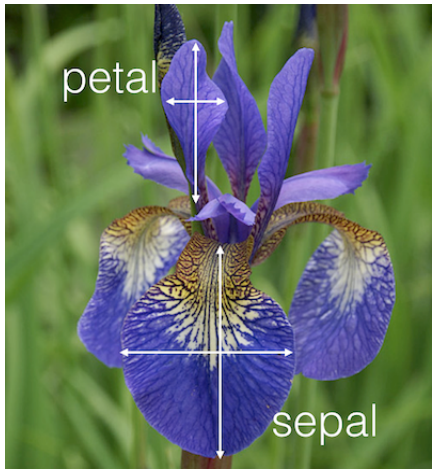
Moderne Methoden der Datenanalyse

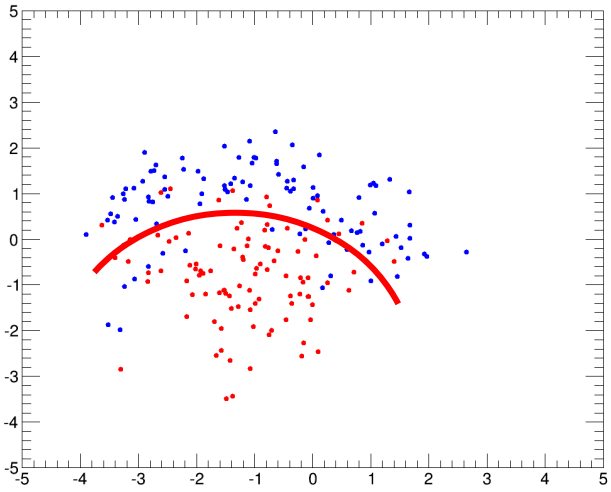
Zusammenfassung
Klassifikation (1)

