Cognitive Algorithms - Exercise 2

Daniel Wujecki

May 9, 2020

1. Task 1 - Linear Classification with NCC and Perceptron
2. Task 2 - Covariance Matrices
3. Task 3 - Linear Discriminant Analysis
4. Task 4 - Understanding LDA: the whitening operation
5. Model evaluation

Classification

- Common problem in machine learning
- find a mapping $f : \mathbb{R}^d \rightarrow \mathcal{C}$ that estimates the class $y \in \mathcal{C}$ for a set of features $\mathbf{x} \in \mathbb{R}^d$

Classification

- Common problem in machine learning
- find a mapping $f : \mathbb{R}^d \rightarrow \mathcal{C}$ that estimates the class $y \in \mathcal{C}$ for a set of features $\mathbf{x} \in \mathbb{R}^d$
- binary classification with $\mathcal{C} = \{-1, 1\}$
- simple mapping $f(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} - \beta)$

Classification

- Common problem in machine learning
- find a mapping $f : \mathbb{R}^d \rightarrow \mathcal{C}$ that estimates the class $y \in \mathcal{C}$ for a set of features $\mathbf{x} \in \mathbb{R}^d$
- binary classification with $\mathcal{C} = \{-1, 1\}$
- simple mapping $f(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} - \beta)$
- $\mathbf{w} \in \mathbb{R}^d$ is called the weight vector
- β ist the bias or offset

Supervised learning

- How do we find a mapping f that estimates the *correct* labels $y \in \mathcal{C}$?

Supervised learning

- How do we find a mapping f that estimates the *correct* labels $y \in \mathcal{C}$?
- usally we need a set of observations with known class labels
- training data set can be given as a matrix $X \in \mathbb{R}^{d \times n}$ and a vector $\mathbf{y} \in \mathcal{C}^n$

$$X = \begin{bmatrix} 0.1 & -3.2 & \cdots & 2.1 & 1.4 \\ & \vdots & & \vdots & \\ 0.1 & -3.2 & \cdots & 2.1 & 1.4 \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} -1 & 1 & \cdots & 1 & -1 \end{bmatrix}$$

Supervised learning

- How do we find a mapping f that estimates the *correct* labels $y \in \mathcal{C}$?
- usally we need a set of observations with known class labels
- training data set can be given as a matrix $X \in \mathbb{R}^{d \times n}$ and a vector $\mathbf{y} \in \mathcal{C}^n$

$$X = \begin{bmatrix} 0.1 & -3.2 & \cdots & 2.1 & 1.4 \\ & \vdots & & \vdots & \\ 0.1 & -3.2 & \cdots & 2.1 & 1.4 \end{bmatrix}$$
$$\mathbf{y} = \begin{bmatrix} -1 & 1 & \cdots & 1 & -1 \end{bmatrix}$$

- choose f such that the number of misclassified observations is minimized

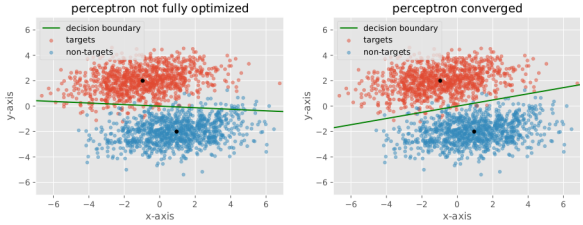
Repetition Perceptron

- algorithm to reduce the classification error iterative with gradient descent
- include the bias β into $\mathbf{w}$ (see first lecture) $\tilde{w} = \begin{pmatrix} -\beta \\ w_1 \\ \vdots \\ w_d \end{pmatrix}$ $\tilde{x} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{pmatrix}$ $w^T x - \beta = \tilde{w}^T \tilde{x}$

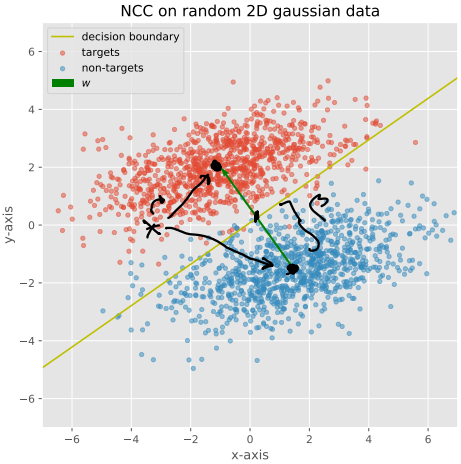
Repetition Perceptron

- algorithm to reduce the classification error iterative with gradient descent
- include the bias β into $\mathbf{w}$ (see first lecture)

1 initialize $\mathbf{w}$ randomly
2 until convergence (runtime constraint or no misclassifications)
   a pick a random misclassified point $\mathbf{x}_k$
   b update the weights with gradient descent: $\mathbf{w}_{new} = \mathbf{w}_{old} - \eta \mathbf{x}_k y_k$



