PCA visualization and PCA-kernel

Paweł Banach, Michał Kawałek

PCA - czym jest? do czego służy?

Principal Component Analysis - Analiza Składowych Głównych

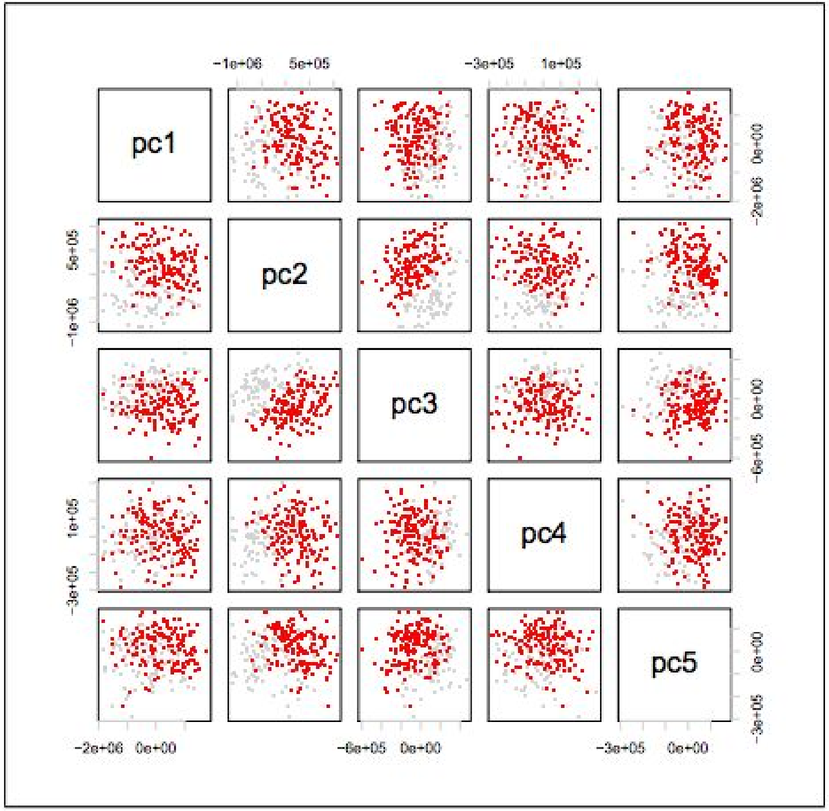
Zastosowania:

- Wizualizacja wielowymiarowych danych w np. 2-, 3-wymiarowej przestrzeni.
- Redukcja wymiarów danych wielowymiarowych
- Takie rzutowanie danych aby osiągnąć jak największą wariancję głównych składowych
- oparta na macierzy kowariancji

Przykładowe zastosowanie - Rak jajnika

- wybór 5 głównych składowych
- rzut na płaszczyznę wyznaczoną przez ich pary
- dzięki parze składowych nr 2 i 3 można przeprowadzić klasyfikację.

https://www.mimuw.edu.pl/~aniag/SADM/pca.pdf



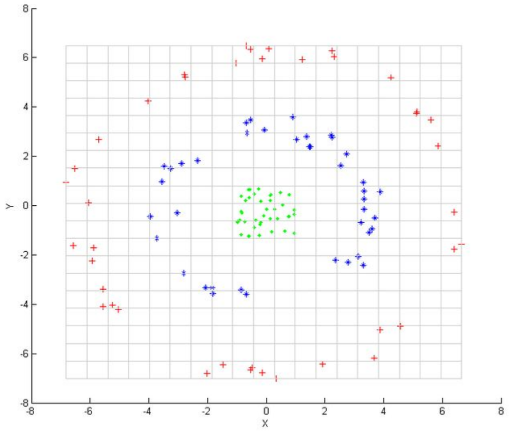
Rysunek 1: Analiza składowych głównych dla zbioru danych z Clinical Proteomics Program Databank dotyczących raka jajnika. Rzutowanie obserwacji na układy współrzędnych wyznaczone przez pary pierwszych pięciu głównych składowych. Czerwone punkty odpowiadają chorym pacjentom, szare zdrowym. W tym przypadku rzutowanie na drugą i trzecią główną składową dobrze rozdziela obserwacje z różnych klas.