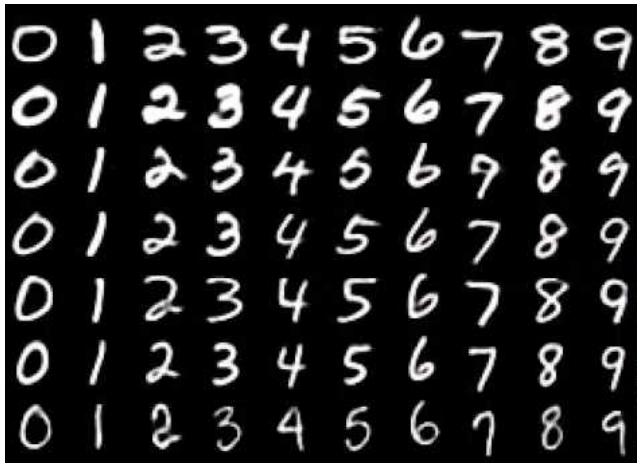
Andreas B.Meyer

27. Juni 2017

Fakultät für Physik
Institut für Experimentelle Kernphysik - IEKP





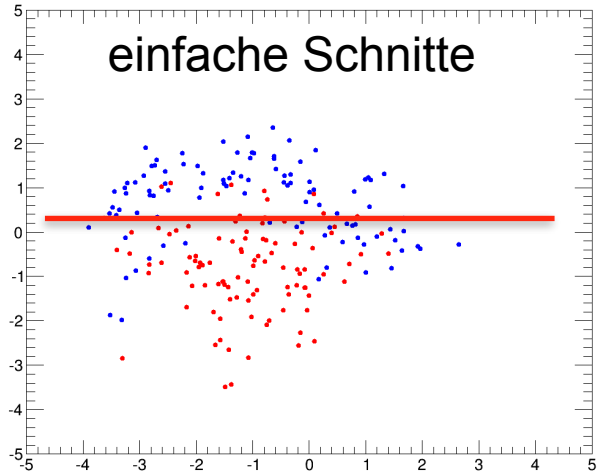


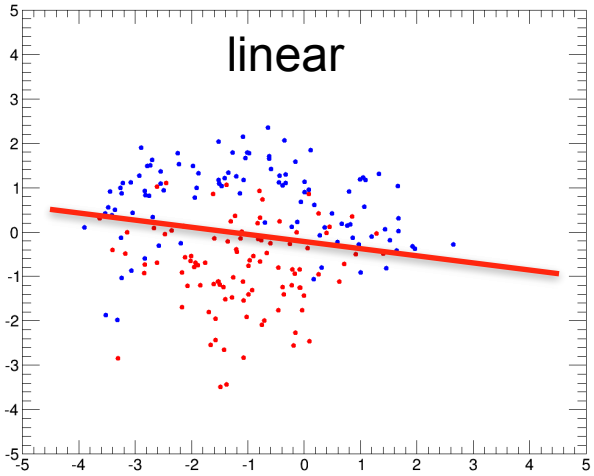
Maschinelles Lernen

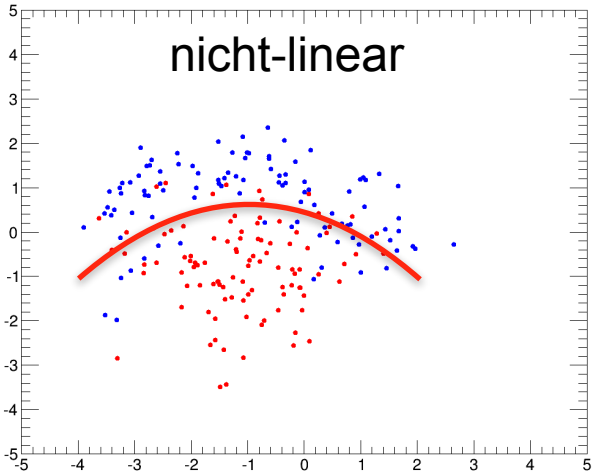
- Klassifikator lernt aus Daten, optimiert selbständig und liefert Schätzung der PDF

- Überwachtes Lernen (engl. “supervised learning”)
  - Training des Klassifikators anhand von vorklassifizierten Daten
  - Differenz zwischen Wahrheit und Ergebnis, beschrieben durch vorgegebene Ziel- oder Kostenfunktion E(||t_i-t_{itrue}||), wird sukzessive verbessert.
- Unüberwachtes Lernen:
  - Selbständiges Erkennen von Signalen, Mustern oder Clustern, Bestimmung des Klassifikators
- Bestärkendes Lernen (engl. “reinforcement learning”):
  - Lernen anhand von Belohnung (oder Strafe), aber ohne Vorgabe von Klassen

Klassifikatoren (Prüfgrößen)







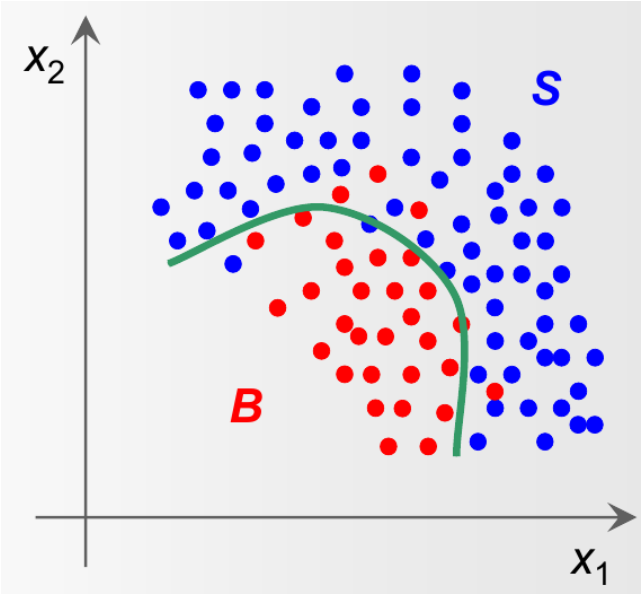
- Lineare Verfahren: analytisch behandelbar (diese Vorlesung)

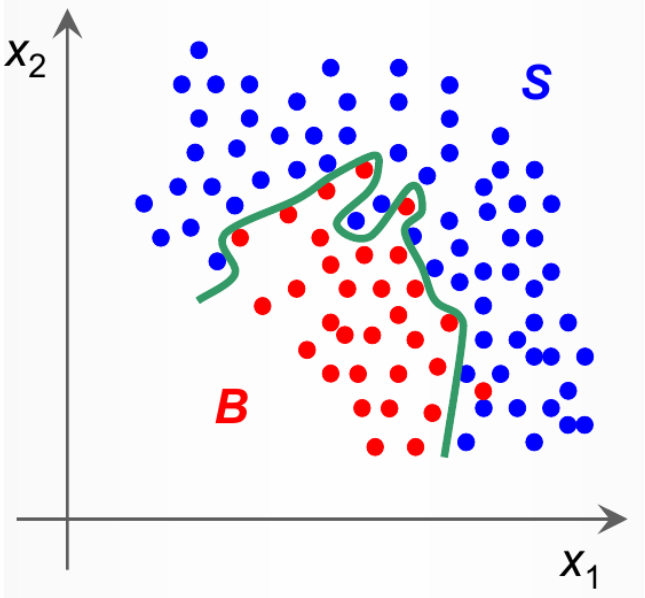
- Nicht-lineare Verfahren analytisch i.a. nicht möglich (folgende Vorlesungen)
  - Automatisiere Algorithmus zur Ermittlung optimaler Trennung
    → Maschinelles Lernen
  - Supervised Learning: Trainieren auf vorklassifizierter Stichprobe
    (aus Simulation oder Erfahrung)
  - Häufiges Problem: PDF sind häufig nur unzureichend bekannt und simulierbar