Repetition Nearest Centroid Classifier



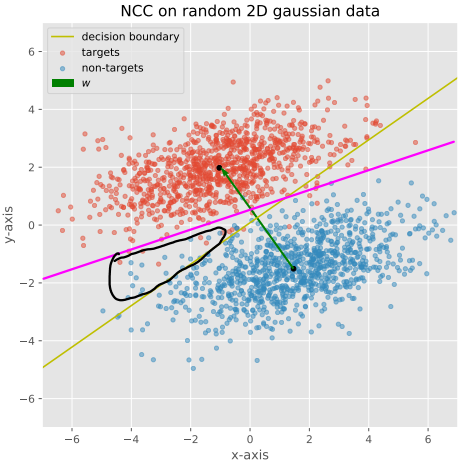
- compare distance of data points $\mathbf{x}$ to class means
- decide for class with closer center

$$\mathbf{w} = \mathbf{w}_o - \mathbf{w}_\Delta$$

$$\beta = \frac{1}{2}(\mathbf{w}_o^T \mathbf{w}_o - \mathbf{w}_\Delta^T \mathbf{w}_\Delta)$$

$$\rightarrow f(\mathbf{x}) = sign(\mathbf{w}^T \mathbf{x} - \beta) \leftarrow$$

Repetition Nearest Centroid Classifier



- compare distance of data points $\mathbf{x}$ to class means
- decide for class with closer center

$$\mathbf{w} = \mathbf{w}_o - \mathbf{w}_\Delta$$
$$\beta = \frac{1}{2}(\mathbf{w}_o^T \mathbf{w}_o - \mathbf{w}_\Delta^T \mathbf{w}_\Delta)$$
$$f(\mathbf{x}) = sign(\mathbf{w}^T \mathbf{x} - \beta)$$

- correlation not considered, thus often not optimal

Task 1: Bias term of the NCC

Show that

$$\beta = \frac{1}{2}(\mathbf{w}_o^T\mathbf{w}_o - \mathbf{w}_\Delta^T\mathbf{w}_\Delta) = \mathbf{w}^T\left(\frac{\mathbf{w}_o + \mathbf{w}_\Delta}{2}\right)$$

$$w^T\left(\frac{w_o+w_\Delta}{2}\right) = w^T\left(\frac{1}{2}(w_o+w_\Delta)\right) = \frac{1}{2}w^T(w_o+w_\Delta)$$

$$= \frac{1}{2}(w_o-w_\Delta)^T(w_o+w_\Delta)$$

$$= \frac{1}{2}\left(w_o^Tw_o + w_o^Tw_\Delta - w_\Delta^Tw_o - w_\Delta^Tw_\Delta\right)$$

$$= \frac{1}{2}\left(w_o^Tw_o - w_\Delta^Tw_\Delta\right)$$

$(\alpha v)^T w = \alpha(v^T w)$

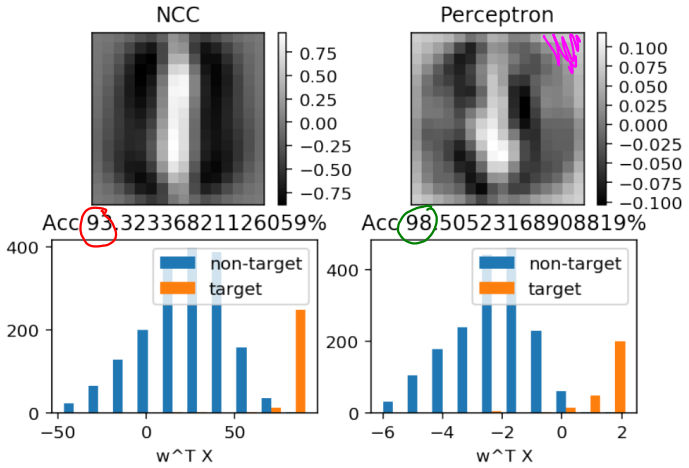
$v^T w = w^T v$

Task 1: Bias term of the NCC

Assignment 1: High accuracy but "lousy" weight vector?

NCC

$w = w_0 - w_\Delta$



Perceptron

w

$w_n = w_0 - n x_k y_k$

Assignment 1: High accuracy but "lousy" weight vector?

- purpose of weight vector is two-fold:
  - amplify important features
  - suppress unimportant features
- Example with some noise ϵ

Assignment 1: High accuracy but "lousy" weight vector?

- purpose of weight vector is two-fold:
  - amplify important features
  - suppress unimportant features
- Example with some noise ϵ

$$\mathbf{x} = \begin{bmatrix} y + \epsilon \\ \epsilon \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\mathbf{w}^T\mathbf{x} - \beta = y + \epsilon - \epsilon - 0 = y$$

$$\omega^T x = \begin{pmatrix} y+\epsilon \\ \epsilon \end{pmatrix}^T \begin{pmatrix} 1 \\ -1 \end{pmatrix} = y + \epsilon - \epsilon = y$$

$\rightarrow$ correct classification of class label $y \in \mathcal{C}$

$\rightarrow$ $\mathbf{w}$ is the optimal weight vector for this data sample

Covariance Matrix

$x_1, \ldots, x_n \in \mathbb{R}^d$

- let $X_1, \ldots, X_d$ be random variables (can hold the different values of d features)
- their covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ is defined such that

$$\Sigma_{i,j} = \Sigma_{j,i} = \mathbb{E}\left[(X_i - \mathbb{E}(X_i))(X_j - \mathbb{E}(X_j))\right]$$

