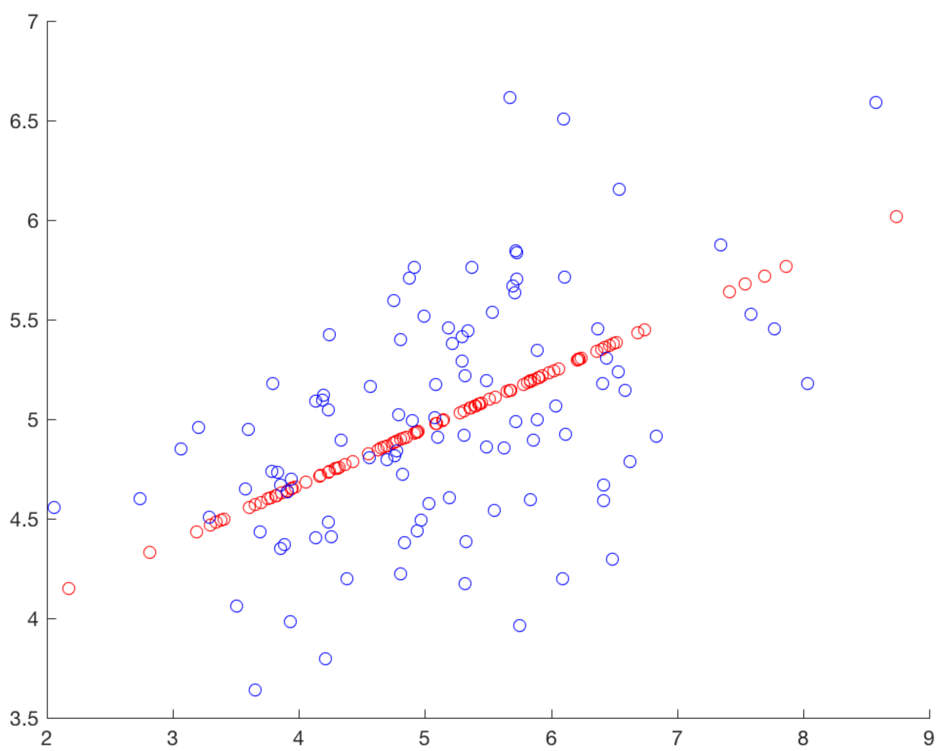
Motion capture

Lukáš Hromadník

1 Úkol 1: Aproximace bodů přímkou

Je dáno m bodů v rovině $a_1, \dots, a_m \in \mathbb{R}^2$. Najděte přímku v rovině (tedy afinní podprostor dimenze 1 prostoru $\mathbb{R}^2$) takovou, aby součet čtverců kolmých vzdáleností bodů k této přímce byl minimální. Představte si např., že někdo body naklikal myší v grafickém rozhraní a vaším úkolem je proložit jimi nejlepší přímku.

1. Zobrazte do jednoho obrázku zadané body $a_1, \dots, a_m$ (modře) a jejich kolmé projekce $\tilde{a}_1, \dots, \tilde{a}_m$ na nalezenou přímku (červeně).



2. Jaký je součet čtverců kolmých vzdáleností bodů k nalezené přímce?

   S využitím spektrálního rozkladu vstupní matice A, nalezneme součet vzdáleností jako součet prvních k vlastních čísel matice A. Námi proložený podprostor je dimenze 1, tedy součet vzdáleností je nejmenší vlastní číslo.

   Výsledek je 24.2419.

3. Nalezněte požadovanou přímku ve dvou různých reprezentacích:

$$\left\{\mathbf{y} \in \mathbb{R}^2 \mid \mathbf{y}^T\mathbf{x} = \alpha\right\} = \{\mathbf{y}_0 + t\mathbf{s} \mid t \in \mathbb{R}\}$$

   Pro obecnou rovnici přímky $\left\{\mathbf{y} \in \mathbb{R}^2 \mid \mathbf{y}^T\mathbf{x} = \alpha\right\}$ stačí vzít první vlastní vektor matice A. Vlastní vektory jsou navzájem kolmé a mají jednotkovou velikost. První vlastní vektor tedy tvoří ortogonální doplněk k podprostoru, jehož bázi tvoří druhý vlastní vektor. Lze ho tedy použít jako normálový vektor přímky.

   K vypočtení parametru α stačí vzít libovolný bod patřící do podprostoru (např. těžiště) a z rovnice $\mathbf{y}^T \cdot \mathbf{x} = \alpha$ získáme α.

