technische universität dortmund

Experimentelle Physik Vb
Astroteilchenphysik

Vorlesung
Statistische Methoden der Datenanalyse
Prof. Dr. Dr. Wolfgang Rhode

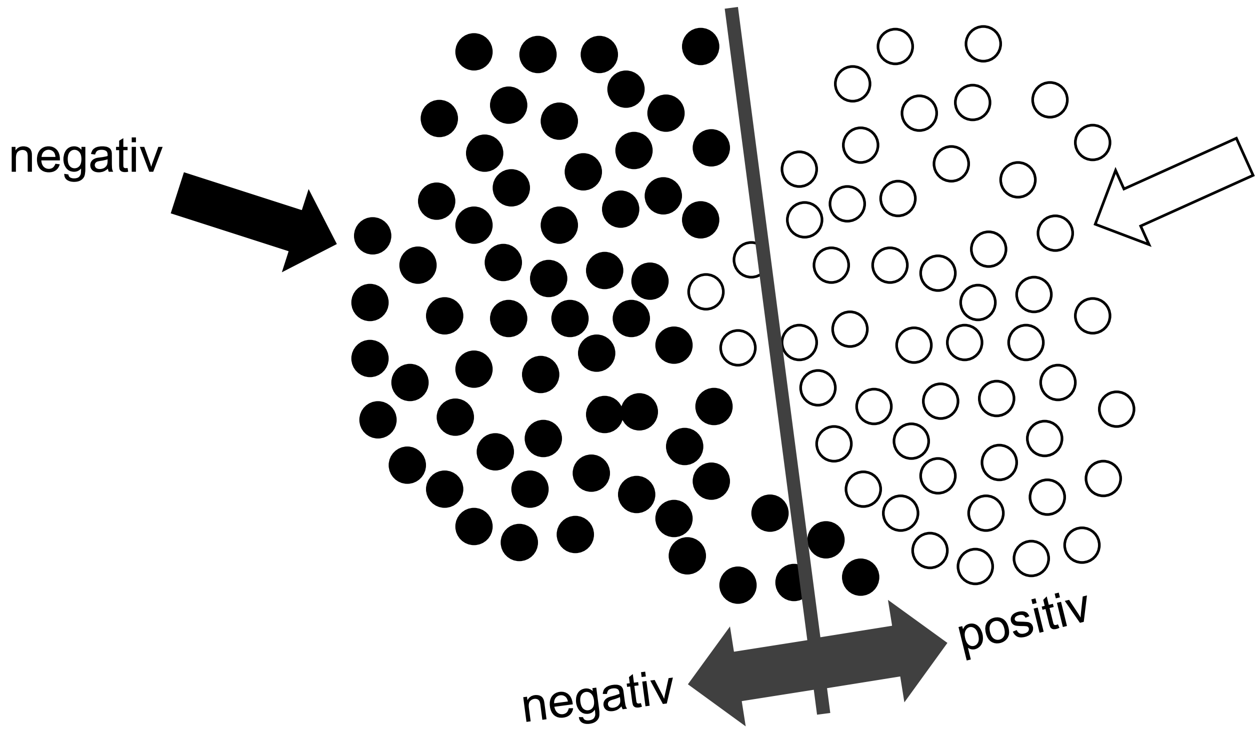
Data-Mining – Teil 1

Data-Mining – Teil 1

- Diskriminanzanalyse
- Grundbegriffe des Data-Minings
- Typischer Aufbau eines Data-Mining-Prozesses
- Datenauswahl
- Datenbereinigung
- Datenreduktion und –transformation
  - Hauptkomponentenanalyse
  - Feature Selection

Motivation

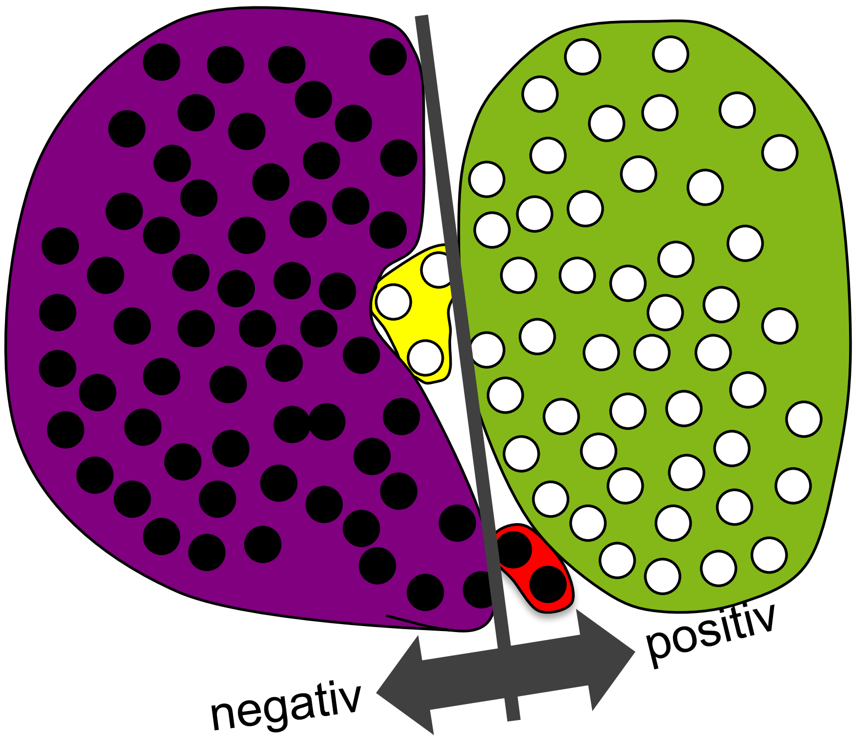
- Ziel ist es die Punkte in zwei Populationen zu trennen
- Im Monte Carlo ist die Zugehörigkeit der Elemente bekannt
- Idee: Suche im Monte-Carlo den *besten* eindimensionalen Schnitt



Population$_1$ = positiv
Population$_2$ = negativ

Motivation

- Was ist der beste Schnitt?
  - true positiv (tp)
    - "positiv" Elemente die nach der Trennung im "positiv" Bereich liegen
  - false negativ (fn)
    - "positive" Elemente die nach der Trennung im "negativ" Bereich liegen
  - true negativ (tn)
    - "negativ" Elemente die nach der Trennung im "negativ" Bereich liegen
  - false positiv (fp)
    - "negativ" Elemente die nach der Trennung im "positiv" Bereich liegen



Motivation

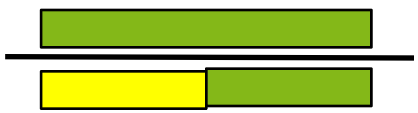
- Was ist der beste Schnitt?
  - Qualitätsmaße für zwei Populationen:
    - Reinheit (bzgl. Population$_1$):

Reinheit = („precision“)

$$\frac{tp}{tp + fp}$$

    - Effizienz (bzgl. Population$_1$):

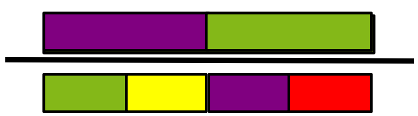
Effizienz = („recall“)



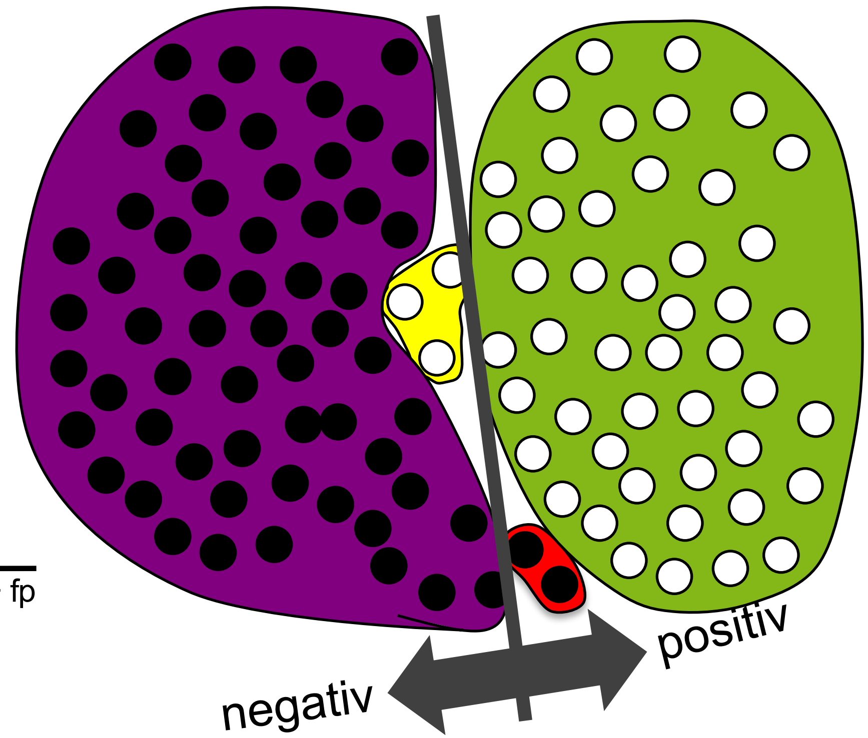
$$\frac{tp}{tp + fn}$$

    - Genauigkeit:

Genauigkeit = („accuracy“)



$$\frac{tp + tn}{tp + fn + tn + fp}$$



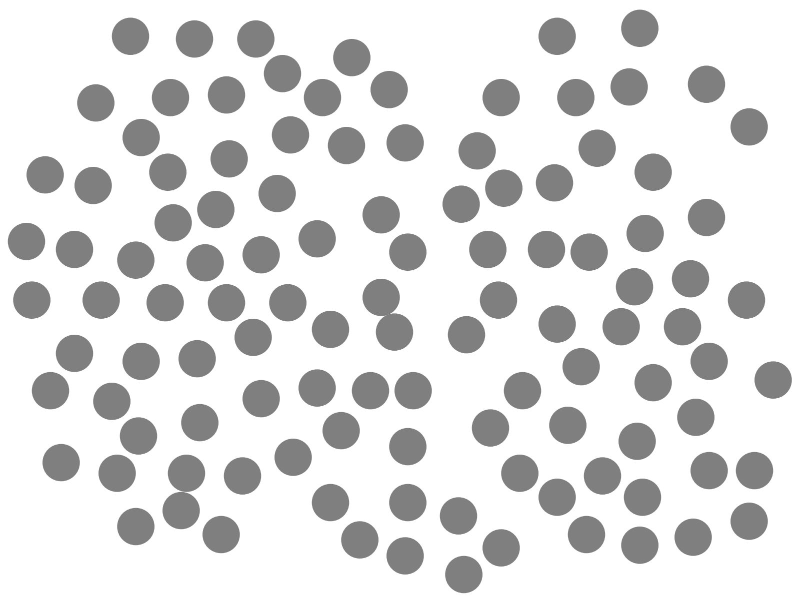
Motivation

- Was ist der beste Schnitt?
  - Qualitätsmaße für zwei Populationen:

		True condition			
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma\ \text{Condition positive}}{\Sigma\ \text{Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV), Precision $= \frac{\Sigma\ \text{True positive}}{\Sigma\ \text{Test outcome positive}}$	False discovery rate (FDR) $= \frac{\Sigma\ \text{False positive}}{\Sigma\ \text{Test outcome positive}}$
	Predicted condition negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) $= \frac{\Sigma\ \text{False negative}}{\Sigma\ \text{Test outcome negative}}$	Negative predictive value (NPV) $= \frac{\Sigma\ \text{True negative}}{\Sigma\ \text{Test outcome negative}}$
	Accuracy (ACC) $= \frac{\Sigma\ \text{True positive} + \Sigma\ \text{True negative}}{\Sigma\ \text{Total population}}$	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma\ \text{True positive}}{\Sigma\ \text{Condition positive}}$	False positive rate (FPR), Fall-out $= \frac{\Sigma\ \text{False positive}}{\Sigma\ \text{Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma\ \text{False negative}}{\Sigma\ \text{Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma\ \text{True negative}}{\Sigma\ \text{Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

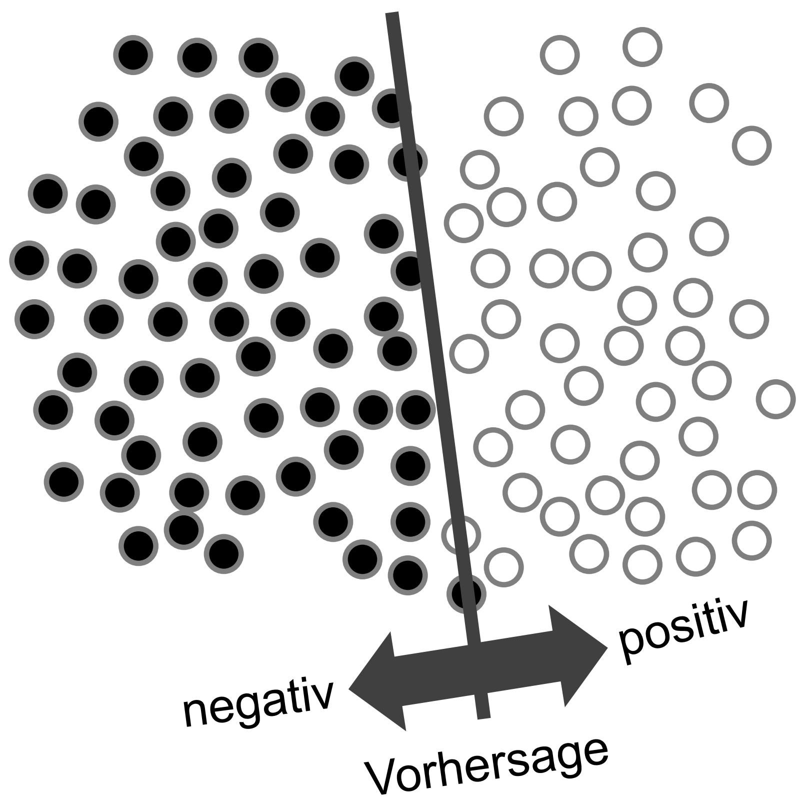
Motivation

- Ziel ist es bei die Punkte in zwei Populationen zu trennen
- Im Monte Carlo ist die Zugehörigkeit der Elemente bekannt
- Idee: Suche im Monte-Carlo den *besten* eindimensionalen Schnitt



Motivation

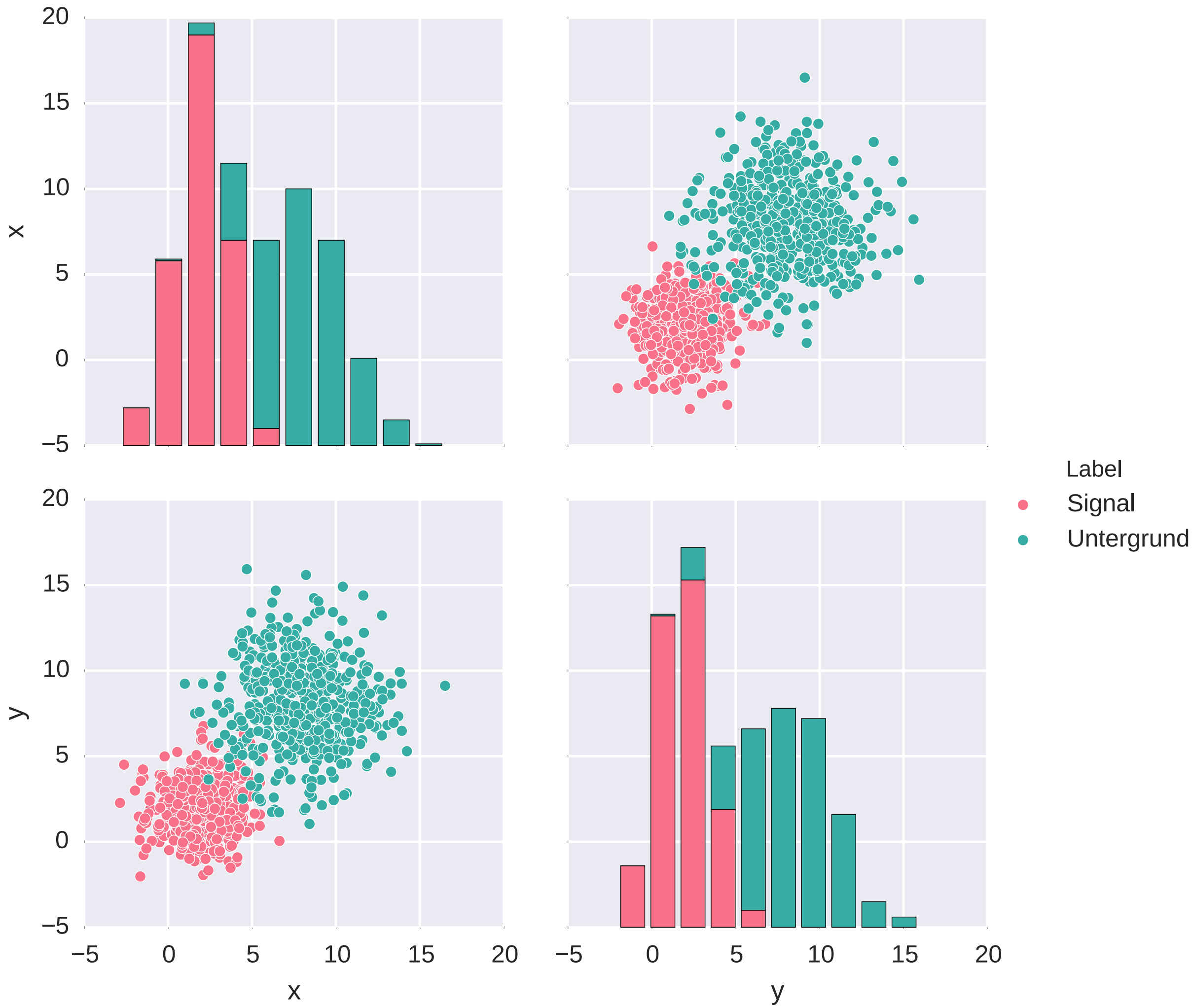
- Ziel ist es bei die Punkte in zwei Populationen zu trennen
- Im Monte Carlo ist die Zugehörigkeit der Elemente bekannt
- Idee: Suche im Monte-Carlo den *besten* (n-1)-dimensionalen Schnitt