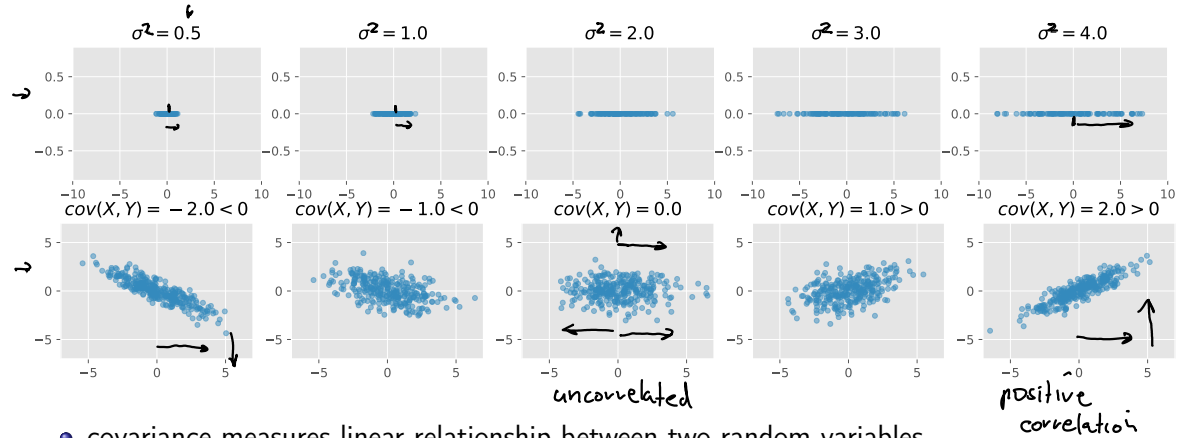
Covariance Matrix

- let $X_1, \ldots, X_d$ be random variables (can hold the different values of d features)
- their covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ is defined such that

$$\Sigma_{i,j} = \Sigma_{j,i} = \mathbb{E}\left[(X_i - \mathbb{E}(X_i))(X_j - \mathbb{E}(X_j))\right]$$

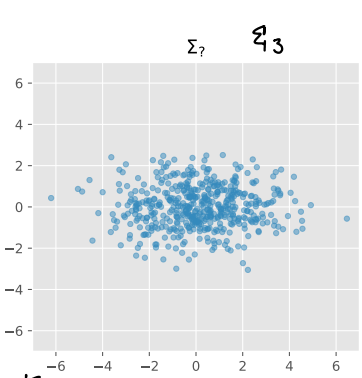
- covariance measures linear relationship between two random variables
- for $i = j$ (the diagonal) we get $var(X_i)$ (expected squared deviation from mean)

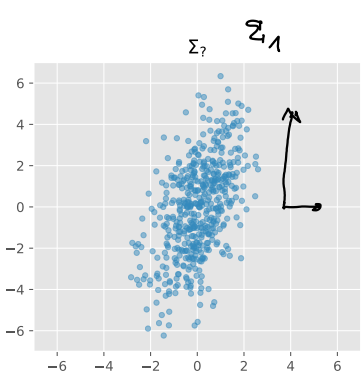
Covariance Matrix

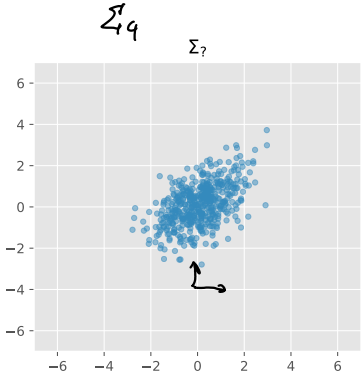


- covariance measures linear relationship between two random variables
- for $i = j$ (the diagonal) we get $var(X_i)$ (expected squared deviation from mean)

Covariance Matrix - Quiz

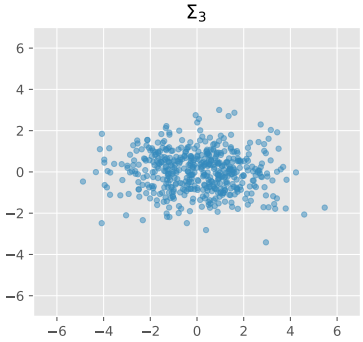


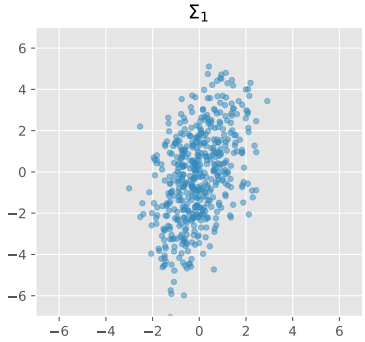


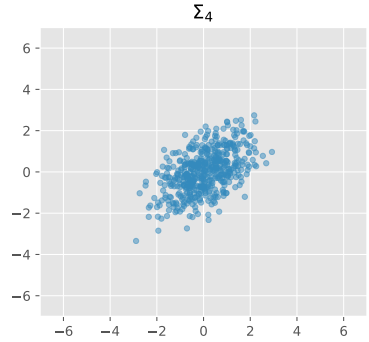


$$\Sigma_1 = \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Sigma_4 = \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}, \quad \Sigma_5 = \begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix}$$

Covariance Matrix - Quiz Solution





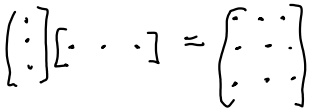


$$\Sigma_1 = \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Sigma_4 = \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}, \quad \Sigma_5 = \begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix}$$

