SWD – Laboratorium 6

Rozkład SVD

Iwo Błądek

18 maja 2020

1 Zadanie domowe (5 punktów) – opcjonalne

Zadanie 1.1: Stwórz program konsolowy w Pythonie, który będzie dokonywał kompresji obrazków (kolorowych) przy użyciu SVD. Wymagania:

- Należy zaimplementować zarówno własną wersję SVD, jak i użyć tej z numpy lub scikit-learn (wybór implementacji będzie określany przez użytkownika odpowiednim argumentem).
- Obsługa argumentów linii poleceń musi być obsługiwana przez bibliotekę `argparse` (lub inną podobnej klasy). Wymagane argumenty:
  - `-f` – plik z oryginalnym obrazkiem.
  - `-out` – nazwa pliku wyjściowego, do którego zapisany ma być skompresowany obrazek. Jeżeli nie zostanie podany, to skompresowany obrazek należy po prostu wyświetlić (domyślne zachowanie).
  - `-svd` – implementacja SVD do użycia. Możliwe wartości: `'custom'` (domyślna), `'scikit'`.
  - `-k` – liczba wartości osobliwych użyta do kompresji (domyślnie wszystkie, czyli brak kompresji).

  Niezgodność z powyższą specyfikacją będzie skutkować odejmowaniem punktów. W celu przetestowania poprawności swojego programu można skorzystać z testera udostępnionego na `https://github.com/iwob/swd_labs/tree/master/lab6`.
- Do znalezienia wektorów i wartości własnych zamiast *linalg.eig* wykorzystać można *linalg.eigh* zoptymalizowane dla macierzy symetrycznych. Wszystkie wartości własne powinny wyjść nieujemne.
- W SVD jednym z problemów jest to, by wektory własne z U i V miały odpowiednie znaki, czego niestety EVD zastosowane osobno do R_{mm} i C_{nn} nie gwarantuje. O rozwiązaniu tego problemu można przeczytać poczynając od slajdu 52 wykładu (`http://www.cs.put.poznan.pl/rsusmaga/Dydaktyka/SkaiWiD--2017/SkaiWiD-wyklad%2310'-sent.pdf`). Algorytm przedstawiony jest na slajdach 55 i 56. Do policzenia T wystarczy znajomość wartości własnych, i po obliczeniu jednej z pozostałych macierzy (U lub V), można znaleźć drugą. To, który wzór trzeba zastosować, wynikać będzie z wymiarów T (Σ), tak by mnożenie T i jej pseudo-odwrotności było wykonywalne.

W celu przetestowania swojego rozwiązania można porównać skompresowane obrazki z tymi wygenerowanymi na stronie: `http://timbaumann.info/svd-image-compression-demo/` (jest tam opcja uploadu własnego obrazka).

Uwaga: numpy do znajdowania wartości i wektorów własnych potrzebuje macierzy typu float, jednak jak takiej nie dostanie to nie rzuci wyjątkiem tylko zwróci niepoprawne wartości własne (zwłaszcza

ujemne, które nie mają prawa wyjść dla macierzy typu AA^T). Niektóre biblioteki przetwarzania obrazów reprezentują wartości pikseli jako inty – należy w takim wypadku przekonwertować macierz obrazka na typ float.