Population$_1$ = positiv
Population$_2$ = negativ

Beispiel:

- Aufgabe: Trennen zweier Populationen
  - Grün: „Untergrund“
  - Rot: „Signal“
- Elemente beider Populationen über Wertepaare (x, y) beschrieben
  - Untergrund: Gaußverteilung mit dem Mittelwert (8, 8) und der Standardabweichungen (2.5, 2.5)
  - Signal: Gaußverteilung mit dem Mittelwert (2, 2) und der Standardabweichungen (1.5, 1.5)
- Suche den *besten* eindimensionalen Schnitt (trennende Hyperebene)



Data-Mining – Teil 1

Beispiel:

- Aufgabe: Trennen zweier Populationen
  - Grün: „Untergrund“
  - Rot: „Signal“
- Elemente beider Populationen über Wertepaare (x, y) beschrieben
  - Untergrund: Gaußverteilung mit dem Mittelwert (8, 8) und der Standardabweichungen (2.5, 2.5)
  - Signal: Gaußverteilung mit dem Mittelwert (2, 2) und der Standardabweichungen (1.5, 1.5)

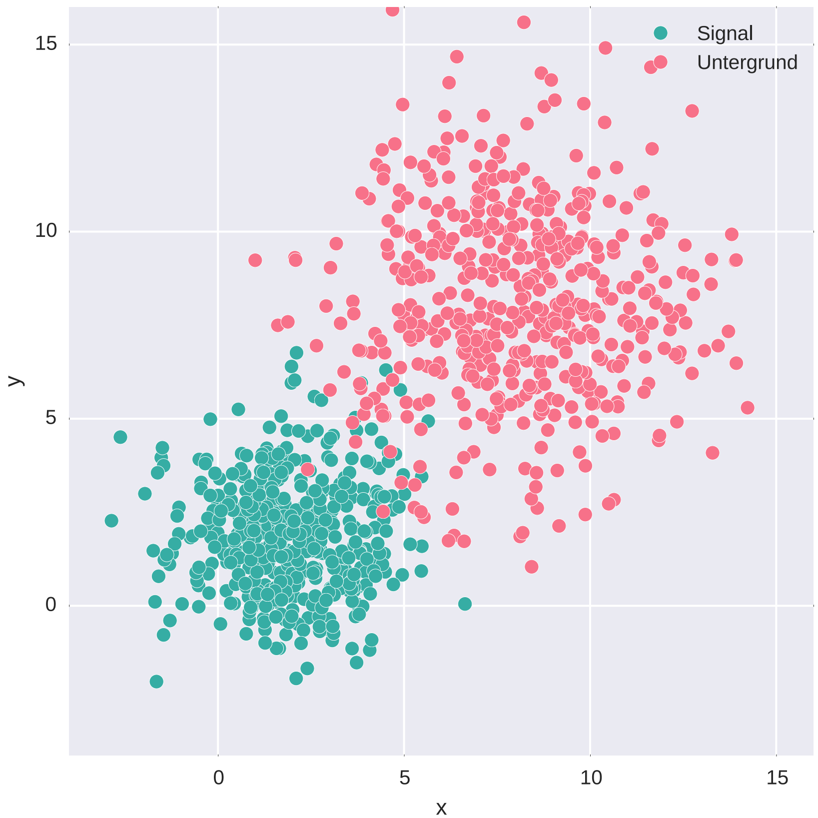
- Suche den *besten* eindimensionalen Schnitt (trennende Hyperebene)

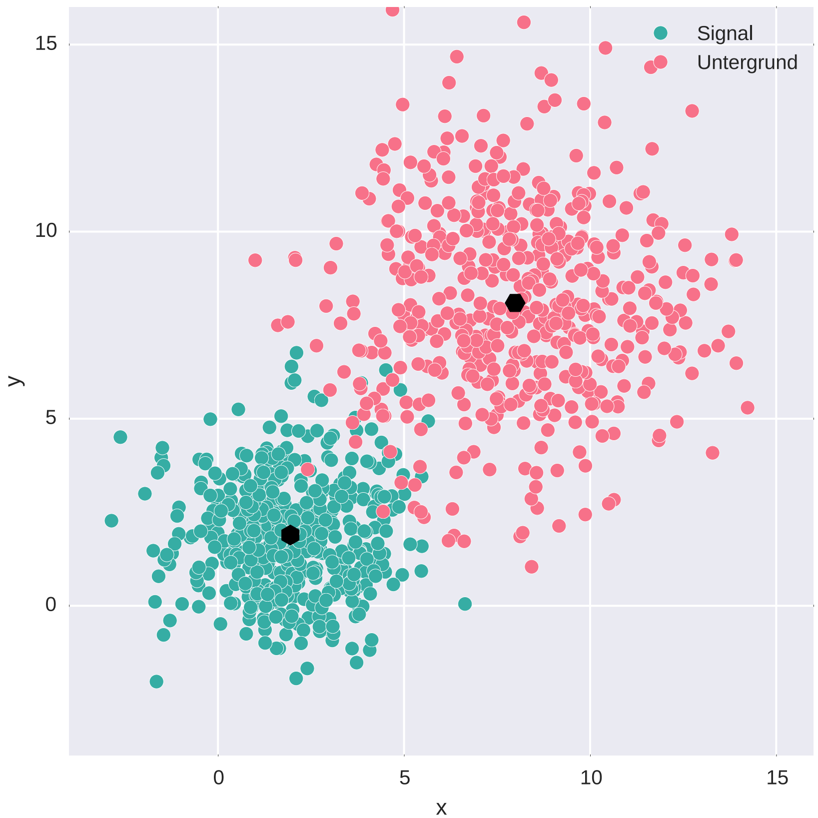
  → Projektion auf den Normalenvektor der Hyperebene muss die Klassen *maximal* trennen

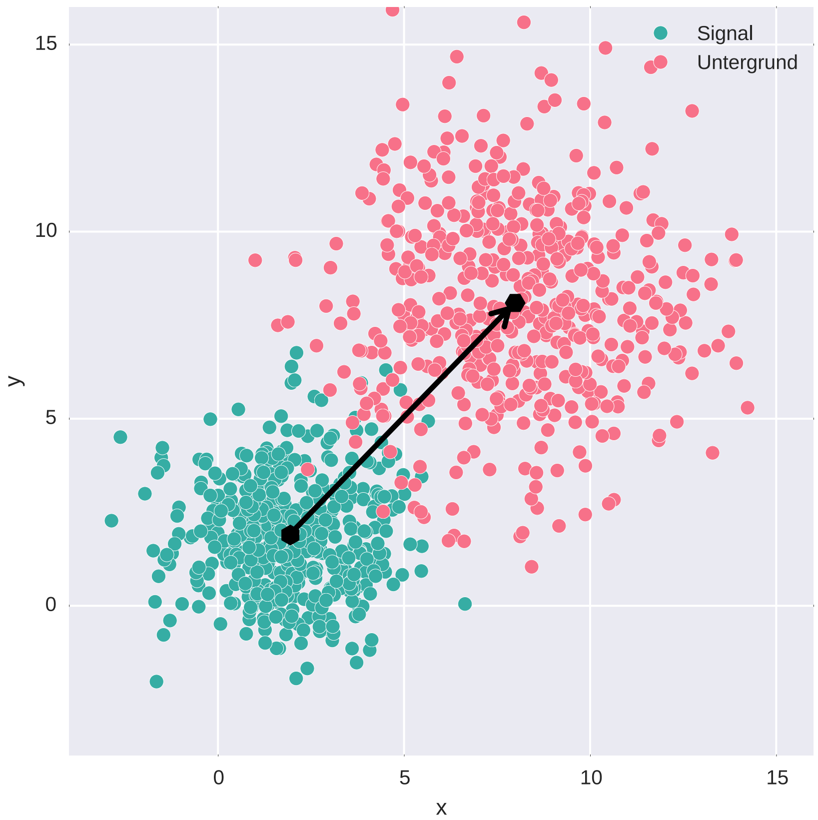
Lineare Fisher Diskriminanzanalyse

- Um eine gute Projektion $\vec{\lambda}$ ($\vec{x}' = \vec{\lambda}^T \vec{x}$) zu finden, muss ein Maß für die Trennbarkeit der Klassen definiert werden
  - Erste (naive) Idee:
    Abstand der Mittelwerte der Klassen auf der Projektionsachse

$$D_{\mathrm{naiv}}(\vec{\lambda}) = |\vec{\mu}_1' - \vec{\mu}_2'| = \left|\vec{\lambda}^T \left(\vec{\mu}_1 - \vec{\mu}_2\right)\right|$$