Empirical Covariance Matrix

- let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ be n observations of d random variables
- let be $\bar{\mathbf{x}} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k$ the empirical mean
- the empirical covariance matrix is defined as

$$\Sigma_X = \frac{1}{n} \sum_{k=1}^{n} (\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{x}_k - \bar{\mathbf{x}})^T$$



Empirical Covariance Matrix

- let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ be n observations of d random variables
- let be $\overline{\mathbf{x}} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k$ the empirical mean
- the empirical covariance matrix is defined as

$$\Sigma_X = \frac{1}{n} \sum_{k=1}^{n} (\mathbf{x}_k - \overline{\mathbf{x}})(\mathbf{x}_k - \overline{\mathbf{x}})^T$$

- let $X = \left(\mathbf{x}_1 - \overline{\mathbf{x}}, \ldots, \mathbf{x}_n - \overline{\mathbf{x}}\right) \in \mathbb{R}^{d \times n}$ denote the matrix of the centered data, then

$$\Sigma_X = \frac{1}{n} X X^T$$

Task 2.1

Compute the empirical covariance matrix for the following four data points

$$X = \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{2,4}$$

$$\bar{x} = \frac{1}{4}\begin{bmatrix} -1-1+1+1 \\ -1+0+0+1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Sigma_X = \frac{1}{n}XX^T = \frac{1}{4}\begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & -1 \\ -1 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

Task 2.1 - practice with numpy

→ see jupyter notebook

Task 2.1 - practice with numpy

→ see jupyter notebook

- unbiased estimation of the covariance matrix

$$\Sigma_X = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{x}_k - \bar{\mathbf{x}})^T$$

- instead of

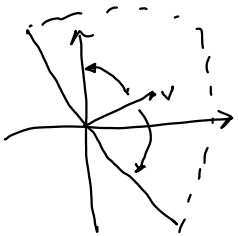
$$\Sigma_X = \frac{1}{n} \sum_{k=1}^{n} (\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{x}_k - \bar{\mathbf{x}})^T$$

- for big n better estimation of the covariance
- based on statistics methods

Task 2.2 - properties of covariance matrices

- covariance matrices are always symmetric $\Sigma_X^T = \Sigma_X$

- covariance matrices are (semi-)positive definite, that is, that vor every vector $\mathbf{v} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ the following inequality holds



$$\mathbf{v}^T \Sigma_X \mathbf{v} \geq 0$$

$$v^T v' \geq 0$$

$$v^T v' = \cos(\alpha) \, \|v\| \|v'\|$$

$$\geq 0 \qquad \geq 0$$

$$\Rightarrow \alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

Task 2.2 - properties of covariance matrices

- covariance matrices are always symmetric $\Sigma_X^T = \Sigma_X$
- covariance matrices are (semi-)positive definite, that is, that vor every vector $\mathbf{v} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ the following inequality holds
$$\mathbf{v}^T \Sigma_X \mathbf{v} \geq 0$$
- this is equivalent to the observation $\lambda \geq 0$ for all eigenvalues λ of Σ_X

Task 2.2 - properties of covariance matrices

Proof I: all eigenvalues $\lambda \geqslant 0 \Rightarrow \forall \mathbf{v} \in \mathbb{R}^d \setminus \{\mathbf{0}\} : \mathbf{v}^T \Sigma_X \mathbf{v} \geq 0$

$y = U^T x$

$y^T = (U^T x)^T = x^T U$

$Uy = x$

$\Sigma_X = U \Lambda U^T$ $\quad x \in \mathbb{R}^d \setminus \{0\}$: $\quad x^T \Sigma_X x = x^T U \Lambda U^T x$

$= y^T \Lambda y$

$= \begin{bmatrix} y_1 & \cdots & y_d \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_d \end{bmatrix}$

$= \begin{bmatrix} y_1 & \cdots & y_d \end{bmatrix} \begin{pmatrix} y_1 \lambda_1 \\ \vdots \\ y_d \lambda_d \end{pmatrix}$

$x^T \Sigma_X x \geqslant 0 \quad \Leftarrow \quad = y_1^2 \lambda_1 + y_2^2 + \lambda_2 + \ldots + y_d^2 \lambda_d \geqslant 0$

Task 2.2 - properties of covariance matrices

Proof II: $\forall \mathbf{v} \in \mathbb{R}^d \setminus \{\mathbf{0}\} : \mathbf{v}^T \Sigma_X \mathbf{v} \geq 0 \Rightarrow$ all eigenvalues $\lambda \geqslant 0$

$\Sigma_x u = u\lambda$ $\qquad$ $\underbrace{u^T \Sigma_X u}_{\geqslant 0} = u^T(\lambda u) = \lambda u^T u = \lambda \underbrace{\|u\|^2}_{\geq 0}$

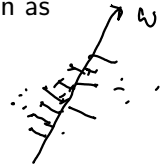
$v^T A v > 0$; $\lambda_i > 0$ $\qquad$ positive definite