   Po provedení výpočtu získáme

$$\alpha = -3.3931$$

   K vypočtení parametrické rovnice přímky nám stačí bod, který náleží přímce, a směrový vektor přímky. Jako bod můžeme použít těžiště a směrový vektor je stejný jako báze daného podprostoru, tedy $\mathbf{y}_0$ je těžiště a

$$\mathbf{s} = [-0.9617, -0.2740]\,.$$

   $\mathbf{s}$ je vlastní vektor z matice V, má tedy jednotkovou délku.

2 Úkol 2: Komprese sekvence z motion capture

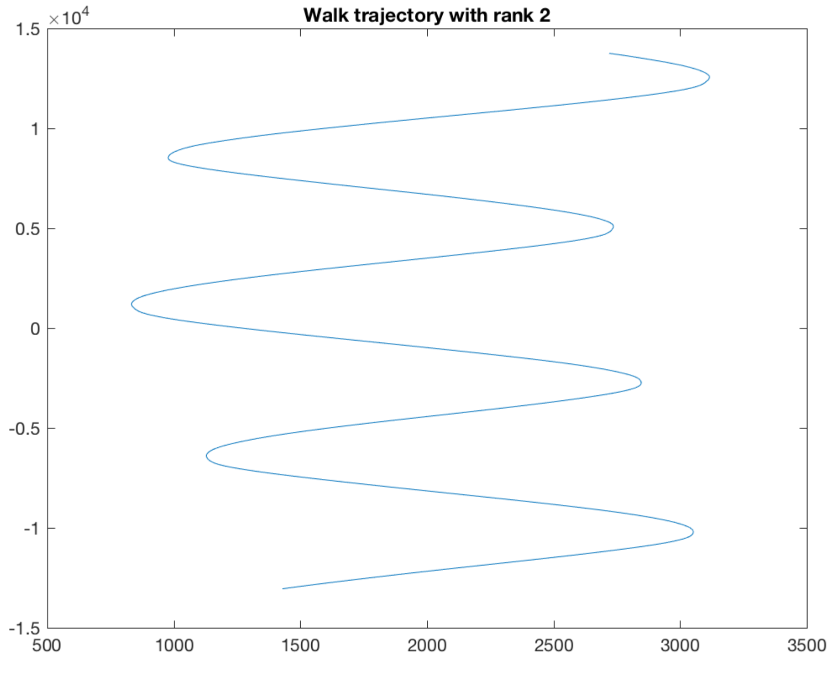
1. Minimalizujte kritérium

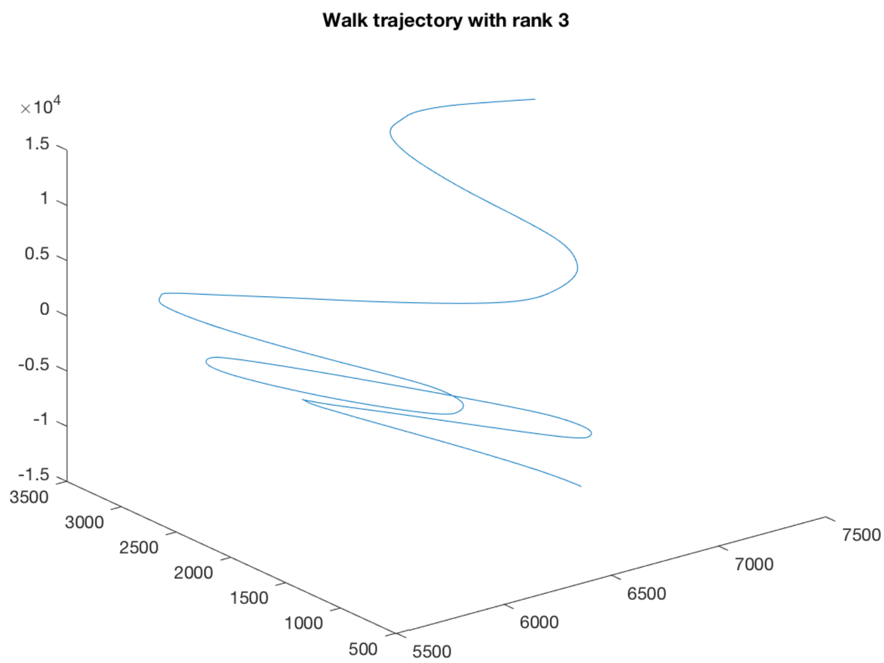
$$\sum_{i=1}^{m} \|\tilde{\mathbf{a}}_i - \mathbf{a}_i\|^2$$

