Project 3

clustering, dimensionality reduction, and non-monotonous neurons

Lukas Drexler, Leif Van Holland, Reza Jahangiri,
Mark Springer, Maximilian Thiessen

January 24, 2018

Rheinische Friedrich-Wilhelms-Universität

Task 1.1

Fun with k-means clustering

Task 3.2

Spectral clustering

Given data, $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^2$, compute similarity matrix:

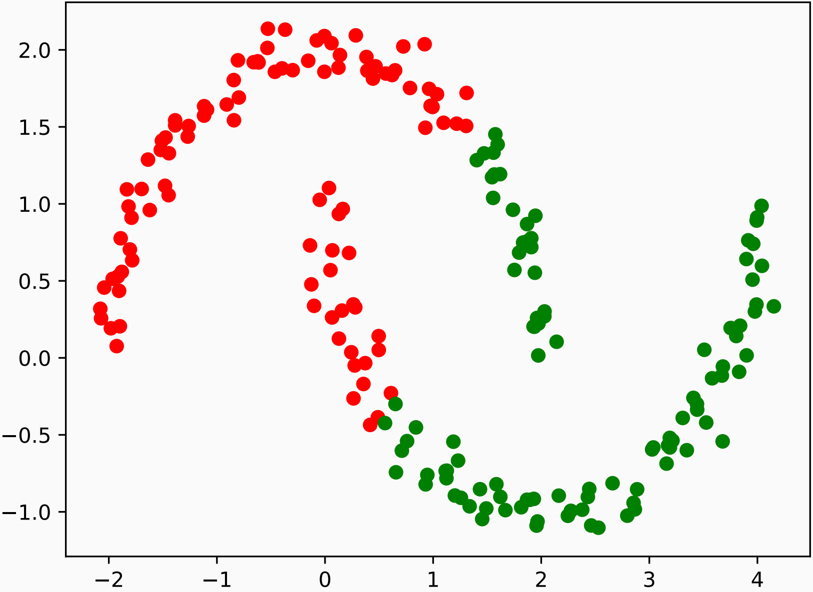
$$S_{ij} = e^{-\beta||x_i - x_j||^2}$$

and the Laplacian $L = D - S$, with diagonal matrix D:

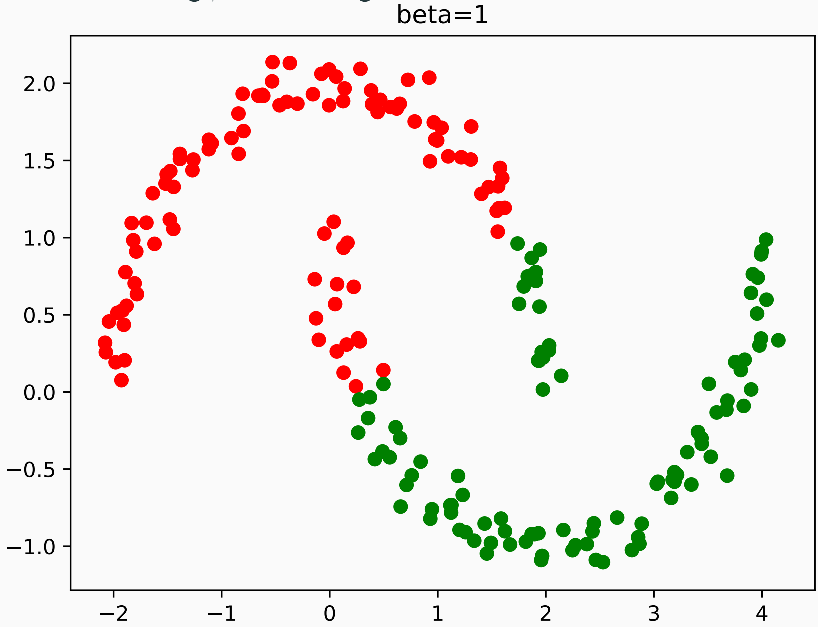
$$D_{ij} = \begin{cases} \sum_j S_{ij} & \text{, if } i = j \\ 0 & \text{, otherwise} \end{cases}$$

Then compute the eigenvalue decomposition of L. Use the eigenvector of the second smallest eigenvalue to partition the data points, according to the sign of the corresponding entry in the eigenvector.