$v^T \Sigma_X v \geqslant 0$; $\lambda_i \geqslant 0$ $\qquad$ (semi) PD

Task 2.3 - Variance in a certain direction

The variance of centered data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ in direction $\mathbf{w} \in \mathbb{R}^d$ is given as

$$\frac{1}{n} \sum_{k=1}^{n} (\mathbf{w}^T \mathbf{x}_k)^2.$$



We show that this is equal to

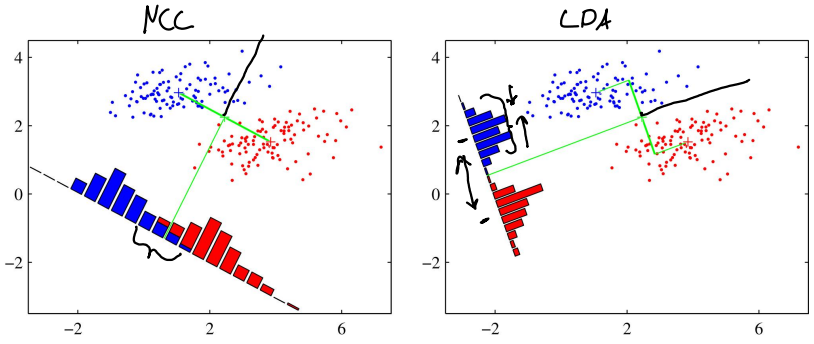
$$\frac{1}{n} \sum_{k=1}^{n} (w^T x_k)^2 = \frac{1}{n} \sum_{k=1}^{n} (w^T x_k)(x_k^T w) = w^T \left[\frac{1}{n} \sum_{k=1}^{n} x_k x_k^T \right] w$$

$w^T \Sigma_X w \geq 0$

$$= w^T \Sigma_X w$$

Task 2.3 - Variance in a certain direction

Linear Discriminant Analysis



LDA gives us a weight vector **w** that

- maximizes the mean class difference
- minimizes the variance in each class

Linear Discriminant Analysis

- given class means $\mathbf{w}_o, \mathbf{w}_\Delta \in \mathbb{R}^d$ and number of points n_o, n_Δ per class

$$\mathbf{w} = \Sigma_X^{-1}(\mathbf{w}_o - \mathbf{w}_\Delta)$$

$$\beta = \mathbf{w}^T \left(\frac{\mathbf{w}_o + \mathbf{w}_\Delta}{2} \right) \left[+ \log \left(\frac{n_o}{n_\Delta} \right) \right]$$

$n_o = n_\Delta$

$\log(1) = 0$

Linear Discriminant Analysis

- given class means $\mathbf{w}_o, \mathbf{w}_\Delta \in \mathbb{R}^d$ and number of points n_o, n_Δ per class

$$\mathbf{w} = \Sigma_X^{-1}(\mathbf{w}_o - \mathbf{w}_\Delta)$$
$$\beta = \mathbf{w}^T \left(\frac{\mathbf{w}_o + \mathbf{w}_\Delta}{2} \right) \left[+ \log \left(\frac{n_o}{n_\Delta} \right) \right]$$

- the term in square brackets is zero for $n_o = n_\Delta$

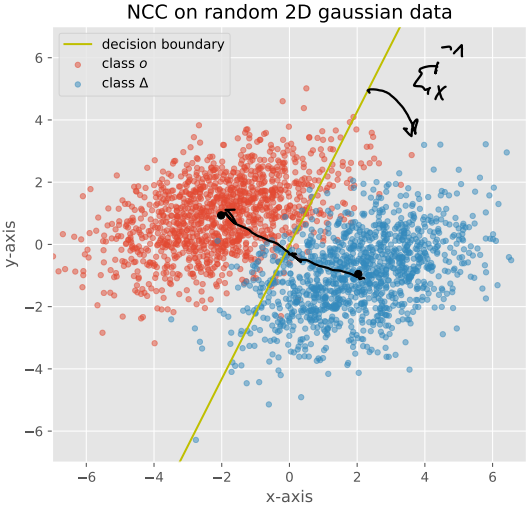
Linear Discriminant Analysis

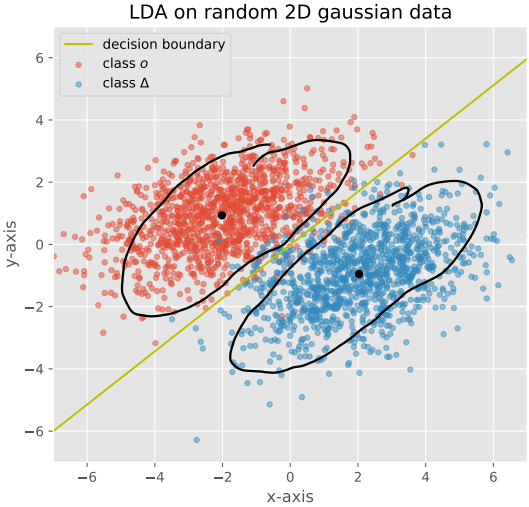
- given class means $\mathbf{w}_o, \mathbf{w}_\Delta \in \mathbb{R}^d$ and number of points n_o, n_Δ per class

$$\mathbf{w} = \Sigma_X^{-1}(\mathbf{w}_o - \mathbf{w}_\Delta)$$
$$\beta = \mathbf{w}^T \left(\frac{\mathbf{w}_o + \mathbf{w}_\Delta}{2} \right) \left[+ \log \left(\frac{n_o}{n_\Delta} \right) \right]$$

- the term in square brackets is zero for $n_o = n_\Delta$
- LDA is the optimal classifier when
  - both classes are gaussian distributed
  - both covariance matrices Σ_X are equal and known