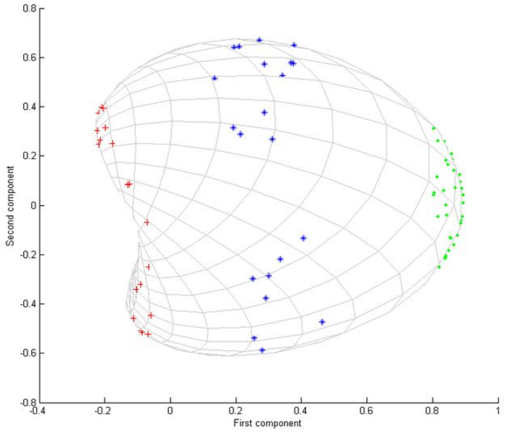
6 kroków PCA

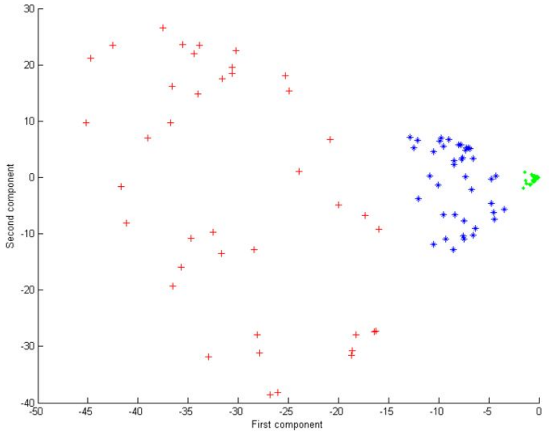
1. Ustandaryzowanie zbioru.
2. Obliczenie wektora wartości średnich.
3. Wygenerowanie macierzy kowariancji.
4. Obliczenie wektorów własnych wraz z wartościami własnymi.
5. Sortowanie wektorów po wartości własnej i wybranie k z nich do stworzenia naszej k-wymiarowej podprzestrzeni.
6. Stworzenie macierzy wektorów własnych i transformacja z jej pomocą danych do nowego zbioru.

Kernel-PCA

- W przypadku gdy dane nie są liniowo podzielne - PCA nam nie pomoże
- Kernel-PCA pozwala na rzutowanie danych na wyższe wymiary przy wykorzystaniu kernel methods
- Na przykład: dane początkowo dwuwymiarowe lecz niemożliwe do prostej klasyfikacji przekształcamy na trójwymiarowe, by następnie łatwo było nam je odpowiednio rzucić na płaszczyznę i zaklasyfikować







Kernel Methods

- kernel methods - pozwalają na zmianę wymiaru danego wektora w przestrzeni poprzez odpowiedni mapping (macierz kowariancji), który zachowuje pewne wewnętrzne (w obrębie przestrzeni) relacje
- Najpopularniejsze kernel methods:
  - Linear

$$k(x, y) = x^T y + c$$

  - Polynomial

$$k(x, y) = (\alpha x^T y + c)^d$$

  - Gaussian

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

DEMO

Jak to wygląda w praktyce ?

Użyta biblioteka python - sklearn

Korzystamy z zawartego w bibliotece modułu ‘decomposition’

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

A w szczególności z procedur:

sklearn.decomposition.**PCA**

sklearn.decomposition.**KernelPCA**

Więcej informacji można znaleźć w dokumentacji scikit-learn.

Użyte materiały

PCA:

http://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_vs_lda.html#sphx-glr-auto-examples-decomposition-plot-pca-vs-lda-py

https://plot.ly/ipython-notebooks/principal-component-analysis/

http://sebastianraschka.com/Articles/2014_pca_step_by_step.html

kernelPCA:
http://rasbt.github.io/mlxtend/user_guide/feature_extraction/RBFKernelPCA/

http://scikit-learn.org/stable/auto_examples/decomposition/plot_kernel_pca.html

Skrypt z tutoriala do ściągnięcia z:

https://github.com/PawelBanach/MRO-PCA