Signal
Untergrund
15
10
5
0
y
0
5
10
15
x


Signal
Untergrund
15
10
5
0
y
0
5
10
15
x


Signal
Untergrund
15
10
5
0
y
0
5
10
15
x

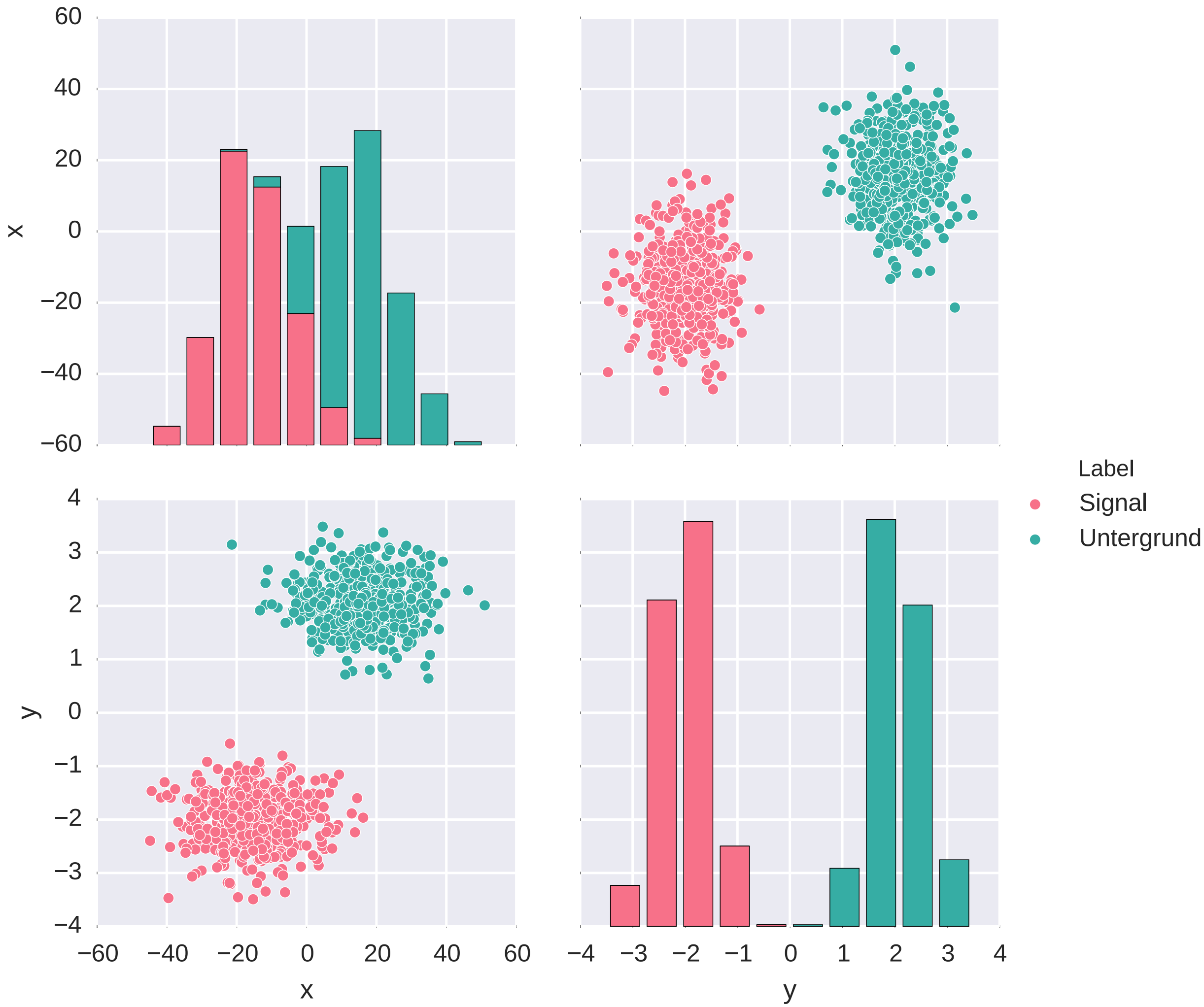

8
6
4
2
0
−2
−4
−6
−8
−10
x
20
15
10
5
0
−5
y
−10 −8 −6 −4 −2 0 2 4 6 8
x
−5 0 5 10 15 20
y
Label
Signal
Untergrund

Lineare Fisher Diskriminanzanalyse

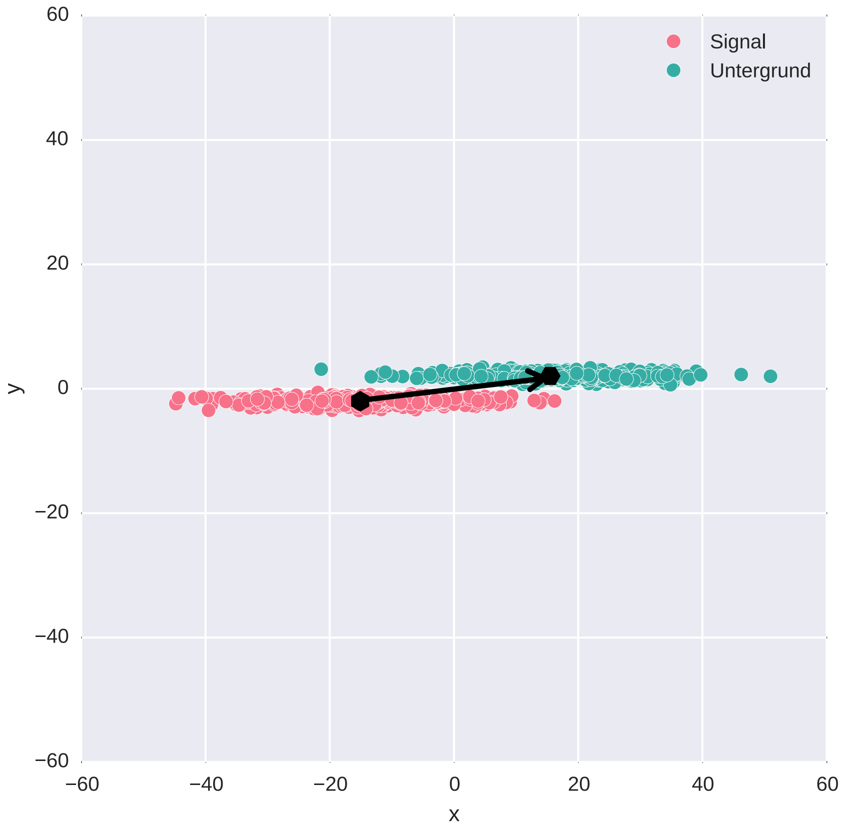
- Um eine gute Projektion $\vec{\lambda}$ ($\vec{x}' = \vec{\lambda}^T \vec{x}$) zu finden, muss ein Maß für die Trennbarkeit der Klassen definiert werden
  - Erste (naive) Idee:
    Abstand der Mittelwerte der Klassen auf der Projektionsachse