Wähle Verfahren unter Berücksichtigung von Aufwand und Nutzen

Supervised Learning (Training)

- Bestimmung der optimalen Prüfgröße (Parameter) anhand von Trainingsdaten, in denen Signal und Untergrund bekannt (vorklassifiziert) sind.

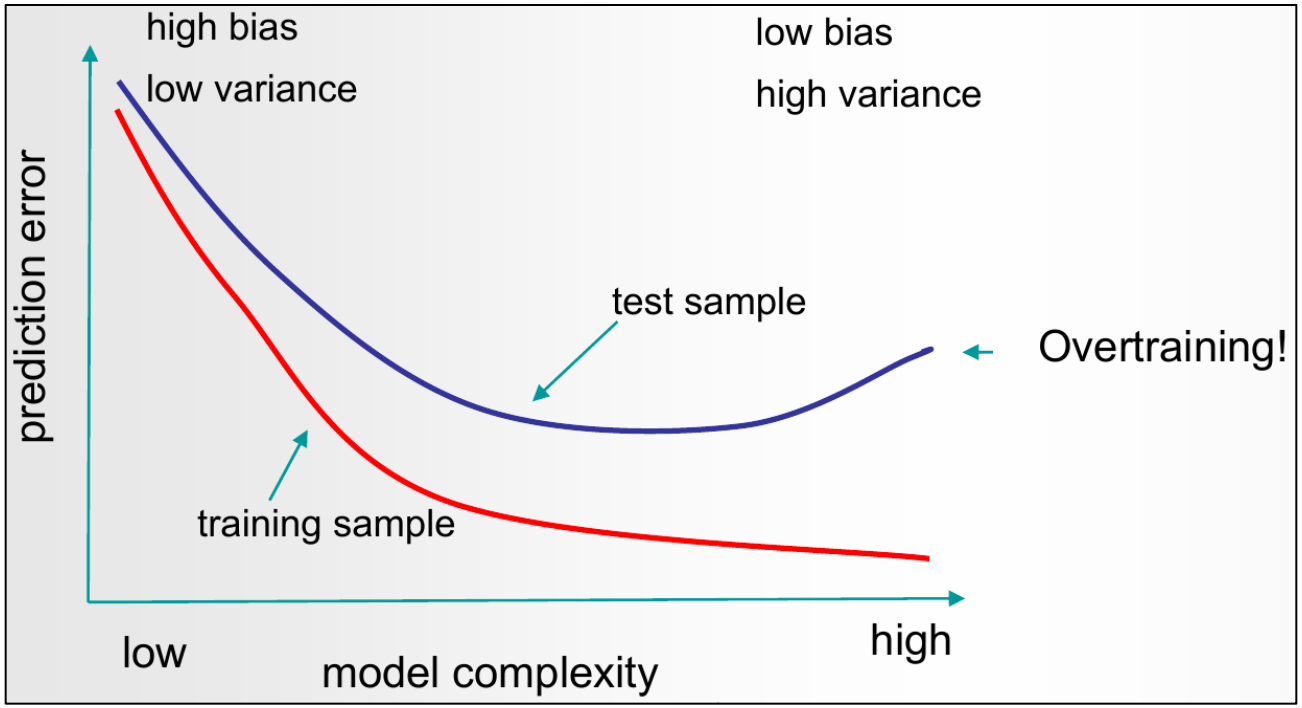




- Gefahr: Algorithmus erlernt statistische Fluktuationen (Details) der Stichprobe (Trainingsdaten) und nicht das allgemeine Konzept (Verlust der Verallgemeinerbarkeit)
- Besonders anfällig, wenn viele Dimensionen oder Parameter → Fluch der Dimensionen → optimale Wahl der Freiheitsgrade zur Reduktion der Dimensionen

