**Звіт:**

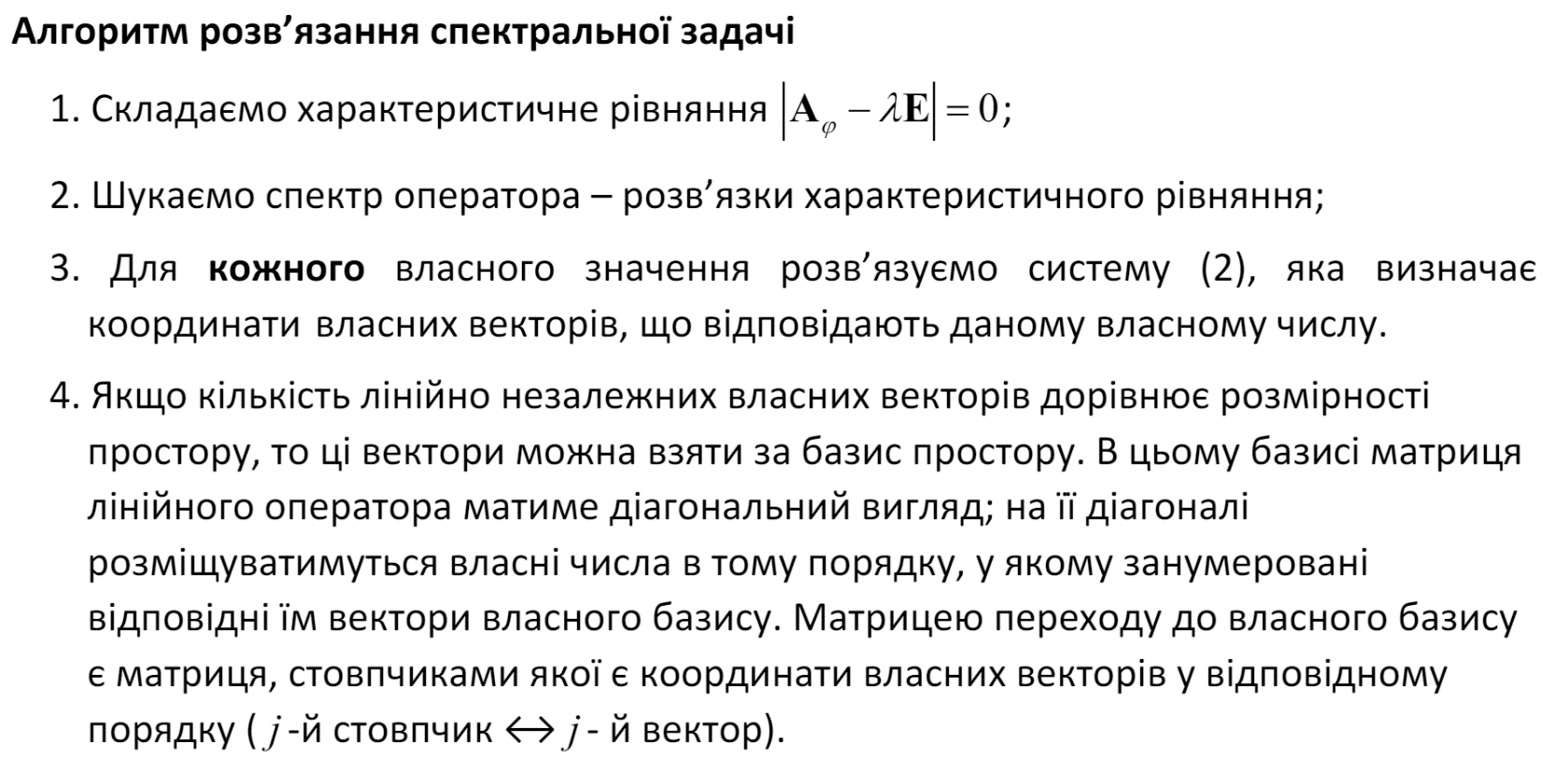
Завдання:

<https://www.geeksforgeeks.org/numpy-linalg-eig-method-in-python/>

<https://www.kaggle.com/code/mirzarahim/introduction-to-pca-image-compression-example>

1. **Зображення, що містить текст, знімок екрана, Шрифт, документ

   Автоматично згенерований описЩо таке власне значення і власний вектор матриці? Як вони обчислюються**?