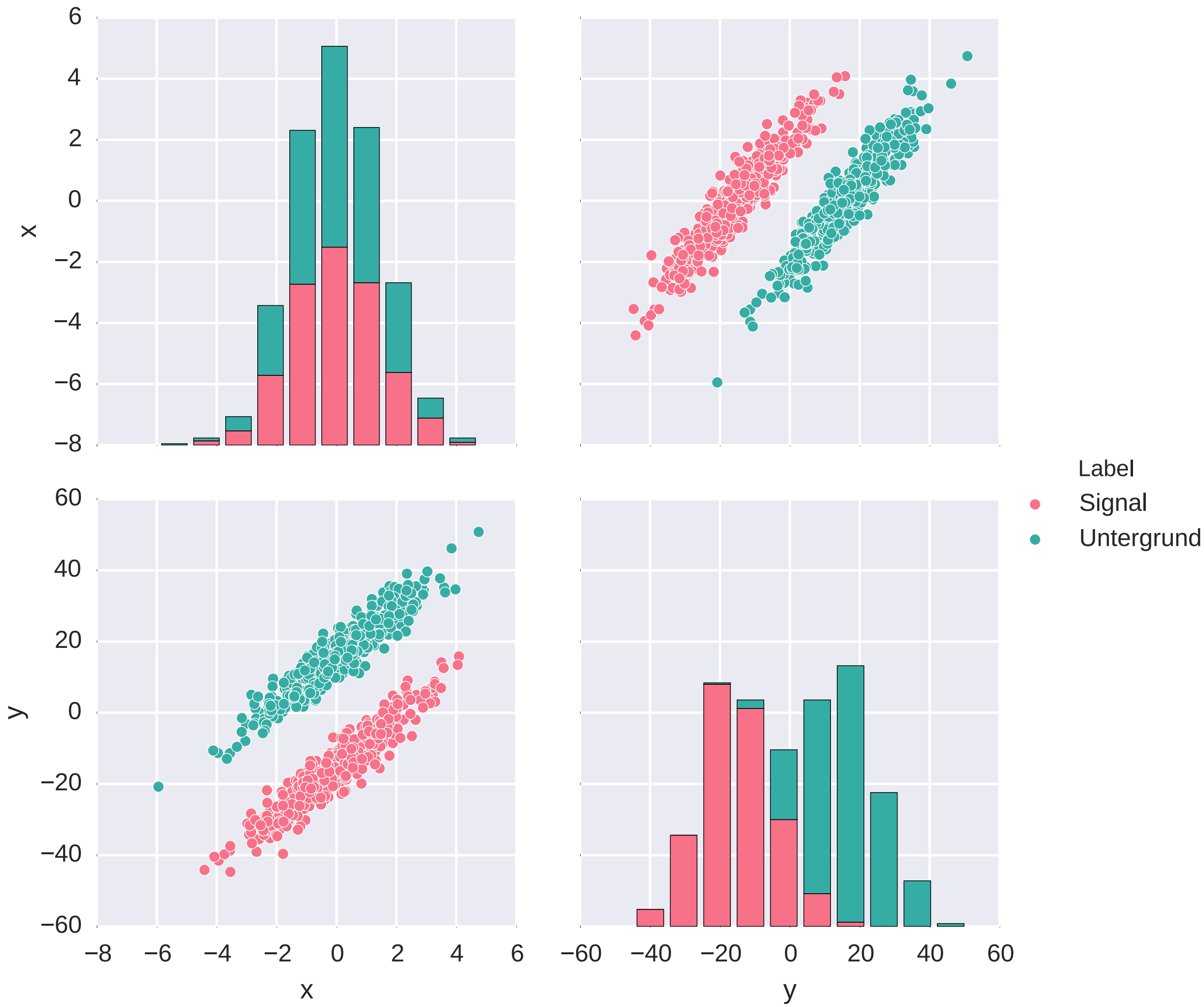
$$D_{\mathrm{naiv}}(\vec{\lambda}) = |\vec{\mu}_1' - \vec{\mu}_2'| = \left|\vec{\lambda}^T (\vec{\mu}_1 - \vec{\mu}_2)\right|$$

Problem: Varianz innerhalb der Klassen wird nicht berücksichtig!



Data-Mining – Teil 1


Signal
Untergrund
60
40
20
0
−20
−40
−60
y
−60
−40
−20
0
20
40
60
x



Data-Mining – Teil 1

Lineare Fisher Diskriminanzanalyse

- Um eine gute Projektion $\vec{\lambda}$ ($\vec{x}' = \vec{\lambda}^T \vec{x}$) zu finden, muss ein Maß für die Trennbarkeit der Klassen definiert werden
  - Erste (naive) Idee:
    Abstand der Mittelwerte der Klassen auf der Projektionsachse

$$D_{\text{naiv}}(\vec{\lambda}) = |\vec{\mu}_1' - \vec{\mu}_2'| = \left| \vec{\lambda}^T \left(\vec{\mu}_1 - \vec{\mu}_2 \right) \right|$$

Problem: Varianz innerhalb der Klassen wird nicht berücksichtig!

  - Idee nach Fisher:
    Quadrat des Abstandes der Mittelwerte der Klassen auf der Projektionsachse, normalisiert mit der Streuung der Klassen

$$D(\vec{\lambda}) = \frac{|\vec{\mu}_1' - \vec{\mu}_2'|^2}{s_1'^2 + s_2'^2}$$

Lineare Fisher Diskriminanzanalyse

- Optimale Trennung zweier Klassen mit je n Observablen durch eine (n-1)-dimensionale Hyperebene
- Gesucht wir die Projektion $\vec{\lambda}$ die $D(\vec{\lambda})$ maximiert
    1. Berechnung der n-dimensionalen Mittelwertvektoren

Lineare Fisher Diskriminanzanalyse

1. Berechnung der n-dimensionalen Mittelwertvektoren
   - Allgemein

$$\vec{\mu}_j = \begin{pmatrix} \bar{x}_{j,1} \\ \ldots \\ \bar{x}_{j,n} \end{pmatrix} = \frac{1}{N_j} \begin{pmatrix} \sum x_{j,1,i} \\ \ldots \\ \sum x_{j,n,i} \end{pmatrix}$$

   - Beispiel

$$\vec{\mu}_1 = \begin{pmatrix} \bar{x}_1 \\ \bar{y}_1 \end{pmatrix} = \frac{1}{N_1} \begin{pmatrix} \sum x_{1,i} \\ \sum y_{1,i} \end{pmatrix}$$

$$\vec{\mu}_2 = \begin{pmatrix} \bar{x}_2 \\ \bar{y}_2 \end{pmatrix} = \frac{1}{N_2} \begin{pmatrix} \sum x_{2,i} \\ \sum y_{2,i} \end{pmatrix}$$

Lineare Fisher Diskriminanzanalyse

- Optimale Trennung zweier Klassen mit n Observablen durch eine (n-1)-dimensionale Hyperebene
- Gesucht wir die Projektion $\vec{\lambda}$ die $D(\vec{\lambda})$ maximiert
  1. Berechnung der n-dimensionalen Mittelwertvektoren
  2. Berechnung der Streumatrizen

Lineare Fisher Diskriminanzanalyse

2. Berechnung der Streumatrizen S_W und S_B

- Streuung innerhalb der Klassen ("Within-class scatter matrix")

Gesamtstreuung: $$S_W = \sum_j^{N_{\text{Klassen}}} S_j$$

Streuung der Klasse j: $$S_j = \sum_i^{n_j} (\vec{x}_i - \vec{\mu}_j)(\vec{x}_i - \vec{\mu}_j)^T$$

- Mit dieser Matrix wird $s_1'^2 + s_2'^2 = \vec{\lambda}^T S_W \vec{\lambda}$, da:

$$s_j'^2 = \sum (\vec{x}' - \vec{\mu}')^2 = \sum (\vec{\lambda}^T \vec{x} - \vec{\lambda}^T \vec{\mu})^2 = \sum (\vec{\lambda}^T (\vec{x} - \vec{\mu}))^2$$
$$= \sum (\vec{\lambda}^T (\vec{x} - \vec{\mu}))(\vec{\lambda}^T (\vec{x} - \vec{\mu}))^T = \sum \vec{\lambda}^T (\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^T \vec{\lambda} = \vec{\lambda}^T S_j \vec{\lambda}$$

Lineare Fisher Diskriminanzanalyse

2. Berechnung der Streumatrizen S_W und S_B

- Streuung zwischen den Klassen ("Between-class scatter matrix")

$$S_B = (\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T$$

- Mit dieser Matrix wird $|\vec{\mu}'_1 - \vec{\mu}'_2|^2 = \vec{\lambda}^T S_B \vec{\lambda}$, da:

$$\begin{aligned} |\vec{\mu}'_1 - \vec{\mu}'_2|^2 &= (\vec{\lambda}^T \vec{\mu}_1 - \vec{\lambda}^T \vec{\mu}_2)^2 \\ &= \vec{\lambda}^T (\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T \vec{\lambda} \\ &= \vec{\lambda}^T S_B \vec{\lambda} \end{aligned}$$

Lineare Fisher Diskriminanzanalyse

2. Berechnung der Streumatrizen S_W und S_B

- Mit den Matrizen S_W und S_B gilt:

$$D(\vec{\lambda}) = \frac{|\vec{\mu}_1' - \vec{\mu}_2'|^2}{s_1'^2 + s_2'^2} = \frac{\vec{\lambda}^T S_B \vec{\lambda}}{\vec{\lambda}^T S_W \vec{\lambda}}$$

- Dieser Ausdruck soll nun maximiert werden

$$\vec{\lambda}^* = \arg\max \left[\frac{\vec{\lambda}^T S_B \vec{\lambda}}{\vec{\lambda}^T S_W \vec{\lambda}} \right]$$

Lineare Fisher Diskriminanzanalyse

- Optimale Trennung zweier Klassen mit n Observablen durch eine (n-1)-dimensionale Hyperebene
- Gesucht wir die Projektion $\vec{\lambda}$ die $D(\vec{\lambda})$ maximiert
  1. Berechnung der n-dimensionalen Mittelwertvektoren
  2. Berechnung der Streumatrizen
  3. Projektion $\vec{\lambda}^*$ berechnen

Lineare Fisher Diskriminanzanalyse

3. Projektion $\vec{\lambda}^{*}$ berechnen

- Zu zeigen: $\vec{\lambda}^{*} = \arg\max \left[\frac{\vec{\lambda}^T S_B \vec{\lambda}}{\vec{\lambda}^T S_W \vec{\lambda}} \right] = S_W^{-1} (\mu_1 - \mu_2)$