Supervised Learning (Testing)

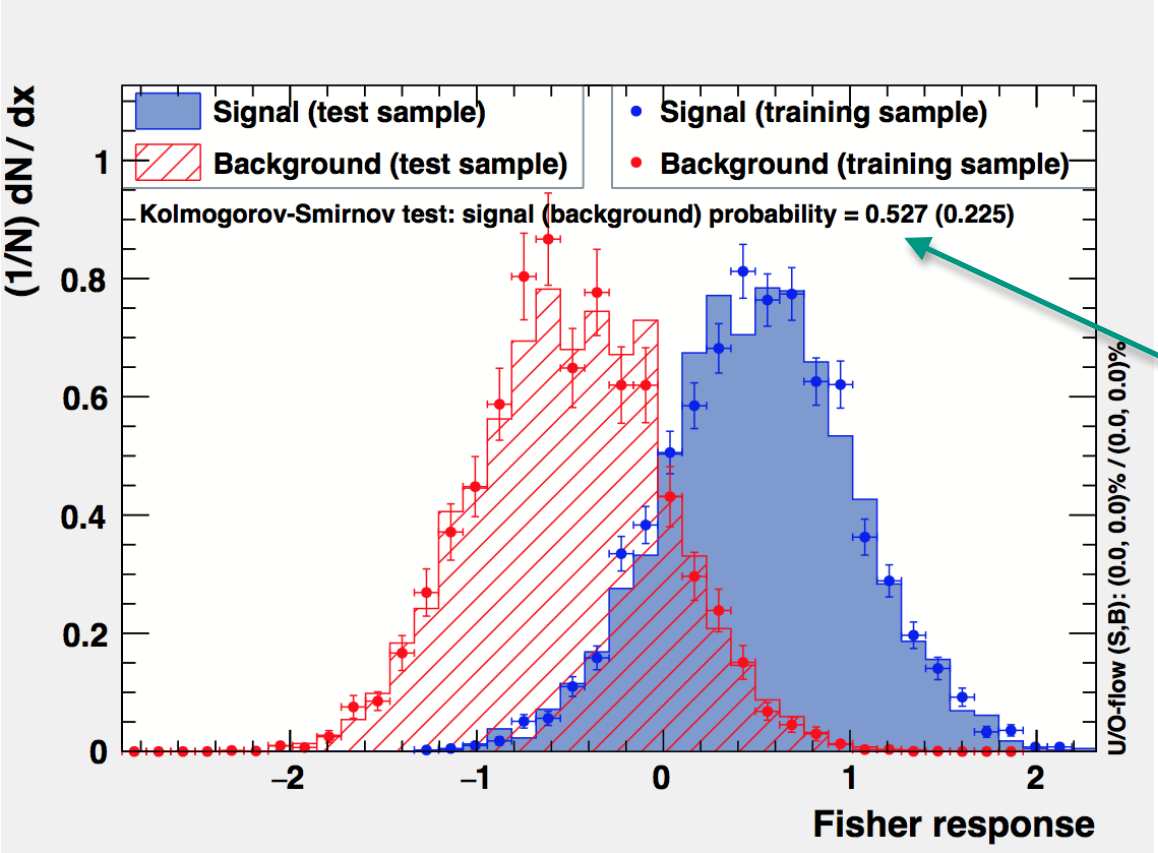
- Test des durch Training ermittelten Algorithmus mit einem statistisch unabhängigen Datensatz



- Niedrigere Komplexität (z.B wenige Parameter oder Trainingszyklen, z.B. lineare Methoden) → ggf. schlechtere Trennung (Verzerrung), kleinere Unterschiede zwischen statistisch unabhängigen Datensätzen (Varianz)
- Höhere Komplexität → kleinere Verzerrung. Übertraining → größere Varianz

Supervised Learning (Testing)

- Vergleich der Verteilungen von Training und Testsample
- Goodness-of-fit: Kolmogorov-Smirnov (Erinnerung (Vorl. 6): KS-Test verwendet maximale Differenz der kumulativen Verteilungsfunktionen)



