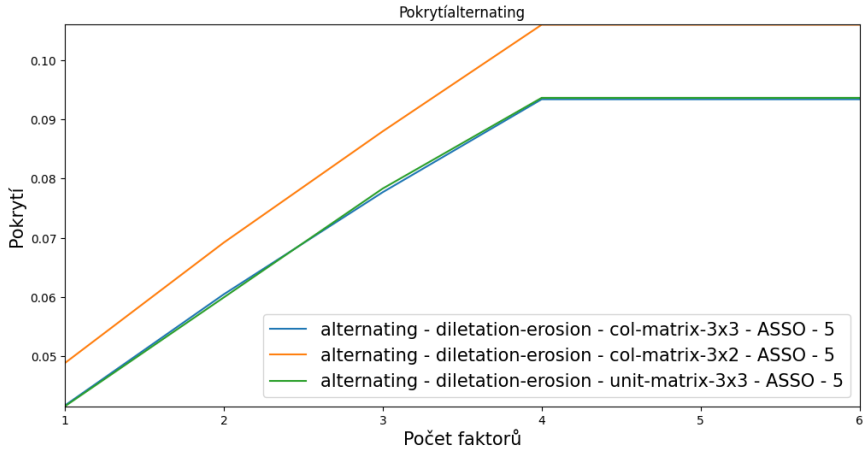
Taktéž pro algoritmus ASSO jsem počítala podobnost faktorů. Příklad grafu podobnosti vidíme na obrázku 24 a 25. Stejně jako u algoritmu GreConD je podobnost velmi vysoká. Jednotlivé faktory i po aplikaci filtrů jsou velmi podobné, což značí, že úprava dat neovlivnila faktory.



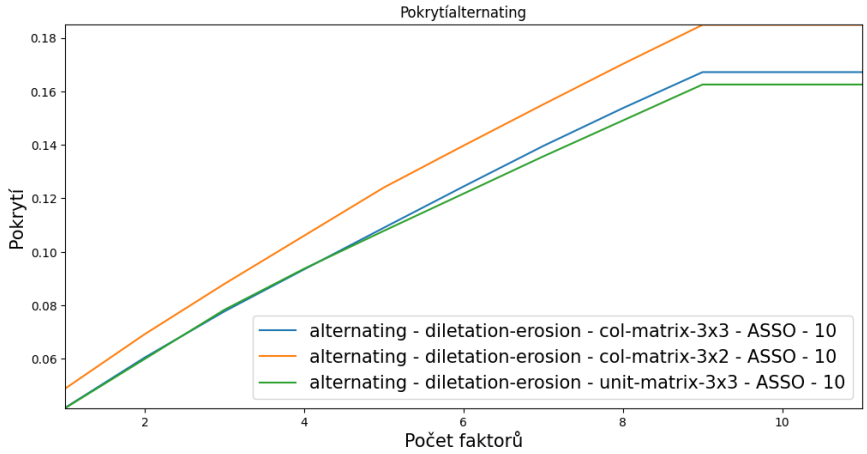
Obrázek 24: Podobnost faktorů získaných dekompozicí algoritmem ASSO matic upravených filtrem SQUARE FILTER na datasetu `paleo` po metodě ALTERNATING



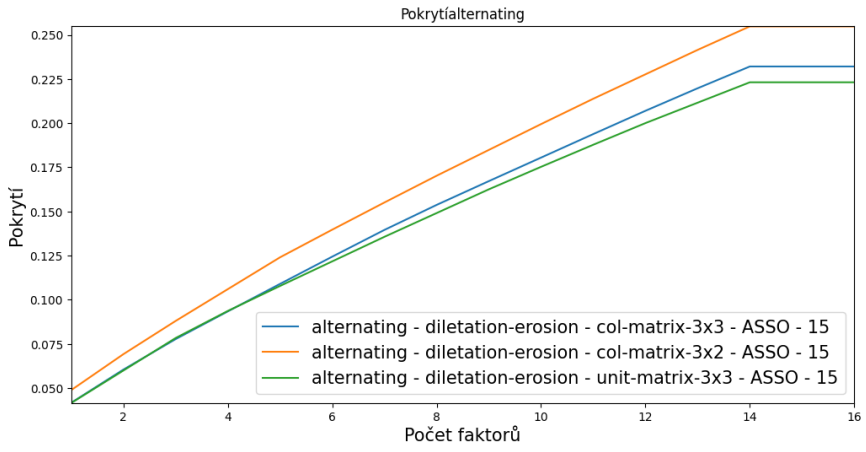
Obrázek 25: Podobnost faktorů získaných dekompozicí algoritmem ASSO matic upravených filtrem DILATACE–EROZE na datasetu `paleo` po metodě ALTERNATING



(a)



(b)



(c)

Obrázek 26: Grafy pokrytí DILATACE–EROZE na dataset `paleo`–alternating metoda při použití algoritmu ASSO

Závěr

V teoretické části této práce je popsána dekompozice binárních matic společně s pásovou strukturou. Následně bylo ukázáno několik algoritmů pro získání pásové struktury, se kterými jsme experimentovali a porovnávali jsme je. Na data v pásové struktuře bylo využito několik filtrů z počítačové grafiky. Poté jsme se zabývali vlivem takovéto úpravy dat na dekompozici binárních matic.

Z experimentálního porovnávání vyplývá, že úprava dat jen nepatrně ovlivňuje vypočítané faktory z pohledu DBP. To mimo jiné znamená, že úprava dat jen minimálně ovlivňuje nejdůležitější faktory. Z pohledu AFP narůstá celkový počet faktorů potřebných pro přesný rozklad. Toto chování je očekávané, jelikož v experimentech nebyl explicitně rozlišován aditivní a subtraktivní šum.