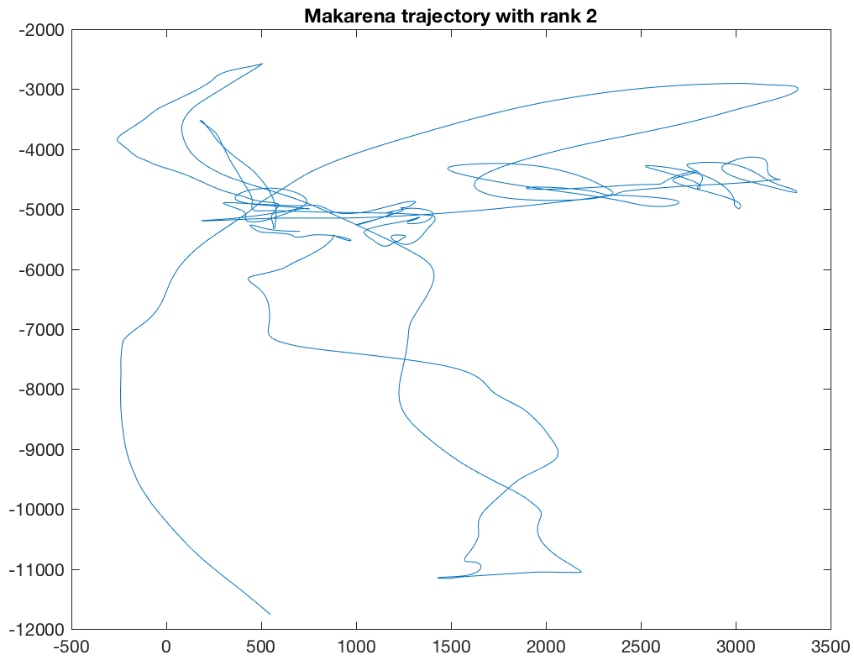
za podmínky, že body $\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_m$ leží v afinním podprostoru dimenze r. Výsledkem bude matice $\tilde{\mathbf{A}}$ s řádky $\tilde{\mathbf{a}}_1^T, \ldots, \tilde{\mathbf{a}}_m^T$. Proveďte pro sekvenci „Chůze“ a pro pět různých hodnot $r \in \{1, 2, 5, 10, 15\}$.

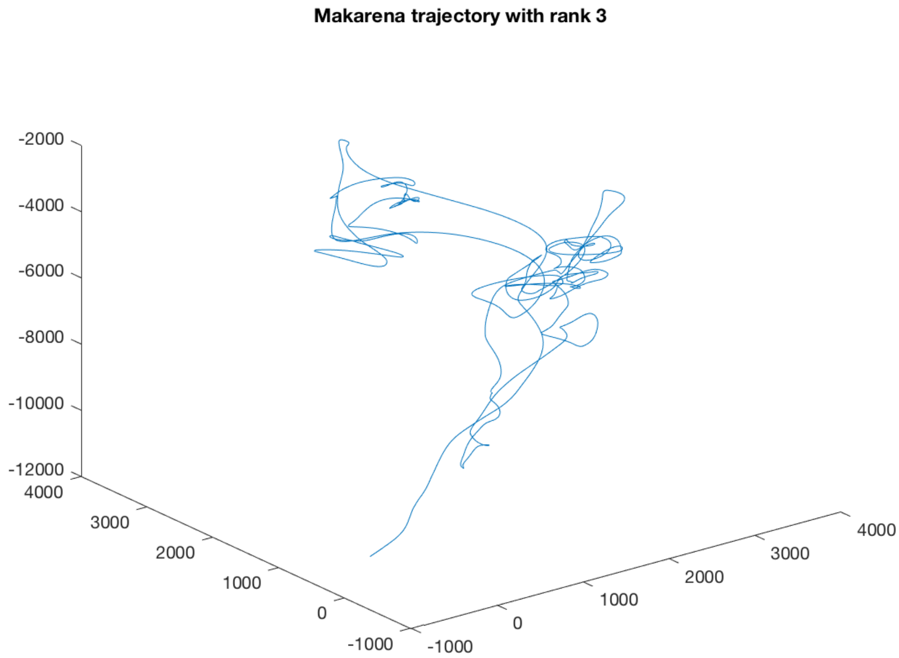
r	1	2	5	10	15
kritérium	4.6166e+08	1.6925e+08	1.0453e+07	1.1982e+06	2.5626e+05

2. Výsledné body vyjádřete jako lineární kombinace $\tilde{\mathbf{a}} = \tilde{\mathbf{V}}\mathbf{y}$ bázových vektorů. Pro $r = 2$ nakreslete sekvenci vektorů $\mathbf{y}_1, \ldots, \mathbf{y}_m$ jako trajektorii v rovině. To samé udělejte i ve třírozměrném prostoru, tedy pro $r = 3$. Proveďte pro sekvence „Tanec Makarena“ a „Chůze“.