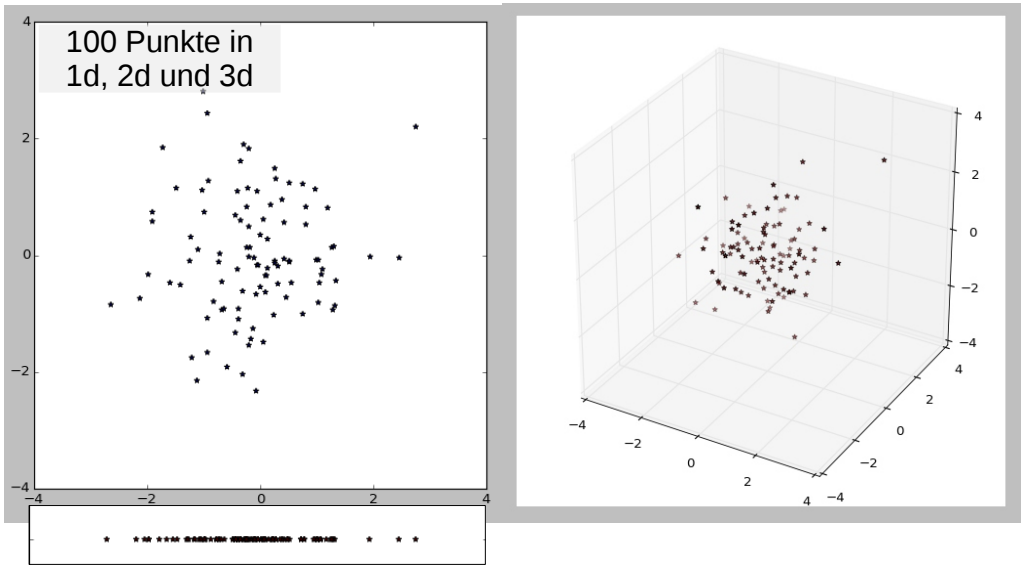
p-Werte

Fluch der Dimensionen (“Curse of Dimensionality”)

- In hochdimensionalen Merkmalsräumen ist die PDF in der Regel nicht genau bekannt:

  d-dimensionales Histogramm des (mit N Einträgen endlich großen) Datensatzes ist bei fester Bin-Anzahl n_b pro Dimension praktisch leer

$$n_i = N/n_{\text{bins}} = \frac{N}{{n_b}^d} \to 0, \text{ wenn } d \to \infty$$



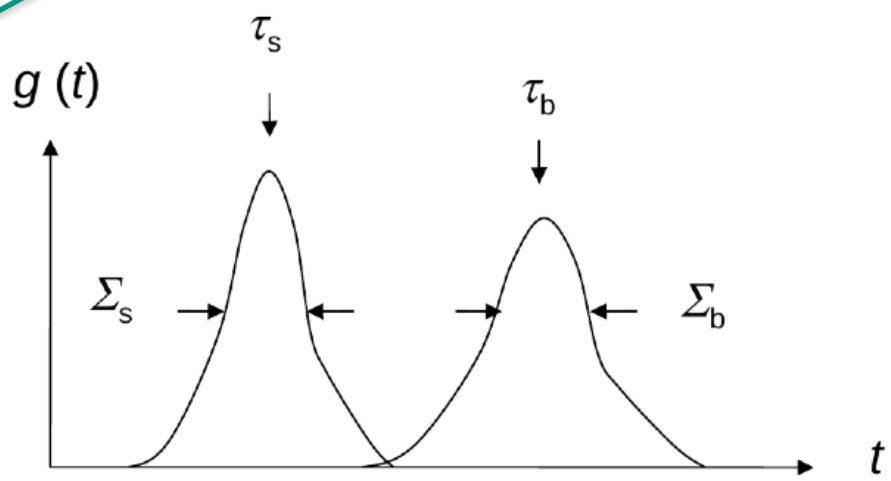
Trainingsdaten liegen bei vielen Dimensionen sehr zerstreut
Würden N^d Trainingsereignisse benötigen, um n_i konstant zu halten

Fisher Diskriminante

- Ansatz: Lineare Prüfgröße $$t(\vec{x}) = \sum_{i=1}^{n} a_i x_i = \vec{a}^T \vec{x}$$
- Parameterwahl: Maximale Separation, wenn Differenz zwischen Mittelwerten der PDF groß und Varianz jeder der Verteilungen klein
  - Differenz zwischen Mittelwerten: $|\mu_s - \mu_b|$ groß
  - Streuung um Mittelwerte: $\Sigma_s^2 + \Sigma_b^2$ klein
- Fisher Diskriminante: Maximiere $$J(\vec{a}) = \frac{(\tau_s - \tau_b)^2}{\Sigma_s^2 + \Sigma_b^2}$$

