

# Porovnání klasického SVD a randomizovaného SVD

## 1. Úvod

Singulární rozklad matice (Singular Value Decomposition, SVD) patří mezi základní nástroje numerické lineární algebry a nachází široké využití v oblastech jako je zpracování dat, redukce dimenze, komprese informací nebo řešení úloh nejmenších čtverců.

Klasické algoritmy pro výpočet SVD jsou numericky stabilní a přesné, avšak jejich výpočetní a paměťová náročnost může být problematická při práci s velkými maticemi.

V posledních letech se proto objevují **randomizované algoritmy**, které umožňují aproximovat SVD s výrazně nižší výpočetní složitostí a paměťovými nároky, a to za cenu malé ztráty přesnosti. Tyto metody jsou zvláště vhodné pro velké a řídké matice nebo situace, kdy postačuje aproximace nízké hodnoty.

Cílem tohoto projektu je **porovnat klasický SVD a randomizovaný SVD** z hlediska:

- výpočetního času,
- paměťové náročnosti,
- přesnosti aproximace.

## 2. Popis použitých metod

### 2.1 Klasický SVD

Klasický singulární rozklad matice  $A \in \mathbb{R}^{m \times n}$  má tvar

$$A = U \Sigma V^T,$$

kde:

- $U \in \mathbb{R}^{m \times r}$  a  $V \in \mathbb{R}^{n \times r}$  jsou ortogonální matice,
- $\Sigma \in \mathbb{R}^{r \times r}$  je diagonální matice singulárních hodnot,
- $r = \min(m, n)$ .

V projektu je klasický SVD realizován pomocí knihovny **SciPy**, která využívá efektivní LAPACK implementace. Tato metoda poskytuje velmi přesné výsledky, avšak její výpočetní složitost je řádově  $\mathcal{O}(mn \min(m, n))$ , což může být náročné pro velké matice.

### 2.2 Randomizovaný SVD

Randomizovaný SVD je aproximační metoda, která nejprve promítne původní matici do prostoru menší dimenze pomocí náhodné projekce. Následně se SVD provádí pouze na této menší matici.

Základní kroky algoritmu jsou:

1. náhodná projekce pomocí matice s náhodnými prvky,
2. volitelné **power iterations** pro zvýšení přesnosti,
3. ortogonalizace (QR rozklad),
4. výpočet SVD menší matice,

5. zpětná projekce do původního prostoru.

Tento přístup výrazně snižuje výpočetní náročnost a paměťové nároky, zejména pokud má matice nízkou efektivní hodnotu.

### 3. Testovací data

Testování bylo provedeno na třech různých typech matic:

1. **Náhodná hustá matice**
  - matice s náhodně generovanými prvky.
2. **Matice s nízkou hodnotou (Low-rank)**
  - matice vytvořená jako součin dvou menších matic, s cílovou hodnotou  $k = 20$ .
3. **Strukturovaná bloková matice**
  - bloková matice složená z menších náhodných podmatic, simulující strukturovaná data.

Tento výběr umožňuje porovnat chování obou metod na různých typech vstupních dat.

### 4. Kritéria hodnocení

Porovnání metod bylo provedeno podle následujících kritérií:

- **Výpočetní čas** – měřen pomocí funkce `time.perf_counter`.
- **Paměťová náročnost** – měřena pomocí knihovny `tracemalloc`.
- **Přesnost aproximace** – vyjádřená relativní chybou ve Frobeniově normě:

$$\frac{\|A - \tilde{A}\|_F}{\|A\|_F}.$$

### 5. Výsledky

Naměřené výsledky ukazují, že:

- **Randomizovaný SVD** je ve všech testovaných případech výrazně rychlejší než klasický SVD.
- Paměťová náročnost randomizovaného SVD je podstatně nižší, zejména u větších a strukturovaných matic.
- Přesnost aproximace je u low-rank matic velmi dobrá a rozdíl oproti klasickému SVD je zanedbatelný.
- U náhodných hustých matic je chyba mírně vyšší, avšak stále přijatelná vzhledem k výrazné úspoře výpočetního času.

Grafické porovnání času výpočtu a paměťové náročnosti je uloženo ve složce `results`.

### 6. Závěr

V této práci bylo provedeno srovnání klasického algoritmu SVD a jeho randomizované varianty z hlediska výpočetního času, paměťové náročnosti a přesnosti aproximace. Testování bylo provedeno na třech různých typech matic: náhodné husté matici, matici s nízkou hodnotou a strukturované (blokové) matici.

Z naměřených výsledků vyplývá, že randomizované SVD dosahuje výrazně lepšího výpočetního času zejména v případě matic s nízkou hodnotí. Tento výsledek odpovídá teoretickým předpokladům, jelikož randomizovaný přístup efektivně využívá nízkou vnitřní dimenzi dat a omezuje výpočet pouze na dominantní singulární hodnoty. U náhodných matic je zrychlení méně výrazné, avšak stále pozorovatelné.

Naopak u strukturovaných matic se výpočetní čas klasického a randomizovaného SVD výrazně neliší. V tomto případě zjevně struktura matice neposkytuje randomizovanému algoritmu výraznou výhodu, což naznačuje, že přínos randomizovaného SVD závisí na vlastnostech vstupních dat.

Z hlediska přesnosti aproximace byla u všech testovaných matic dosažena srovnatelná relativní chyba ve Frobeniově normě. Randomizované SVD tedy představuje vhodný kompromis mezi rychlostí výpočtu a přesností, zejména pro velké a nízko-hodnotní matice, kde klasické SVD může být výpočetně náročné.

Celkově lze konstatovat, že randomizované SVD je efektivní alternativou ke klasickému SVD v situacích, kdy je kladen důraz na rychlost a paměťovou úspornost výpočtu, a mírná aproximace výsledku je akceptovatelná.