Зображення, що містить Шрифт, білий, Графіка, типографія

Автоматично згенерований опис

Зображення, що містить текст, почерк, папір, документ

Автоматично згенерований опис

1. **Які властивості мають власні вектори симетричних матриць?**

1. Ортогональність: Власні вектори, що відповідають різним власним числам симетричної матриці, є ортогональними один до одного. Це означає, що їх скалярний добуток дорівнює нулю.

2. Нормалізація: Власні вектори можна нормалізувати таким чином, що їхні норми дорівнюють одиниці.

3. Повнота: Для симетричної матриці всі власні вектори (відповідно до різних власних чисел) утворюють повну ортогональну систему векторів у просторі, що має таку саму розмірність, як і матриця.

4.Базис: Власні вектори симетричної матриці можуть бути використані як базис у відповідному векторному просторі. Це дає можливість зручно виразити матрицю у вигляді діагональної матриці за допомогою перетворення базису, де власні числа становлять елементи головної діагоналі.

<https://builtin.com/data-science/symmetric-matrix>

1. **Які можуть бути недоліки використання PCA, і які стратегії можуть використовуватися для подолання цих недоліків?**

Аналіз основних компонентів (PCA) — це техніка, яка використовується в обробці зображень для зменшення його розмірності. У PCA зображення представляється як матриця піксельних значень, і алгоритм ідентифікує головні компоненти зображення шляхом знаходження власних векторів коваріаційної матриці піксельних значень.

Коваріаційна матриця піксельних значень є інструментом у статистиці і обробці зображень, що використовується для вимірювання статистичних залежностей між пікселями у зображенні.

Простіше кажучи, вона вказує, наскільки і як пікселі в зображенні змінюються разом. Це значення використовується для оцінки кореляції між яскравістю або кольором сусідніх пікселів: чим вище коваріація, тим сильніше залежність.

Ці власні вектори представляють найважливіші характеристики або шаблони в зображенні.

Зменшуючи розмірність даних зображення за допомогою PCA, стає легше стискати та зберігати зображення, а також виконувати такі операції, як розпізнавання та класифікація зображень.

Недоліки PCA і способи їх подолання:

1. Втрата змісту ознак: PCA може змішувати ознаки, тому нові "компоненти" можуть бути важко зрозуміти.

Як подолати: Використовуйте інші методи, які зберігають зміст ознак, наприклад, Factor Analysis або Kernel PCA.

2. Чутливість до виняткових значень: Якщо є великі виняткові значення, PCA може дати неправильні результати.

Як подолати: Використовуйте більш стійкі до виняткових значень методи, як Robust PCA.

3. Не підходить для складних даних: Якщо залежності між ознаками не лінійні, PCA може бути неефективним.

Як подолати: Використовуйте методи, які краще розуміють складні залежності, наприклад, Kernel PCA або нейронні мережі.

4. Вибір кількості компонент: Вибрати правильну кількість компонентів для збереження інформації може бути складно.

Як подолати: Використовуйте перехресну перевірку або правила, які дозволяють зберегти достатньо інформації.

5. Обмеження кореляційних залежностей: PCA працює тільки з лінійними залежностями між ознаками.

Як подолати: Використовуйте інші методи, які розглядають нелінійні або незалежні залежності, наприклад, Independent Component Analysis (ICA).

Як PCA знаходить головні компоненти?