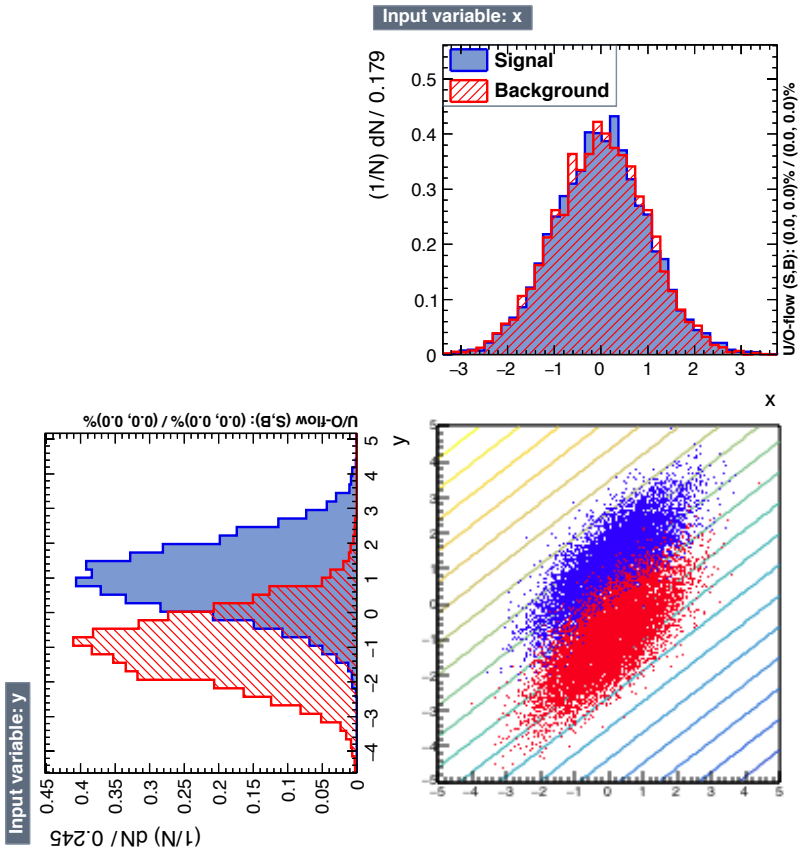
"Kostenfunktion"

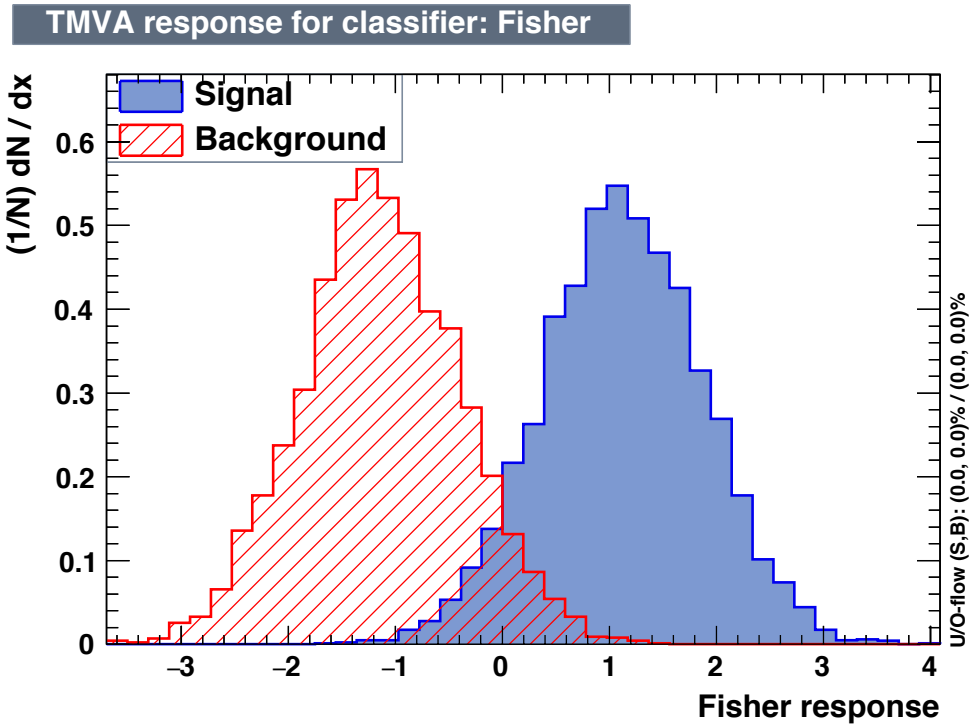
- Bestimme Fisher-Koeffizienten: Suche a, so dass $$\vec{\nabla} J(\vec{a}) = 0$$