Celkově nám experimenty ukazují, že předzpracování dat pomocí grafických filtrů má smysl a umožňuje odstranit méně důležité informace v datech a zaměřit se tedy pouze na DBP.

Výstupem praktické části práce je implementace všech zmíněných algoritmů v jazyce Python, kompletní přehled výsledků porovnání algoritmů.

V budoucnu by bylo možné na práci navázat s podrobnějším zkoumáním jednotlivých metod pro získání pásové struktury. Dalším možným směrem je využití dalších filtrů z počítačové grafiky a rozlišování aditivního subtraktivního šumu. Taktéž se nabízí porovnat získané faktorizace na jednotlivých datasetech a interpretovat jejich konkrétní podobu.

Conclusions

The theoretical part of this thesis describes the decomposition of binary matrices together with the banded structure. Subsequently, several algorithms for obtaining the band structure are presented, with which we experimented with and compared. Several computer graphics filters were applied to the data in the banded structure. We then examined the effect of such data modifications on the decomposition of binary matrices.

From the experimental comparison, it follows that data modification only slightly affects the calculated factors from the DBP point of view. This means, among other things, that data modification only minimally affects the most important factors. From AFP point of view, the total number of factors required for accurate decomposition is increasing. This behavior is expected, since additive and subtractive noise were not explicitly distinguished in the experiments.

Overall, the experiments show us that preprocessing the data using graphic filters makes sense and allows us to remove less important information in the data and thus focus only on the DBP.

The practical part of the thesis includes the implementation of all mentioned algorithms in Python and a comprehensive overview of the comparison results of these algorithms.

In the future, it would be possible to continue the work with a more detailed investigation of individual methods for obtaining the banded structure. Another possible direction is the use of additional filters from computer graphics and the differentiation of additive subtractive noise. It is also offered to compare the obtained factorizations on individual datasets and to interpret their specific form.

A Obsah elektronických dat

Součástí práce je elektronická příloha dat v systému katedry informatiky s následující strukturou:

`text/`
: Adresář obsahuje text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce. Obsahuje také ZIP soubor obsahující všechny soubory potřebné pro bezproblémové vytvoření tohoto PDF dokumentu.

`README.md`
: Příručka k elektronickým datům práce. Obsahuje popis kroků, které je potřeba provést před spuštěním kódu.

`src/`
: Adresář obsahuje veškeré moduly vytvořené pro tuto práci. Také obsahuje adresář data s výsledky experimentů a adresář web s webovým zobrazením těchto výsledků.

Literatura

[1] Gemma C. Garriga Esa Junttila, Heikki Mannila. Banded Structure in Binary Matrices. *Springer-Verlag London Limited*. 2010.

[2] P., Miettinen. *Matrix Decomposition Methods for Data Mining: Compu- tational Complexity and Algorithms*. 2009.

[3] Bělohlávek R., Trnečka M. From-Below Approximations in Boolean Matrix Factorization: Geometry and New Algorithm. *Journal of Computer and System Sciences 81 (8)*. 2015, s. 1678–1697.

[4] Belohlavek R., Vychodil V. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences*. 2010, s. 3–20.

[5] M., Trnečka. *Decompositions of matrices with relational data: foundations and algorithms*. 2016.

[6] R., Bělohlávek. Konceptuální svazy a formální konceptuální analýza. 2004. Dostupný také z: ⟨http://belohlavek.inf.upol.cz/publications/Bel_Ksfka.pdf⟩.

[7] Miettinen P. Mielik¨ainen T., Gionis A. et al. The discrete basis problem. *IEEE transactions knowledge and data engineering*. 2008, s. 1348–1362.

[8] Atkins J Boman E, Hendrickson B. A spectral algorithm for seriation and the consecutive ones problem. *SIAM J Comput*. 1999, s. 297–310.

[9] Sugiyama K Tagawa S, Toda M. Methods for visual understanding of hierarchical system structures. *IEEE Trans Syst Man Cybern*. 1981, s. 109–125.

[10] Rafael C. Gonzalez, Richard E. Woods. *Digital Image Processing*. Third. 2008.

[11] Harris, Charles R.; Millman, K. Jarrod; Walt, Stéfan J. van der aj. Array programming with NumPy. *Nature*. 2020, roč. 585, č. 7825, s. 357–362. Dostupný také z: ⟨https://doi.org/10.1038/s41586-020-2649-2⟩.

[12] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007, roč. 9, č. 3, s. 90–95. Dostupný také z: ⟨http://dx.doi.org/10.1109/MCSE.2007.55⟩.

[13] Forsyth, Richard. *Zoo*. 1990. UCI Machine Learning Repository⟨http://dx.doi.org/10.24432/C5R59V⟩.

[14] 2023, The NOW Community. *New and Old Worlds Database of Fossil Mammals (NOW)*. Retrieved [2023] from https://nowdatabase.org/now/database/. UCI Machine Learning Repository⟨http://dx.doi.org/doi:10.5281/zenodo.4268068⟩.

[15] Lichman, M. *UCI Machine Learning Repository*. 2013. Dostupný také z: ⟨http://archive.ics.uci.edu/ml.⟩.

[16] Ene, Alina; Horne, William; Milosavljevic, Nikola aj. Fast Exact and Heuristic Methods for Role Minimization Problems. In. *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*. Estes Park, CO, USA: Association for Computing Machinery, 2008, s. 1–10. SACMAT '08. Dostupný také z: ⟨https://doi.org/10.1145/1377836.1377838⟩. ISBN 9781605581293.