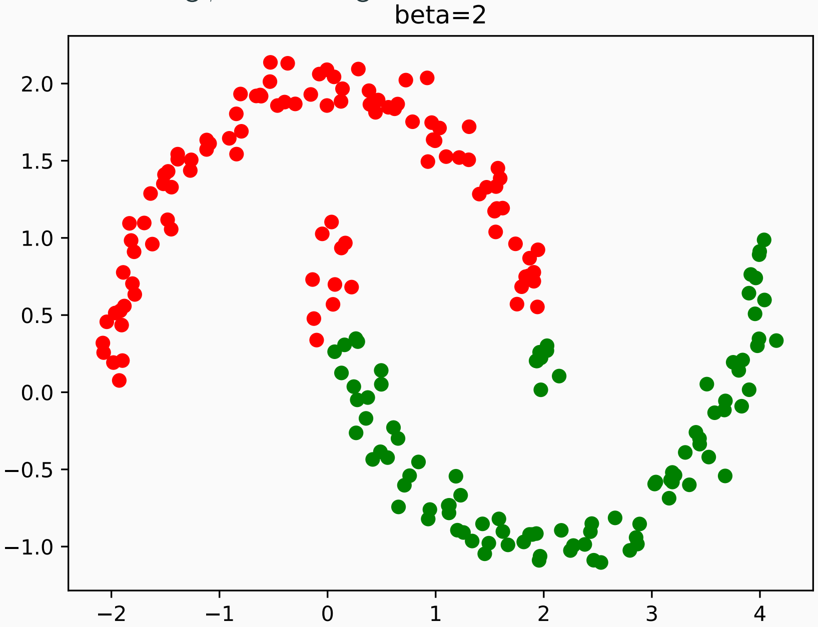
The result using kmeans++:



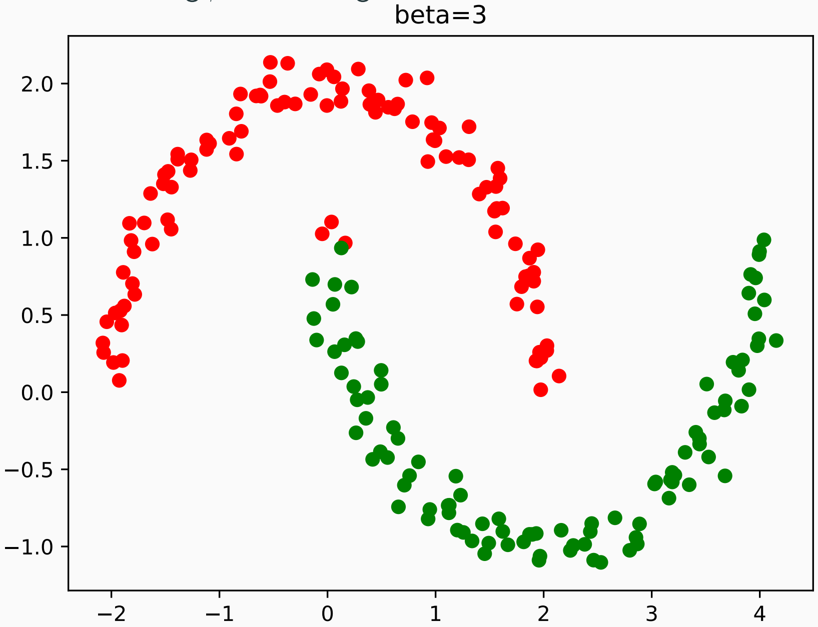
The result using $\beta = 1$ through 4:



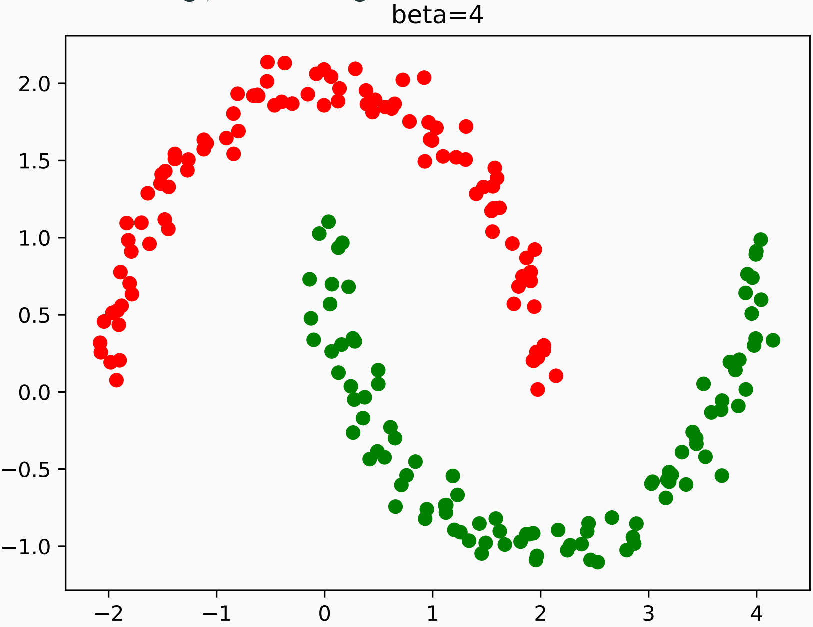
The result using $\beta = 1$ through 4:



The result using $\beta = 1$ through 4:



The result using $\beta = 1$ through 4:



Task 3.3

Dimensionality reduction

This task explores mappings $\mathbb{R}^{500} \to \mathbb{R}^2$, in particular Principal Component Analysis (PCA) and multi-class LDA.

Dimensionality reduction

Idea: Project $n = 150$ *zero-mean* data $x_1, ..., x_n$ onto eigenvectors u_1 and u_2 corresponding to largest eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$:

PCA: eigenvectors of sample covariance matrix

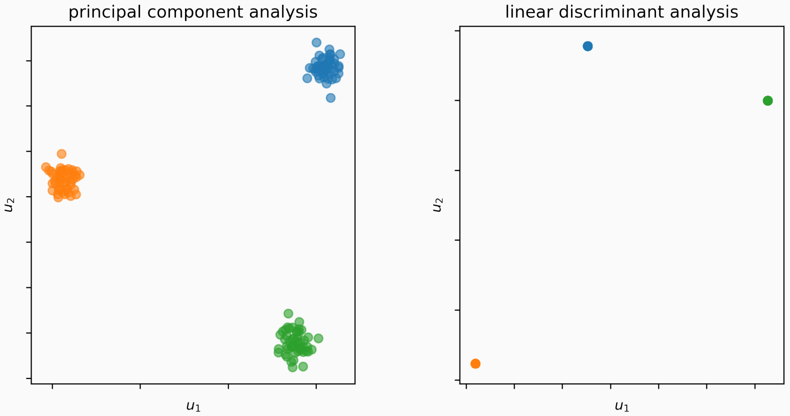
$$\Sigma = \frac{1}{n+1} \sum_{i=1}^{n} x_i \cdot x_i^T$$

LDA: eigenvectors of

$$S_W^{-1} S_B = \left(\sum_{j=1}^{k} \Sigma_j \right)^{-1} \sum_{j=1}^{k} \mu_j \cdot \mu_j^T$$

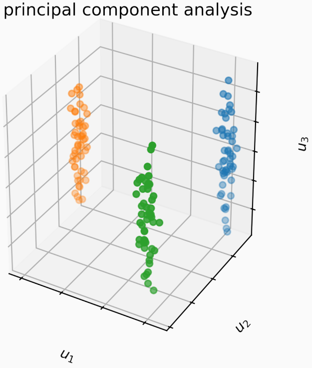
where μ_j are the mean vectors and Σ_j are the covariance matrices for the $k = 3$ classes the data points are from

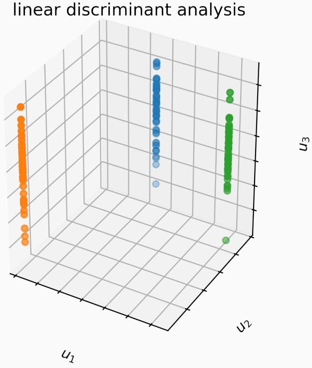
PCA and LDA in two dimensions



Observe: Data points with same label form clusters if projected.