Beispiel: Fisher Diskriminante in TMVA

- Je 10000 Signal- und Untergrundereignisse: verschobene Gauß-Funktionen jeweils korreliert zwischen x und y.
- Fisher Diskriminante berücksichtigt Korrelation

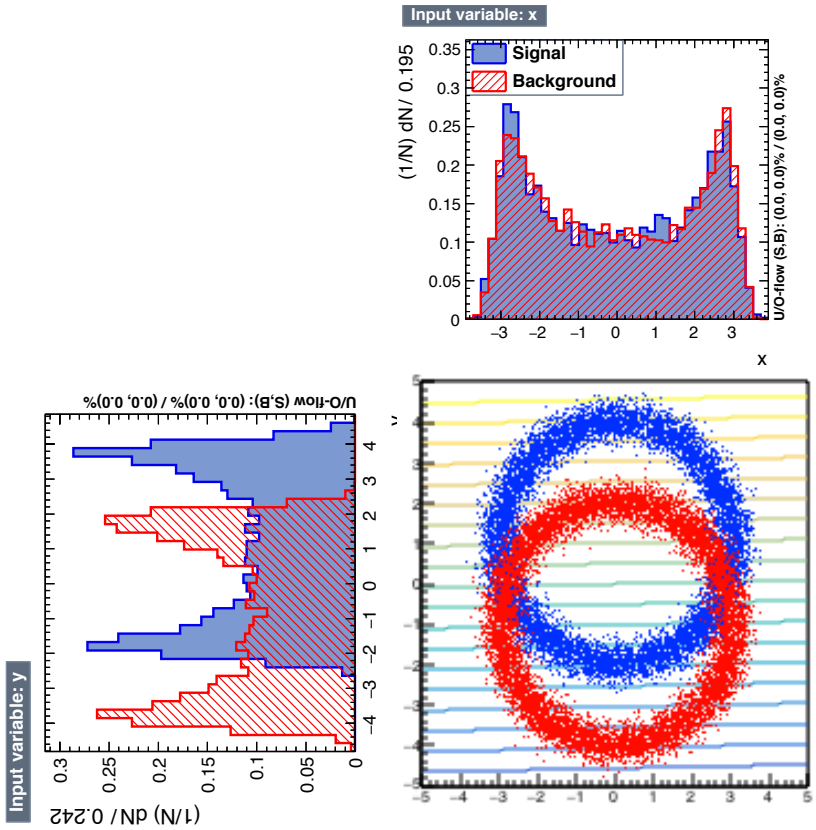


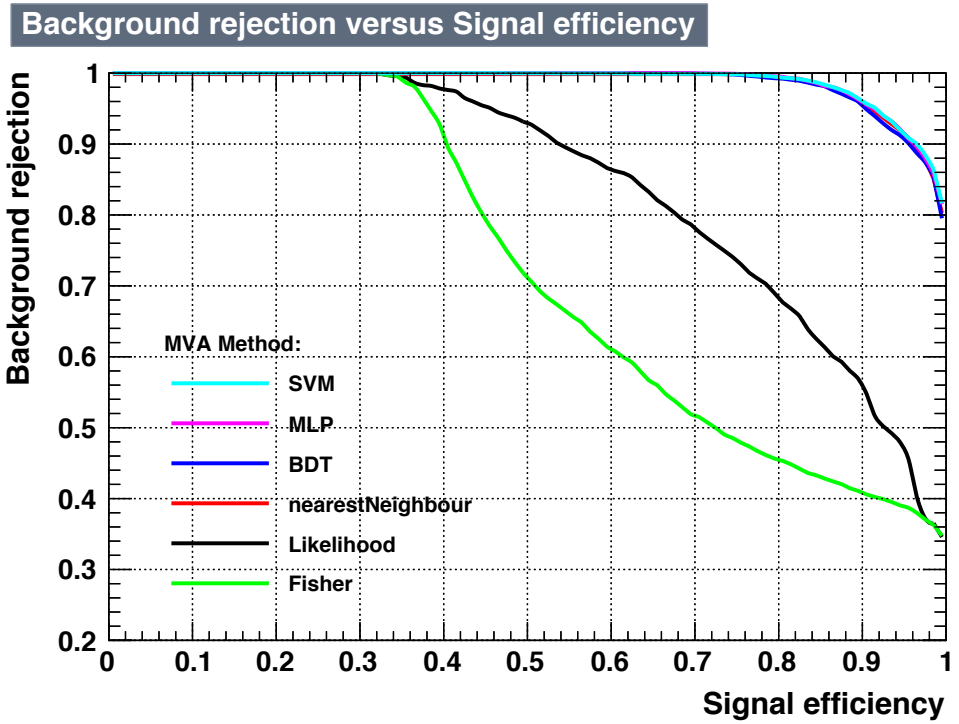


Beispiel: “python TrainEvaluateResponse.py Gaussian”