  Ableitung von $D(\vec{\lambda})$ und mit 0 gleichsetzen:

$$\frac{\mathrm{d}}{\mathrm{d}\vec{\lambda}} \left[D(\vec{\lambda}) \right] = \frac{\mathrm{d}}{\mathrm{d}\vec{\lambda}} \left[\frac{\vec{\lambda}^T S_B \vec{\lambda}}{\vec{\lambda}^T S_W \vec{\lambda}} \right] = 0$$

$$\Leftrightarrow \left[\vec{\lambda}^T S_W \vec{\lambda} \right] \frac{\mathrm{d} \left[\vec{\lambda}^T S_B \vec{\lambda} \right]}{\mathrm{d}\vec{\lambda}} - \left[\vec{\lambda}^T S_B \vec{\lambda} \right] \frac{\mathrm{d} \left[\vec{\lambda}^T S_W \vec{\lambda} \right]}{\mathrm{d}\vec{\lambda}} = 0$$

$$\Leftrightarrow \left[\vec{\lambda}^T S_W \vec{\lambda} \right] 2 S_B \vec{\lambda} - \left[\vec{\lambda}^T S_B \vec{\lambda} \right] 2 S_W \vec{\lambda} = 0$$

Lineare Fisher Diskriminanzanalyse

3. Projektion $\vec{\lambda}^*$ berechnen (Fortsetzung)

$$\Leftrightarrow \left[\vec{\lambda}^T S_W \vec{\lambda}\right] 2 S_B \vec{\lambda} - \left[\vec{\lambda}^T S_B \vec{\lambda}\right] 2 S_W \vec{\lambda} = 0$$

- Durch $\vec{\lambda}^T S_W \vec{\lambda}$ teilen:

$$\Leftrightarrow \left[\frac{\vec{\lambda}^T S_W \vec{\lambda}}{\vec{\lambda}^T S_W \vec{\lambda}}\right] S_B \vec{\lambda} - \overbrace{\left[\frac{\vec{\lambda}^T S_B \vec{\lambda}}{\vec{\lambda}^T S_W \vec{\lambda}}\right]}^{\text{(Skalar)}} S_W \vec{\lambda} = 0$$

$$\Leftrightarrow S_B \vec{\lambda} - D S_W \vec{\lambda} = 0$$

$$\Leftrightarrow S_W^{-1} S_B \vec{\lambda} = D \vec{\lambda}$$

- Lösung des Eigenwert-Problems $S_W^{-1} S_B \vec{\lambda} = D \vec{\lambda}$

Lineare Fisher Diskriminanzanalyse

3. Projektion $\vec{\lambda}^*$ berechnen (Fortsetzung)

- Lösung des Eigenwert-Problems $S_W^{-1} S_B \vec{\lambda} = D\vec{\lambda}$:

Erinnerung: $S_B = (\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T$ d.h. S_B auf einen beliebigen Vektor $\vec{v}$ angewendet liefert immer einen Vektor in Richtung $(\vec{\mu}_1 - \vec{\mu}_2)$

$$S_B \vec{v} = (\vec{\mu}_1 - \vec{\mu}_2) \underbrace{(\vec{\mu}_1 - \vec{\mu}_2)^T \vec{v}}_{k} = k(\vec{\mu}_1 - \vec{\mu}_2)$$

$$\Rightarrow S_W^{-1} S_B \vec{\lambda} = k S_W^{-1} (\vec{\mu}_1 - \vec{\mu}_2) = D\vec{\lambda}$$

Offensichtlich ist eine mögliche Lösung:

$$\vec{\lambda}^* = \arg\max \left[\frac{\vec{\lambda}^T S_B \vec{\lambda}}{\vec{\lambda}^T S_W \vec{\lambda}} \right] = S_W^{-1} (\vec{\mu}_1 - \vec{\mu}_2)$$

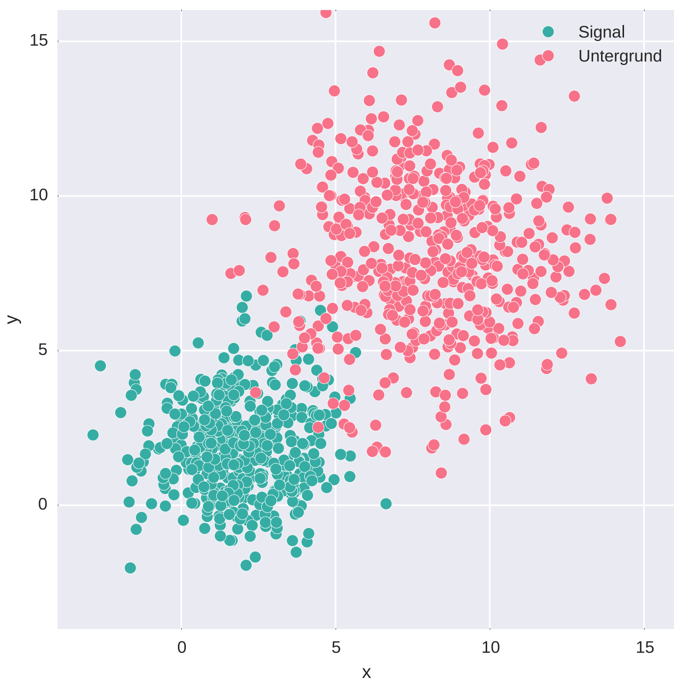
Lineare Fisher Diskriminanzanalyse

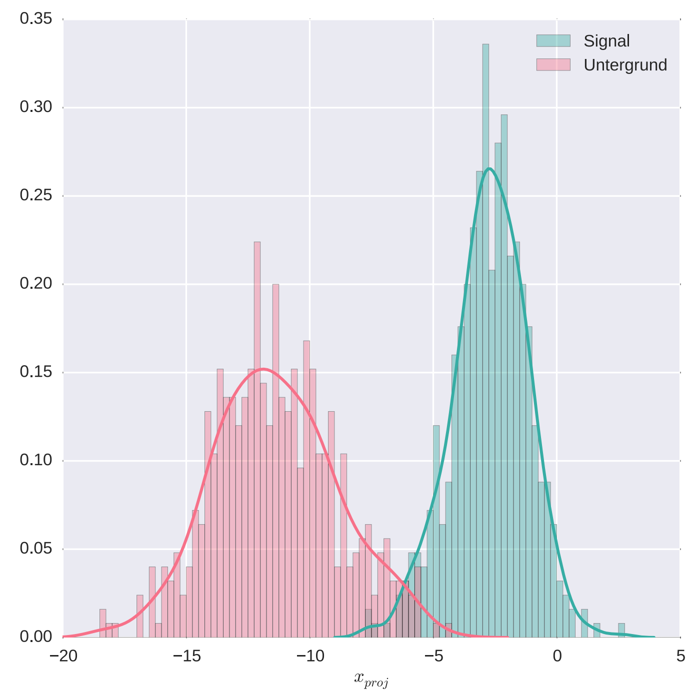
- Optimale Trennung zweier Klassen mit n Observablen durch eine (n-1)-dimensionale Hyperebene
- Gesucht wir die Projektionsachse $\vec{\lambda}$ die $D(\vec{\lambda})$ maximiert
  1. Berechnung der n-dimensionalen Mittelwertvektoren
  2. Berechnung der Streumatrizen
  3. Projektion $\vec{\lambda}^*$ berechnen
  4. Schnitt auf der Projektionsachse festlegen

