

Redukce dimenzionality (PCA)

Redukce dimenzionality je proces snižování počtu vstupních proměnných (rysů/dimenzí) v datové sadě. Cílem je zjednodušit data, odstranit šum a umožnit vizualizaci při zachování co největšího množství původních informací.

Nejznámější a nejpoužívanější lineární metodou pro tento úkol je **Analýza hlavních komponent** (Principal Component Analysis – **PCA**).

1. Proč redukovat dimenze?

- **Prokletí dimenzionality (Curse of Dimensionality):** S rostoucím počtem dimenzí se data stávají „řídkými“, což zhoršuje výkon mnoha algoritmů (např. [K-Means](#) nebo k-NN).
- **Vizualizace:** Lidé dokáží vnímat data maximálně ve 3D. PCA umožní promítnout např. 100-rozměrná data do 2D grafu.
- **Odstranění multikolinearity:** PCA vytvoří nové rysy, které mezi sebou nejsou korelované.
- **Zrychlení výpočtů:** Méně dat znamená rychlejší trénování modelů a nižší nároky na paměť.

2. Jak PCA funguje? (Princip)

PCA matematicky transformuje data do nového souřadnicového systému:

1. **Hledání směru největšího rozptylu:** Algoritmus najde směr (osu), ve kterém se data nejvíce mění. Tato osa se nazývá **První hlavní komponenta (PC1)**.
2. **Ortogonalita:** Druhá hlavní komponenta (**PC2**) musí být kolmá na první a musí vysvětlovat druhý největší podíl zbývajících rozptylu.
3. **Projekce:** Původní data jsou promítnuta na tyto nové osy.

3. Postup výpočtu (Algoritmus)

Pro zájemce o matematické pozadí probíhá PCA v těchto krocích:

- **Standardizace:** Data musí mít průměr 0 a rozptyl 1 (jinak by proměnné s velkými čísly dominovaly).
- **Výpočet kovarianční matice:** Určení vztahů mezi všemi dvojicemi rysů.
- **Výpočet vlastních čísel (eigenvalues) a vlastních vektorů (eigenvectors):** Ty určují směr a sílu hlavních komponent.
- **Seřazení:** Komponenty se seřadí podle velikosti vlastních čísel.

4. Praktické využití

- **Kompresie obrazu:** Snížení počtu pixelů při zachování rozpoznatelnosti objektů.

- **Genetika:** Analýza tisíců genů a jejich seskupování do skupin podle podobnosti.
- **Finance:** Identifikace hlavních faktorů ovlivňujících trh (např. úrokové sazby, inflace).
- **Preprocessing pro ML:** Často se používá jako krok před nasazením [regrese](#) nebo klasifikace.

5. Omezení PCA

- **Ztráta interpretovatelnosti:** Hlavní komponenty jsou lineární kombinací původních rysů. Např. PC1 už není „věk“, ale složitý vzorec složený z věku, příjmu a vzdělání.
- **Lineární omezení:** PCA předpokládá lineární vztahy. Pro nelineární data jsou vhodnější metody jako **t-SNE** nebo [Autoenkodéry](#).

Ukázka v Pythonu (Scikit-Learn)

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# 1. Standardizace dat
X_std = StandardScaler().fit_transform(X)

# 2. Inicializace PCA (chceme zachovat 2 nejdůležitější komponenty)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_std)

# 3. Zjištění, kolik informací jsme zachovali
print(f"Zachovaný rozptyl: {pca.explained_variance_ratio_.sum()}")
```

Související články:

- [Standardizace a normalizace dat](#)
- [Autoenkodéry \(Nelineární redukce\)](#)
- [Shlukování a jeho problémy v mnoha dimenzích](#)

Tagy: *ai ml pca statistika dimensionality_reduction math*

From:

<https://www.serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:

<https://www.serviceit.cz/doku.php?id=it:ml:pca>

Last update: **2026/01/02 12:46**