Beispiel: Fisher Diskriminante in TMVA

- Je 10000 Signal- und Untergrundereignisse: verschobene verschmierte Kreise - nichtlineares Problem
- Fisher-Diskriminante ist nicht-linearen Methoden (überwiegend basierend auf Maschinen-Lernverfahren) deutlich unterlegen.





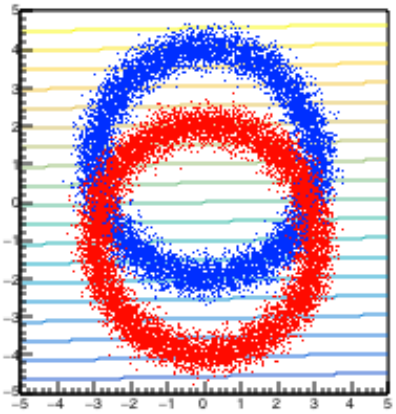
Beispiel: “python TrainEvaluateResponse.py Gaussian”

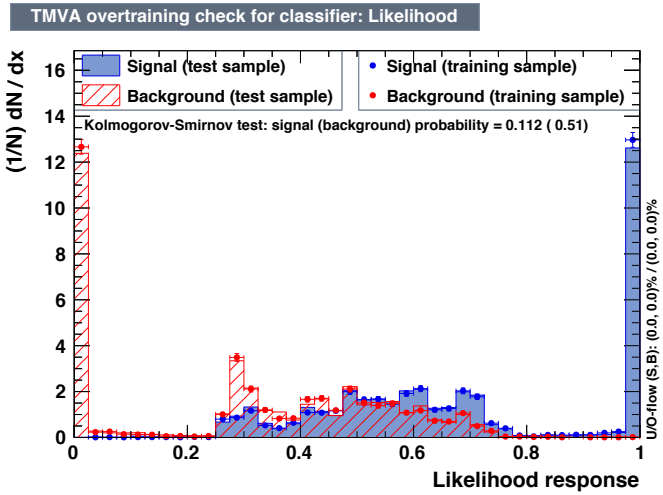
Andere Methoden: Projective Likelihood

- Wenn Korrelationen zwischen den Merkmalen klein (z.B. nach linearer Dekorrelation):
- Verwendung des Produkts der d 1-dimensionalen Randverteilungen der Likelihoods

$$t(\vec{x}) = \frac{\prod_{i=1}^{d} g_i(x_i|H_s)}{\prod_{i=1}^{d} g_i(x_i|H_s) + \prod_{i=1}^{d} g_i(x_i|H_b)}$$

- Funktioniert auch für relativ große Zahl d von Merkmalen

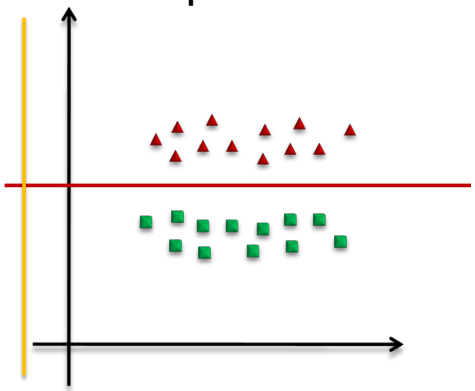




Andere Methoden: Hauptkomponentenanalyse

- Principal Component Analysis (PCA)
- Anders als Fisher: Keine Unterscheidung zwischen Signal- und Untergrundverteilungen (“unsupervised learning”)
- Annahme: Relevante Information liegt in den Komponenten mit großer Varianz
- Lineare Transformation auf Hauptachsen (Achsen mit maximaler Varianz)
- Vernachlässigung der Komponenten mit kleinen Eigenwerten (→ Dimensionsreduktion)

Beispiel: Daten



Fisher LDA:

gute Trennung

PCA:

keine Trennung