Linear Discriminant Analysis

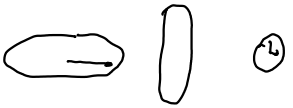




Task 3 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

1. How does the covariance matrix look like? Which class of matrix is that?



$$\Sigma_K = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$ diagonal

Task 3 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

1. How does the covariance matrix look like? Which class of matrix is that?

$$\Sigma_X = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$ this is a diagonal matrix

Task 3 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

1. How does the covariance matrix look like? Which class of matrix is that?

$$\Sigma_X = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$ this is a diagonal matrix

2. Which property is represented by the diagonal elements of the covariance matrix?

Task 3 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

1. How does the covariance matrix look like? Which class of matrix is that?

$$\Sigma_X = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$ this is a diagonal matrix

2. Which property is represented by the diagonal elements of the covariance matrix?
   - σ_1^2 is the variance along the x-axis
   - σ_2^2 is the variance along the y-axis

Task 3 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

1. How does the covariance matrix look like? Which class of matrix is that?

$$\Sigma_X = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \qquad \text{this is a diagonal matrix}$$

2. Which property is represented by the diagonal elements of the covariance matrix?
   - σ_1^2 is the variance along the x-axis
   - σ_2^2 is the variance along the y-axis

- Note that not necessarily $\sigma_1^2 = \sigma_2^2$

Task 3.3 - NCC vs. LDA

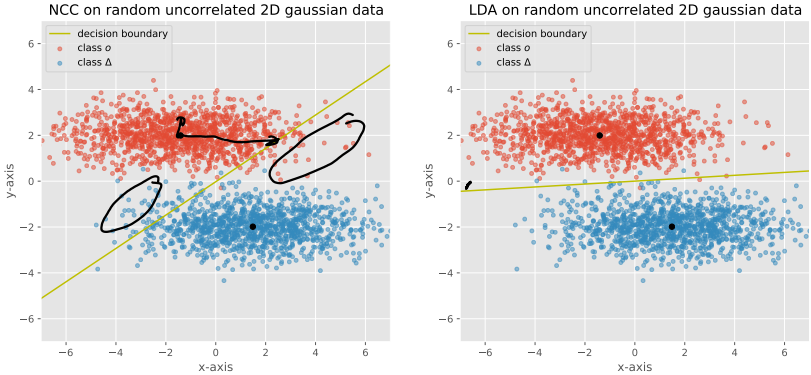
Imagine a classification task with two equally gaussian distributed but uncorrelated classes

- Make a sketch of an imaginary data set with uncorrelated features in which NCC and LDA would classify some data points differently.

Task 3.3 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

- Make a sketch of an imaginary data set with uncorrelated features in which NCC and LDA would classify some data points differently.



Task 3.4 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

- In which case are NCC and LDA equivalent? Proof that they are equivalent in this case.

Task 3.4 - NCC vs. LDA

Imagine a classification task with two equally gaussian distributed but uncorrelated classes

- In which case are NCC and LDA equivalent? Proof that they are equivalent in this case.

$$\Sigma_X = \alpha \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}, \qquad \alpha \in \mathbb{R}$$

Task 3.4 - NCC vs. LDA

$$0 < w^T x - \frac{1}{2} w^T (\omega_0 + \omega_1) \Leftrightarrow$$

$$0 < \left(\Sigma_x^{-1}(\omega_0 - \omega_1)\right)^T x - \frac{1}{2}\left(\Sigma_x^{-1}(\omega_0 - \omega_1)\right)^T (\omega_0 + \omega_1) \Leftrightarrow$$

$$0 < \left(\frac{1}{\alpha} I (\omega_0 - \omega_1)\right)^T x - \frac{1}{2}\left(\frac{1}{\alpha} I (\omega_0 - \omega_1)\right)^T (\omega_0 + \omega_1) \Leftrightarrow$$

$$0 < \frac{1}{\alpha}\left(I(\omega_0 - \omega_1)\right)^T x - \frac{1}{2}\left(I(\omega_0 - \omega_1)\right)^T (\omega_0 + \omega_1) \Leftrightarrow$$

$$0 < \frac{1}{\alpha}\left((\omega_0 - \omega_1)^T x - \frac{1}{2}(\omega_0 - \omega_1)^T (\omega_0 + \omega_1)\right) \quad | \cdot \alpha$$

$\alpha \geq 0$

$\alpha < 0$

$$w_{NCC} = \omega_0 - \omega_1$$

$$\beta_{NCC} = \frac{1}{2}(\omega_0 - \omega_1)^T (\omega_0 + \omega_1)$$

$$0 < w_{NCC}^T x - \beta_{NCC}$$

$$\Sigma_x = \alpha I \Leftrightarrow \Sigma_x^{-1} = \frac{1}{\alpha} I$$

$$\Sigma_x \Sigma_x^{-1} = (\alpha I)\left(\frac{1}{\alpha} I\right) = \alpha \frac{1}{\alpha}(I I) = I$$

$$-1 \cdot w = \tilde{w}$$

Orthogonal Matrices

- matrices with orthonormal column vectors are called orthogonal matrices

$$U = [u_1, \ldots, u_d] \qquad u_i^T u_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

$$\|u_i\| = \|u_i\|^2 = 1$$

Orthogonal Matrices

- matrices with orthonormal column vectors are called orthogonal matrices
- they always have a full rank, thus they are invertible