PCA and LDA in three dimensions





Observe: Data points are (per class) located in a 1D-subspace of $\mathbb{R}^{500}$.

PCA vs LDA

Why do the results differ?

PCA: Yields axes of maximum variance in complete data set.
$\Rightarrow$ Preserves as much *variance* as possible.

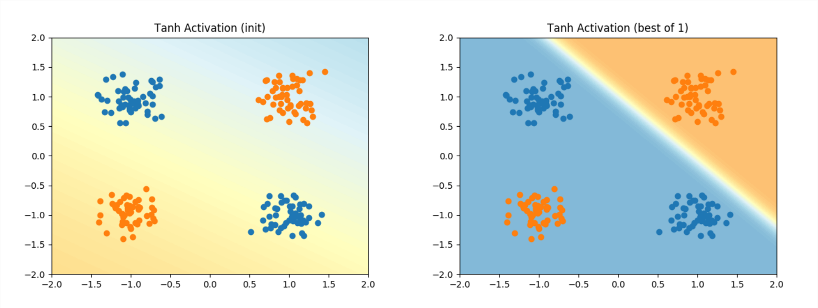
LDA: Uses class labels to maximize distance between class means and minimize class variances.
$\Rightarrow$ Preserves as much *discriminatory* information as possible.

Task 3.4

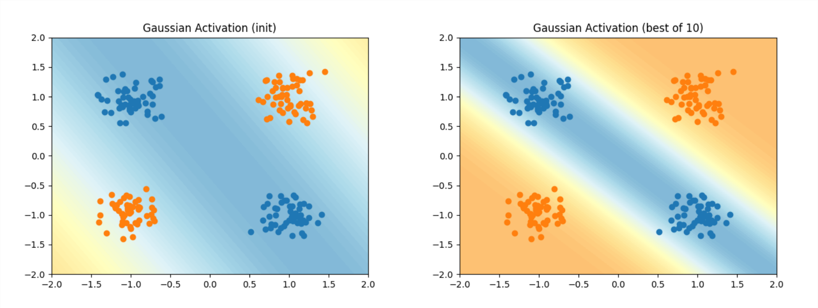
Perceptrons

- Gradient descent for perceptron learning:
    - line search version
    - use best result of k runs, to prevent local minima
    - monotonous activation separates only linearly
    - avoid this limitation with non-monotonous activation

Monotonous Perceptron (tanh)



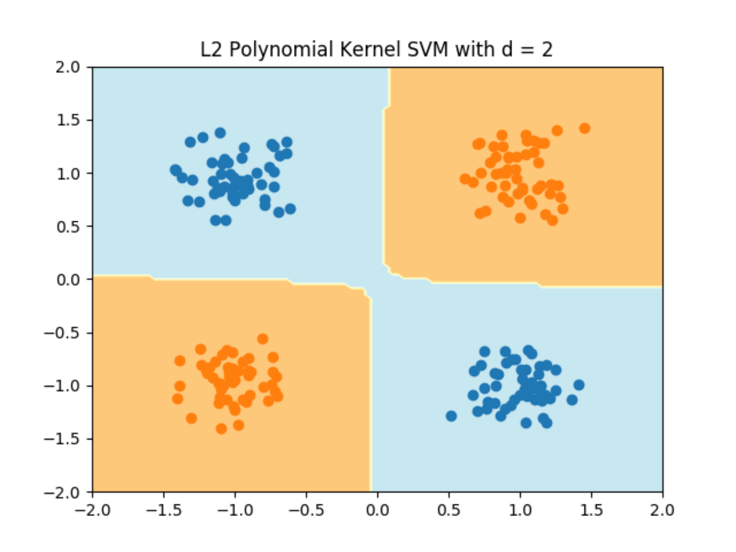
Non-monotonous Perceptron (Gaussian)



SVM

- L2-SVM with polynomial kernel:
    - Train with Frank-Wolfe algorithm
    - $d = 2$ (i.e. quadratic kernel) is enough

Polynomial Kervel SVM



Task 3.5

Exploring numerical instabilities