Walk trajectory with rank 3
×10^4
1.5
1
0.5
0
-0.5
-1
-1.5
3500
3000
2500
2000
1500
1000
500
5500
6000
6500
7000
7500


Makarena trajectory with rank 2
-2000
-3000
-4000
-5000
-6000
-7000
-8000
-9000
-10000
-11000
-12000
-500
0
500
1000
1500
2000
2500
3000
3500



3. Uvažujte, že postava dělá čistý translační pohyb, tj. konfigurace bodů se nemění a jejich souřadnice se pohybují po přímce. Jaká je minimální dimenze podprostoru, aby aproximační chyba byla nulová?

   Jelikož se body pohybují po přímce, budou se měnit pouze jedna souřadnice a zbytek zůstane neměnný. Z toho výplývá, že dimenze podprostoru bude stejná jako dimenze přímky, tedy 1.

4. Dejme tomu, že bychom chtěli spočítat optimální chybu aproximace

$$\sum_{i=1}^{m} \|\tilde{\mathbf{a}}_i - \mathbf{a}_i\|^2$$

   pro různé hodnoty $r \leq n$. Dostali bychom tedy n čísel, z nichž poslední by bylo nulové. Jaký vztah mají tato čísla k singulárním číslům?

   Singulární čísla udávají nejmenší vzdálenost matice k jiné matici s nižší hodností. V našem případě se tedy singulární čísla rovnají optimální aproximační chybě.

3 Příloha

3.1 Úkol 1

```
load data_A

teziste = sum(A) / size(A, 1);
Atez = A - teziste;
[V, D] = eig(Atez' * Atez);
baze = V(:, 2);
projektor = baze * baze';
nove_body = projektor * Atez';
nove_body_tez = nove_body' + teziste;

hold on
scatter(nove_body_tez(:,1), nove_body_tez(:,2), [], 'red');
scatter(A(:,1),A(:,2), [], 'blue');
hold off
saveas(gcf, 'prolozeni-primkou.png');

soucet_vzdalenosti = D(1,1);
```

3.2 Úkol 2.1 - výpočet kritéria

```
A = load('walk1.txt','-ASCII');

m = size(A, 1); % number of rows
n = size(A, 2); % number of cols

center_of_gravity = sum(A) / m;

A0 = A - center_of_gravity; % translation to the origin

[~, D] = eig(A0' * A0); % spectral decomposition
eigenvalues = sum(D);

ranks = [1 2 5 10 15];
criterions = arrayfun(@(x) sum(eigenvalues(1:end - x)), ranks);
```

3.3 Úkol 2.1 - aproximace vstupní matice

```
function [A_approx, base] = approximation(A, rank)

m = size(A, 1); % number of rows
n = size(A, 2); % number of cols

center_of_gravity = sum(A) / m;

A0 = A - center_of_gravity; % translation to the origin

[V, ~] = eig(A0' * A0); % spectral decomposition
base = V(:,end - rank + 1:end);
projector = base * inv(base' * base) * base';
projection = projector * A0';
A_approx = projection' + center_of_gravity;

end
```

3.4 Úkol 2.2 - výpočet vektoru y

```
A = load('walk1.txt','-ASCII');

rank = 2;
[A_approx, base] = approximation(A, rank);
y = base \ A_approx';
plot(y(1,:), y(2,:));
title('Walk trajectory with rank 2');
saveas(gcf, 'walk-r2.png');

rank = 3;
[A_approx, base] = approximation(A, rank);
y = base \ A_approx';
plot3(y(1,:), y(2,:), y(3,:));
title('Walk trajectory with rank 3');
saveas(gcf, 'walk-r3.png');

A = load('makarena1.txt','-ASCII');

rank = 2;
[A_approx, base] = approximation(A, rank);
```

```
y = base \ A_approx';
plot(y(1,:), y(2,:));
title('Makarena trajectory with rank 2');
saveas(gcf, 'makarena-r2.png');

rank = 3;
[A_approx, base] = approximation(A, rank);
y = base \ A_approx';
plot3(y(1,:), y(2,:), y(3,:));
title('Makarena trajectory with rank 3');
saveas(gcf, 'makarena-r3.png');
```