Orthogonal Matrices

- matrices with orthonormal column vectors are called orthogonal matrices
- they always have a full rank, thus they are invertible
- they are numerical stable when applying them to vectors or other matrices

Orthogonal Matrices

- matrices with orthonormal column vectors are called orthogonal matrices
- they always have a full rank, thus they are invertible
- they are numerical stable when applying them to vectors or other matrices
- for a orthogonal matrix the following equations holds

$$U^T = U^{-1}$$
$$U^T U = U U^T = I$$

Orthogonal Matrices

- matrices with orthonormal column vectors are called orthogonal matrices
- they always have a full rank, thus they are invertible
- they are numerical stable when applying them to vectors or other matrices
- for a orthogonal matrix the following equations holds

$$U^T = U^{-1}$$
$$U^T U = U U^T = I$$

- they not always are symmetric

$V^T = V^{-\frac{1}{2}}$ $V = V^{-1}$ $VV = I$

$V = V^T$

Orthogonal Matrices

- matrices with orthonormal column vectors are called orthogonal matrices
- they always have a full rank, thus they are invertible
- they are numerical stable when applying them to vectors or other matrices
- for a orthogonal matrix the following equations holds

$$U^T = U^{-1}$$
$$U^T U = U U^T = I$$

- they not always are symmetric
- rotation matrices R are orthogonal matrices with $det|R| = +1$

Eigendecomposition

- symmetric matrices $A \in \mathbb{R}^{n \times n}$ have n orthogonal eigenvectors and n real eigenvalues

Eigendecomposition

- symmetric matrices $A \in \mathbb{R}^{n \times n}$ have n orthogonal eigenvectors and n real eigenvalues
- thus they can be decomposed into

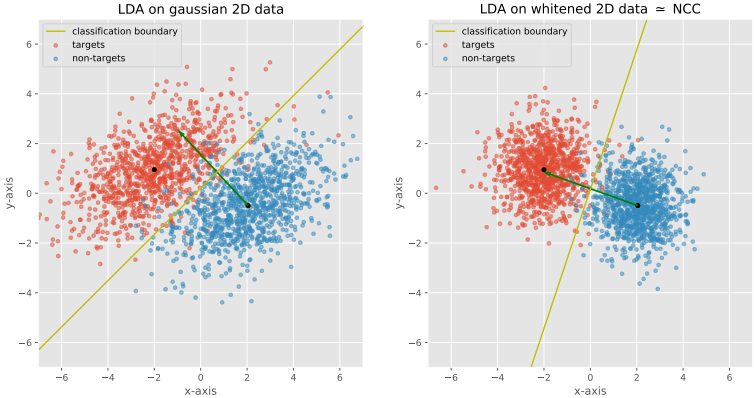
$$A = U \Lambda U^T$$

- U being a orthogonal matrix with the eigenvectors in the columns
- Λ being a diagonal matrix with the eigenvalues at the diagonal

LDA as whitening followed by NCC

LDA first whitens the data and then applies the NCC

$$\Sigma_S^{-1} \times = I$$



LDA as whitening followed by NCC

LDA first whitens the data and then applies the NCC

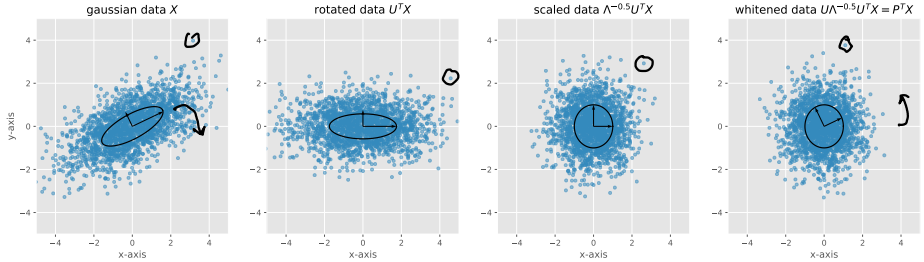
$$
\begin{aligned}
\mathbf{w}^T\mathbf{x} &= (\Sigma_X^{-1}(\mathbf{w}_o - \mathbf{w}_\Delta))^T\mathbf{x} \\
&= (\mathbf{w}_o - \mathbf{w}_\Delta)^T\Sigma_X^{-1}\mathbf{x} \\
&= (\mathbf{w}_o - \mathbf{w}_\Delta)^T U\Lambda^{-1}U^T\mathbf{x} \\
&= (\mathbf{w}_o - \mathbf{w}_\Delta)^T U\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}U^T\mathbf{x} \\
&= \underbrace{(\mathbf{w}_o - \mathbf{w}_\Delta)^T U\Lambda^{-\frac{1}{2}}}_{\text{mean class difference of whitened data}}\underbrace{\Lambda^{-\frac{1}{2}}U^T\mathbf{x}}_{\text{whitened } \mathbf{x}}
\end{aligned}
$$

$w_{NCC}^T \tilde{x}$

Whitening as a "standalone" transformation

In the following tasks we define the mapping $P : \mathbb{R}^d \to \mathbb{R}^d$ as follows

$$\mathbf{z}_k := P^T \mathbf{x}_k = U \Lambda^{-\frac{1}{2}} U^T \mathbf{x}_k = \Sigma_X^{-\frac{1}{2}} \mathbf{x}_k$$