PCA (Principal Component Analysis, аналіз головних компонент) використовується для зменшення розмірності даних, зберігаючи якомога більше інформації. Ось як це працює крок за кроком:

1. Центрування даних: Першим кроком є віднімання середнього значення від кожної ознаки (наприклад, від кожного пікселя). Це робиться для того, щоб дані були "зосереджені" навколо нуля. Тобто, якщо у вас є набір зображень, кожен піксель кожного зображення центрується, щоб його середнє значення по всьому набору було нульовим.

2. Коваріаційна матриця: Після цього створюється матриця коваріацій. Ця матриця показує, як зміни в одному пікселі пов'язані зі змінами в інших пікселях.

3. Власні вектори і власні значення: Далі, з коваріаційної матриці обчислюються власні вектори і власні значення. Власні вектори - це напрямки в просторі пікселів, які показують основні напрямки варіації даних (тобто, в якій стороні дані найбільше "розкидані"). Власні значення показують, наскільки важливий кожен із цих напрямків.

4. Головні компоненти:

- Перша головна компонента: Це напрямок, в якому дані мають найбільшу розбіжність (варіацію). Тобто, якщо уявити дані як хмару точок, то перша головна компонента - це лінія, вздовж якої ця хмара є найдовшою.

- Друга головна компонента: Це напрямок, який має найбільшу розбіжність після першої компоненти і є перпендикулярним до неї. Тобто, друга головна компонента знаходиться під прямим кутом до першої і показує напрямок другої за важливістю варіації.

- Третя і наступні головні компоненти: Це інші напрямки, які мають найбільшу розбіжність, але при цьому є перпендикулярними до всіх попередніх компонент.

5. Проекція даних: Після знаходження головних компонент кожне зображення можна "спроектувати" на новий простір з меншою кількістю вимірів. Тобто замість тисяч пікселів ми описуємо кожне зображення кількома головними компонентами, що зберігають основну інформацію.

Приклад:

Уявімо, що у вас є набір зображень розміром 100x100 пікселів, кожне з яких можна розглядати як точку в 10 000-вимірному просторі.

1. Центрування: Від кожного значення пікселя віднімаємо середнє значення цього пікселя по всім зображенням.

2. Створюємо коваріаційну матрицю розміром 10 000x10 000.

3. Обчислюємо власні вектори і власні значення цієї матриці.

4. Вибираємо, наприклад, 50 головних компонент, що мають найбільші власні значення.

5. Проектуємо кожне зображення на новий простір з 50 вимірів, замість початкових 10 000.

1. **Які переваги має діагоналізація матриці в криптографії? Як вона застосовується для шифрування та дешифрування повідомлень?**

1. Підвищення безпеки шифрування: Діагоналізація матриці може використовуватися для створення ключів шифрування. Це означає, що кожен ключ або шифрувальний параметр зберігається як частина діагональної матриці, що ускладнює злам шифру.

2. Ефективність алгоритмів шифрування: Діагоналізація може спрощувати алгоритми шифрування, оскільки вона може зводитися до простих операцій з матрицями, такими як множення і обернення.

3. Збереження конфіденційності: Застосування діагоналізації дозволяє зберігати важливі параметри (наприклад, ключі) у вигляді діагональних елементів матриці, що підвищує захищеність від несанкціонованого доступу.

Як це застосовується для шифрування та дешифрування повідомлень?

- Шифрування: Перед шифруванням повідомлення, його можна перетворити у вектор (або матрицю) та діагоналізувати цю матрицю, застосовуючи ключ шифрування як параметри діагоналізації. Це перетворення може ускладнити злам шифру, оскільки ключеві параметри знаходяться у діагональних елементах матриці.

- Дешифрування: Для дешифрування зашифрованого повідомлення застосовується обернена діагоналізація, де використовується обернена матриця до тієї, яка була використана при шифруванні. Цей процес відновлює початковий вектор (або матрицю) з зашифрованого тексту.