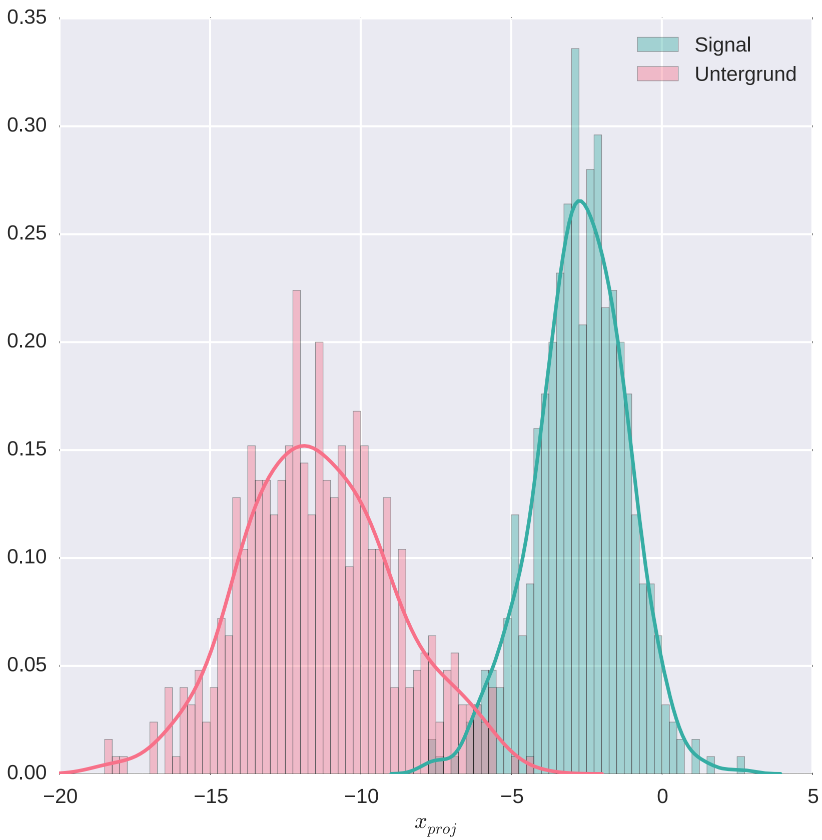
Lineare Fisher Diskriminanzanalyse

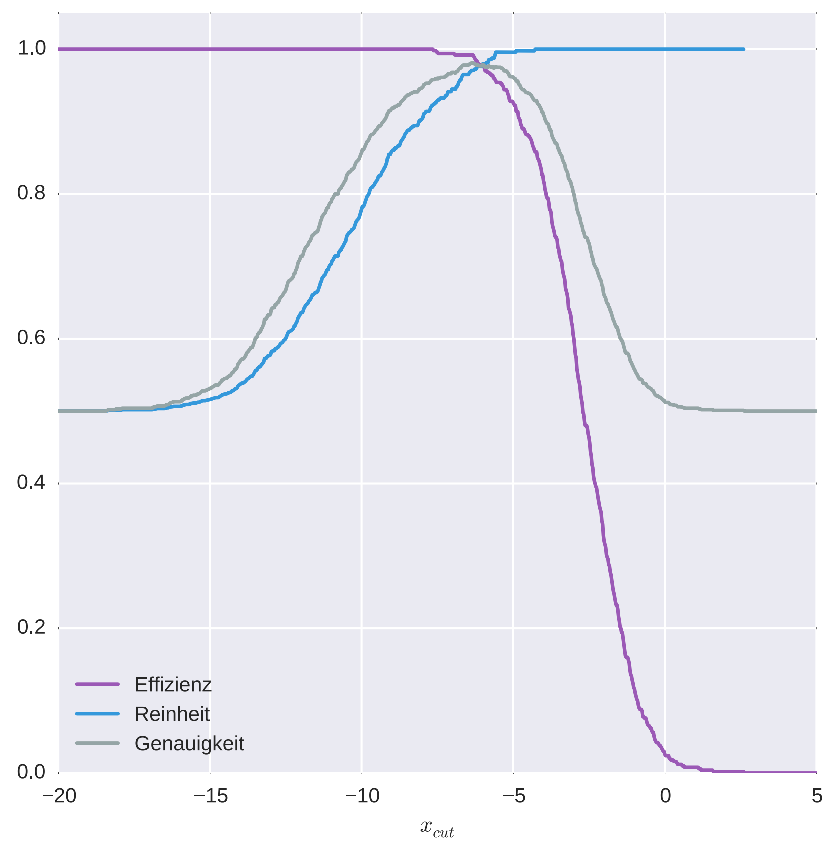
4. Schnitt auf der Projektionsachse festlegen
   - Jeder n-dimensionale Punkt wir dein eine Dimension projeziert
   - Gesucht ist ein Schnitt in auf der Projektionsachse, anhand dessen zwischen beiden Populationen entschieden wird

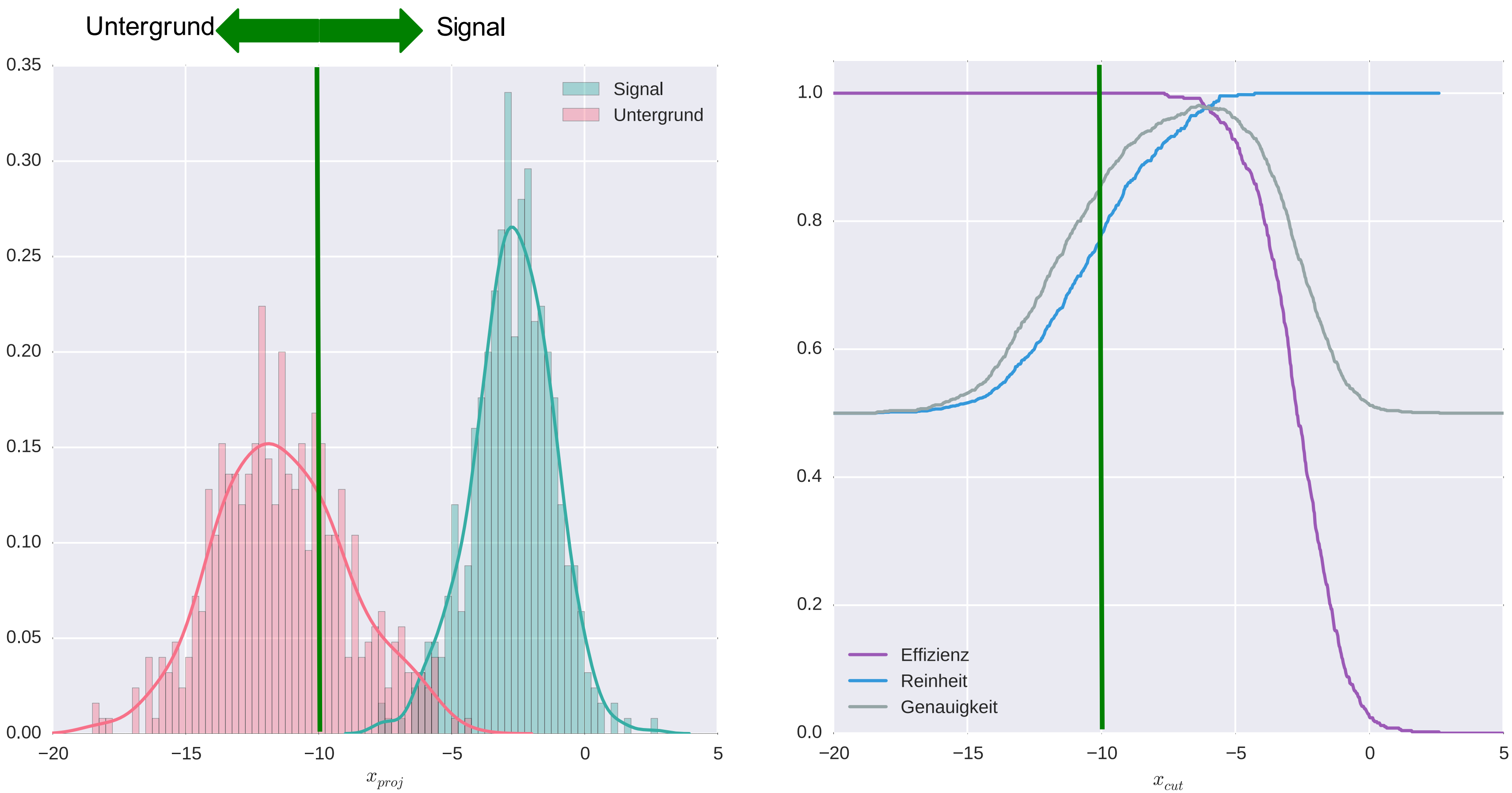
Projektion der Fisher Diskriminanzanalyse



