

Cvičení z optimalizace: PCA

Autor: Vojtěch Franc, Tomáš Werner,
úprava Ondřej Drbohlav

Úloha 1: Aproximace bodů přímkou

Je dáno m bodů v rovině $\mathbf{a}_1, \dots, \mathbf{a}_m \in \mathbb{R}^2$. 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 přímkou.

Nápověda: Optimální přímka prochází těžištěm bodů.

Vstupní data: Stáhněte si z gitlabu soubor `data_A`, který obsahuje matici $\mathbf{A} \in \mathbb{R}^{m \times 2}$ s řádky $\mathbf{a}_1^T, \dots, \mathbf{a}_m^T$.

Úkoly k vypracování.

1. Zobraďte do jednoho obrázku zadané body $\mathbf{a}_1, \dots, \mathbf{a}_m$ (modře) a jejich kolmé projekce $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_m$ na nalezenou přímkou (červeně).

Výstup: graf.

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

Výstup: číslo.

3. Nalezněte požadovanou přímkou ve dvou různých reprezentacích:

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

Výstupem úkolu budou $\mathbf{x} \in \mathbb{R}^2$, $\alpha \in \mathbb{R}$, $\mathbf{y}_0 \in \mathbb{R}^2$, $\mathbf{s} \in \mathbb{R}^2$. Vektory zvolte tak, aby $\|\mathbf{x}\| = 1$, $\|\mathbf{s}\| = 1$, a aby norma $\|\mathbf{y}_0\|$ byla minimální (tj. bod \mathbf{y}_0 byl kolmý průmět počátku $\mathbf{0}$ na přímkou).

Výstup: vektory $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{y}_0 \in \mathbb{R}^2$, $\mathbf{s} \in \mathbb{R}^2$ a číslo $\alpha \in \mathbb{R}$.

Úloha 2: Kompresí sekvence z motion capture

Při tvorbě počítačových her nebo animovaných filmů se používá technologie *motion capture* (pokud víme, termín nemá zavedený český ekvivalent). Na živého herce se připevní terčíky odrážející infračervené světlo. Terčíky se připevňují na významné body na těle, jako klouby apod. Množství speciálních kamer pak snímá polohy terčíků a z těch se počítá poloha každého terčíku v třírozměrném prostoru pro každý snímek. Polohy terčíků v prostoru se pak použijí např. pro animaci postav syntetizovaných počítačovou grafikou. Viz např. Wikipedia.

Pro získání plynulého pohybu je třeba snímat s vysokou frekvencí. Například data použitá v naší úloze byla snímána s vzorkovací frekvencí 120 Hz. Ve výsledku je pak třeba pracovat s velkými objemy dat, což je nevýhoda. Naším úkolem bude snížit objem dat tak, abychom ztratili co nejméně užitečné informace.

Prostorová poloha jednoho terčíku v jednom snímku je dána trojicí souřadnic. V našem případě máme $\ell = 41$ terčíků. Poloha všech terčíků v jednom snímku je dána vektorem $\mathbf{a} \in \mathbb{R}^n$ délky $n = 3\ell$. Ten si lze představit jako bod v n -rozměrném prostoru. Celkově máme m snímků, tedy vektory $\mathbf{a}_1, \dots, \mathbf{a}_m \in \mathbb{R}^n$.

Hledáme body, které co nejlépe aproximují původní body a zároveň se dají reprezentovat menším objemem dat. Přesněji, hledáme body $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_m$, které leží v afinním podprostoru dimenze $r < n$ lineárního prostoru \mathbb{R}^n a zároveň minimalizují součet čtverců vzdáleností od původních bodů, tj. číslo

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

Z přednášek víte, že optimální afinní podprostor prochází těžištěm bodů

$$\mathbf{c} = \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i.$$

Když tedy body posunete tak, aby jejich těžiště leželo v počátku, tak místo afinního podprostoru můžete hledat podprostor lineární.

Jak dochází ke kompresi dat? Na první pohled se zdá, že původní i aproximované body zabírají stejné množství paměti ($m \times n$ čísel), takže jsme nic neušetřili. Ovšem nové body leží v podprostoru nižší dimenze r , který lze reprezentovat bází. Nechť tato báze je ortonormální a tvoří sloupce matice $\tilde{\mathbf{V}} \in \mathbb{R}^{n \times r}$. Pak každý bod v podprostoru je lineární kombinace bázevých vektorů,

$$\tilde{\mathbf{a}} = \tilde{\mathbf{V}} \mathbf{y}, \quad (2)$$

kde složky vektoru $\mathbf{y} \in \mathbb{R}^r$ jsou koeficienty lineární kombinace. Pro uložení všech bodů $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_m$ tedy musíme uložit bázevé vektory ($r \times n$ čísel, což je zanedbatelné) a pak vektory $\mathbf{y}_1, \dots, \mathbf{y}_m$ ($r \times m$ čísel).

Vstupní data a další soubory: Stáhněte si data `makarena1.txt` a `walk1.txt`¹. Tyto soubory popisují sekvence ‘Tanec Makarena’ a ‘Chůze’. Každý soubor obsahuje matici rozměru $m \times n$ (označte ji \mathbf{A}) s řádky $\mathbf{a}_1^T, \dots, \mathbf{a}_m^T$. Pro vizualizaci sekvencí použijte skript `show_motion.m` (vyžaduje stáhnout soubor `connected_points.txt`).

Úkoly k vypracování

1. Minimalizujte kritérium (1) za podmínky, že body $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_m$ leží v afinním podprostoru dimenze r . Výsledkem bude matice $\tilde{\mathbf{A}}$ s řádky $\tilde{\mathbf{a}}_1^T, \dots, \tilde{\mathbf{a}}_m^T$. Proveďte pro sekvenci ‘Chůze’ a pro pět různých hodnot $r \in \{1, 2, 5, 10, 15\}$. Pro kontrolu si jednotlivá řešení přehrajte skriptem `show_motion2.m`, který zobrazí sekvenci v matici \mathbf{A} spolu se sekvencí v matici $\tilde{\mathbf{A}}$. Pokud je vaše řešení správné, budou obě sekvence podobné.

¹ Sekvence byly staženy ze stránek <http://movlab.ulusofoa.pt>.

Výstup: tabulka s optimálními hodnotami kritéria (1) pro zadané hodnoty $r \in \{1, 2, 5, 10, 15\}$.

2. Výsledné body vyjádřete jako lineární kombinaci (2) bázevých vektorů. Pro $r = 2$ nakreslete sekvenci vektorů $\mathbf{y}_1, \dots, \mathbf{y}_m$ jako trajektorii v rovině (po sobě jdoucí body spojte úsečkou). To samé udělejte i ve třírozměrném prostoru, tedy pro $r = 3$. Proveďte pro sekvence ‘Tanec Makarena’ a ‘Chůze’.

Výstup: dva grafy zobrazující 2D trajektorii a dva grafy zobrazující 3D trajektorii.

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

Výstup: číslo a stručné teoretické odůvodnění.

4. Dejme tomu, že bychom chtěli spočítat optimální chybu aproximace (1) 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? Jako inspiraci si vykreslete singulární čísla seřazená do grafu.

Výstup: vzorec.