Task 4.1 - P is symmetric

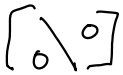
Show that $P = U\Lambda^{-\frac{1}{2}}U^T$ is a symmetric matrix

$P = P^T$

$P^T = \left(U\Lambda^{-\frac{1}{2}}U^T\right)^T = U^{TT}\Lambda^{-\frac{1}{2}T}U^T$

$= U\Lambda^{T-\frac{1}{2}}U^T$

$= U\Lambda^{-\frac{1}{2}}U^T = P$



Task 4.2 - Matrix square roots of positive definite matrices

Show that ~~$\Sigma_X^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^T$~~ is a a valid square root of a positive definite matrix

$$\Sigma_X^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^T \leftarrow$$

$U^TU = I$

$$\Sigma_X^{\frac{1}{2}}\Sigma_X^{\frac{1}{2}} = \Sigma_X$$

$$\Sigma_X^{\frac{1}{2}}\Sigma_X^{\frac{1}{2}} = (U\Lambda^{\frac{1}{2}}U^T)(U\Lambda^{\frac{1}{2}}U^T)$$

$$= U\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}U^T$$

$$= U\begin{bmatrix}\lambda_1^{\frac{1}{2}} & & 0\\ & \ddots & \\ 0 & & \lambda_d^{\frac{1}{2}}\end{bmatrix}\begin{bmatrix}\lambda_1^{\frac{1}{2}} & & 0\\ & \ddots & \\ 0 & & \lambda_d^{\frac{1}{2}}\end{bmatrix}U^T$$

$$= U\Lambda U^T = \Sigma_X$$

Task 4.3 - Inverse of matrix square root

Show that $P = U\Lambda^{-\frac{1}{2}}U^T$ is a valid inverse of the square root of a positive definite matrix

$$\Sigma_x^{-1} = U\Lambda^{-1}U^T = PP = U\Lambda^{-\frac{1}{2}}U^T U\Lambda^{-\frac{1}{2}}U^T$$
$$= U\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}U^T$$
$$= U\Lambda^{-1}U^T$$
$$= \Sigma_x^{-1}$$

Task 4.4 - Covariance Matrix of whitened data is the identity

Show that the covariance matrix of whitened data $P^T X$ is the identity Matrix. Assume that the data has zero mean.

$\Sigma_X = \frac{1}{n} X X^T$

$Z = P^T X \quad \Sigma_Z = \Sigma_{P^T X} = \frac{1}{n} (P^T X)(P^T X)^T$

$= \frac{1}{n} P^T X X^T P = P^T \left(\frac{1}{n} X X^T\right) P$

$= P^T (\Sigma_X) P$

$= P^T U \Lambda U^T P = P U \Lambda U^T P$

$= U \Lambda^{-\frac{1}{2}} U^T U \Lambda U^T U \Lambda^{-\frac{1}{2}} U^T$

$= U \Lambda^{-\frac{1}{2}} \Lambda \Lambda^{-\frac{1}{2}} U^T = U \Lambda^{-\frac{1}{2}} \Lambda^{-\frac{1}{2}} \Lambda U^T$

$= U \Lambda^{-1} \Lambda U^T = U U^T = I$

Task 4.5 - Equivalence of NCC and LDA on whitened data

Proof that classification with LDA is equivalent to classification with NCC on whitened data

$$w_{NCC}^T z = \left(P^T w_0 - P^T w_A\right)^T P^T x = \left(P^T (w_0 - w_A)\right)^T P^T x$$

$$= (w_0 - w_A)^T P P^T x$$

$$= (w_0 - w_A)^T \Sigma_X^{-1} x$$

$$= \underbrace{\left(\Sigma_X^{-1} (w_0 - w_A)\right)^T}_{w_{LDA}^T} x$$

$$P = \Sigma_X^{-\frac{1}{2}}$$

$$PP = \Sigma_X^{-\frac{1}{2}} \Sigma_X^{-\frac{1}{2}} = \Sigma_X^{-1}$$

Task 4.5 - Equivalence of NCC and LDA on whitened data

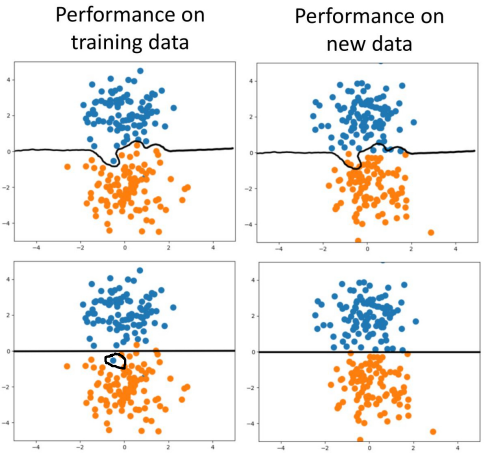
Proof that classification with LDA is equivalent to classification with NCC on whitened data

Generalization

- goal of classification is generalization
- correct categorization of new, unseen data

- correct classification on training data not implies that a classifier is optimal

Overfitting



overfit

- always compare test accuracy on new data with training accuracy
- should be as similar as possible

Q&A

- The videos are pretty long. I am sorry for that!
- I hope I didn't was to fast or slow. I am happy about feedback :)

- next office hour for questions: wednesday, may 13 at 4pm on zoom