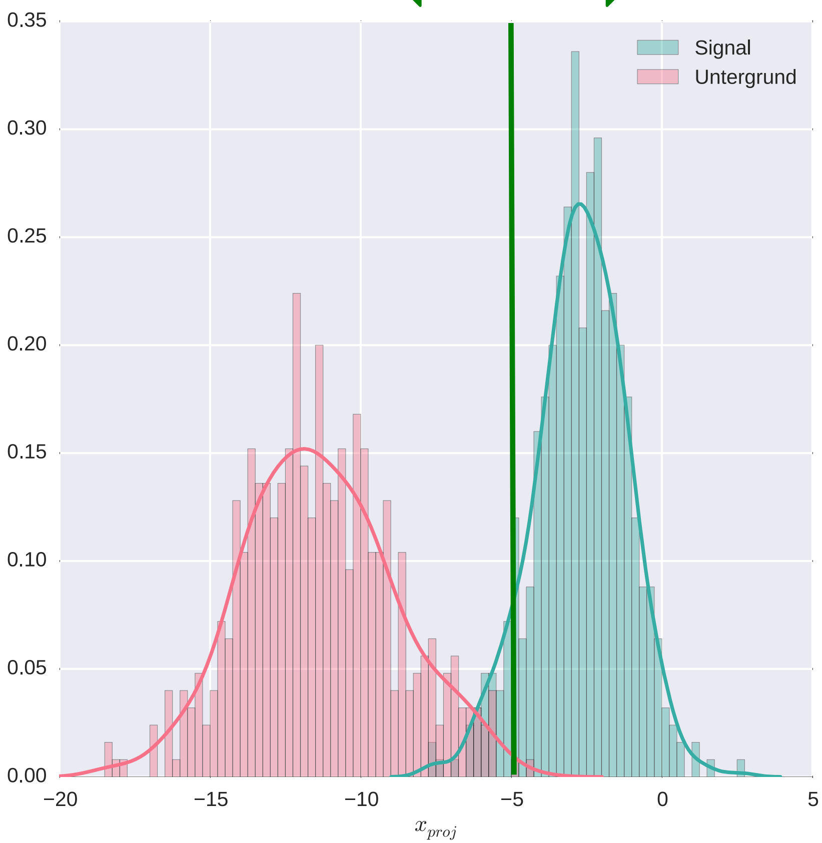


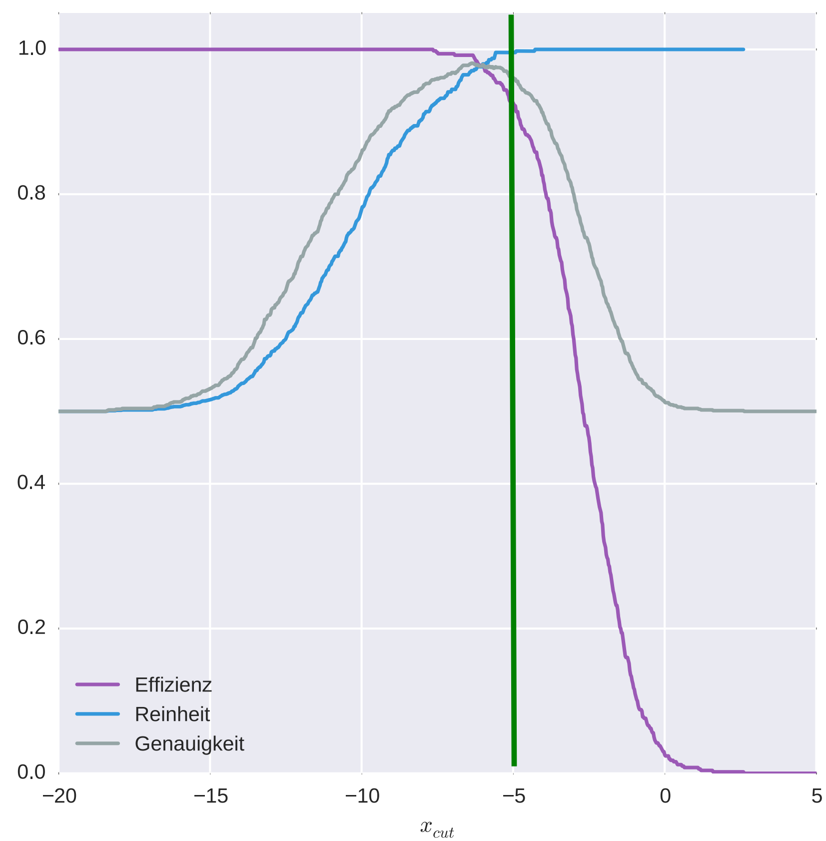

Signal
Untergrund
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−20
−15
−10
−5
0
5
x_{proj}


1.0
0.8
0.6
0.4
0.2
0.0
Effizienz
Reinheit
Genauigkeit
−20
−15
−10
−5
0
5
x_{cut}


Untergrund
Signal
Signal
Untergrund
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−20
−15
−10
−5
0
5
x_{proj}
1.0
0.8
0.6
0.4
0.2
0.0
Effizienz
Reinheit
Genauigkeit
x_{cut}

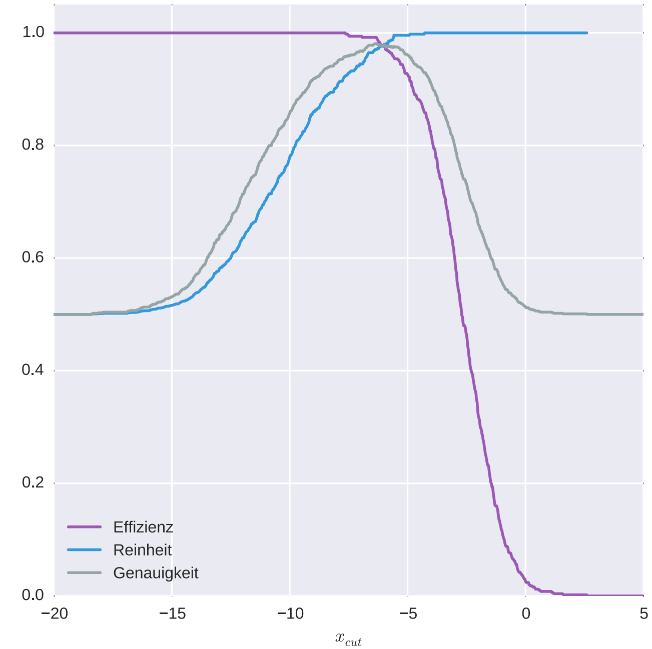
Untergrund ⟵ ⟶ Signal





Lineare Fisher Diskriminanzanalyse

4. Schnitt auf der Projektionsachse festlegen
   - Jeder n-dimensionale Punkt wird auf eine Dimension projeziert
   - Gesucht ist ein Schnitt in auf der Projektionsachse, anhand dessen zwischen beiden Populationen entschieden wird



   - Allgemein kann kein bester Schnitt angegeben werden
   - Muss für jedes konkrete Problem motiviert werden
     - Trade-Off zwischen Effizienz und Reinheit

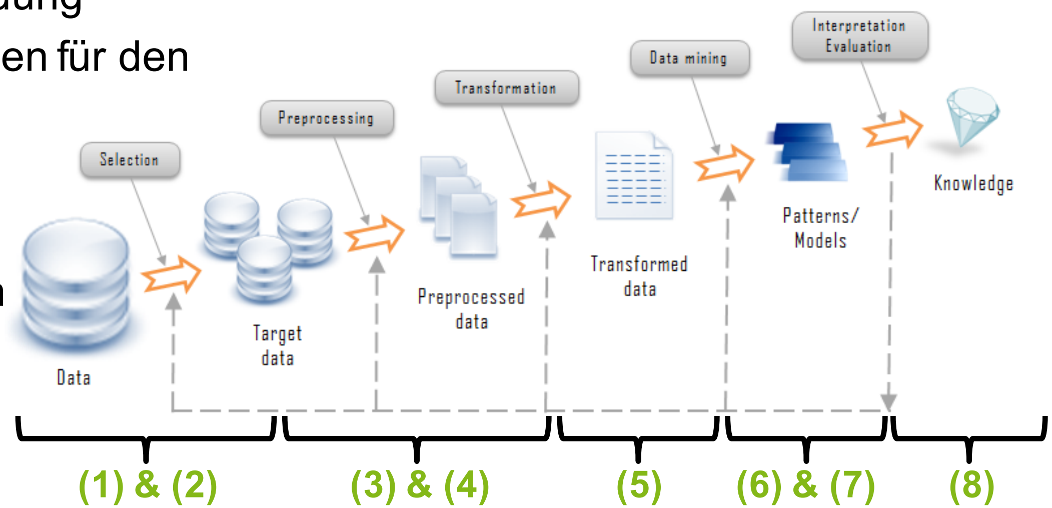
Data-Mining

- Ursprünglich ein Schritt von sogenannten „**K**nowledge **D**iscovery in **D**atabases“-Prozessen; mittlerweile meist gleichbedeutend zu KDD
- Data-Mining meint häufig die Anwendung von Algorithmen des maschinellen Lernens
  - „[Machine learning is a] field of study that gives computers the ability to learn without being explicitly programmed.“ (Arthus Smith, 1959)
  - „A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” (Tom M. Mitchell, 1997)
- Dieser Teil der Vorlesung soll einen **Einblick** in das große Feld des Data Minings geben

Data Mining

- Die Teilschritte eines Data Mining-Prozesses sind (Fayyad et al.):

(1) Definition der Ziele der Wissensfindung
(2) Bereitstellung von Hintergrundwissen für den jeweiligen Fachbereich
(3) Datenauswahl
(4) Datenbereinigung
(5) Datenreduktion und -transformation
(6) Auswahl eines Modells
(7) Data-Mining
(8) Interpretation



- **KDD-/Data-Mining-Prozesse sind iterativ und interaktiv**
- In der Praxis sind manche Schritte nicht voneinander zu trennen und die Reihenfolge kann leicht unterschiedlich sein

Kleines Data Mining Wörterbuch

- Feature: Attribut, Observable, Messgröße, Merkmal
- Label: Zielgröße → labeled/unlabeled: Wert der Zielgröße bekannt/unbekannt
- Klassen: Werte einer diskreten Zielgröße (häufig wird Label und Klasse äquivalent genutzt)
- Überwachtes Lernen: „Supervised learning is the machine learning task of inferring a function from labeled training data.“ (Foundations of Machine Learning, 2012)
- Unüberwachtes Lernen: Erkennen von Strukturen unabhänig von Zielgrößen, Optimierungskriterien, Feedback Signalen oder sonstiger Informationen, die über die tatsächlichen Daten hinausgehen

- Warnung: Selten sind die Bedeutungen der Begriffe allgemeingültig definiert