

Kapitel 6 - Unüberwachtes Lernen

Annika Liebgott

November 21, 2025

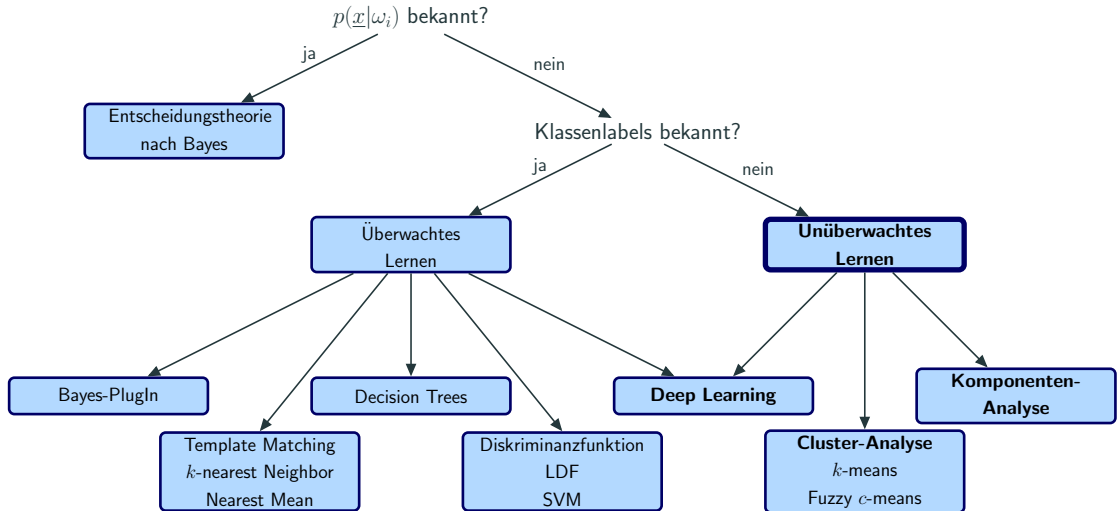
Nachteile von überwachtem Lernen

Im überwachtem Lernen werden direkte Zusammenhänge zwischen Samples \underline{x}_n und ihrer Klassenzugehörigkeit gelernt.

Nachteil: Klassenlabels werden benötigt, um den Zusammenhang herzustellen
⇒ Klassenlabels müssen *allen* Trainingssamples im Vorfeld zugeordnet werden
⇒ Zuordnung muss meist manuell durchgeführt werden → zeitaufwändig
⇒ Häufig ist für Zuordnung Expertenwissen nötig → teuer

Ist das Labeln von Daten nicht möglich, können Methoden des **unüberwachten Lernens** hilfreich sein

Übersicht



6.1 - Cluster-Analyse

Grundidee: Unterteile Daten auf Basis ihrer Lage im Merkmalsraum in Cluster, *ohne Klassenlabels zu nutzen*

Fragestellungen:

- Welche Samples gehören zu welchem Cluster?
- Wie viele Cluster existieren?

Frage: Warum nicht einfach Trial-and-Error?

Antwort: schon mit wenigen Samples extrem aufwändig.

Beispiel: $c = 2$ Cluster $\Rightarrow 2^{N-1} - 1$ Kombinationsmöglichkeiten

$N = 20$: 524.287 Möglichkeiten

$N = 100$: $\approx 6 \cdot 10^{29}$ Möglichkeiten!

Grundidee:

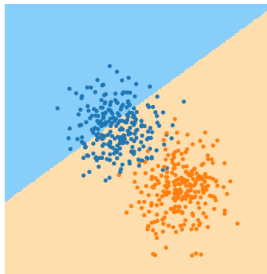
- Iterative Clustering-Methode
- Basierend auf Abständen zwischen Samples \underline{x}_n und Cluster-Mittelpunkten $\underline{\mu}_j$

Vorgehensweise:

1. k initiale Cluster-Mittelpunkte $\underline{\mu}_j$ zufällig wählen
2. Samples \underline{x}_n dem Cluster mit dem nächstgelegenen Mittelpunkt zuordnen
3. Auf Basis der so gebildeten Cluster neue Cluster-Mittelpunkte berechnen
4. Schritte 2 & 3 so lange wiederholen, bis sich die Cluster-Mittelpunkte nicht mehr ändern

Veranschaulichung k -Means

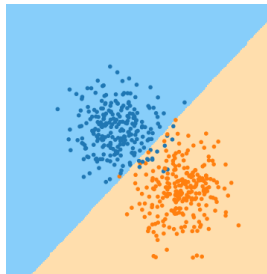
Datenset 1 (Blobs), zufällige Initialisierung



1. Iteration



2. Iteration



3. Iteration



4. Iteration

⇒ Algorithmus konvergiert für das vorliegende 2-Klassen-Problem

Varianten von k -Means

Nachteil von k -Means mit zufälliger Initialisierung:

Algorithmus findet evtl. nicht die richtigen Cluster oder keine optimale Lösung

¹Arthur und Vassilvitskii: "k-means++: the advantages of careful seeding", 2007

Varianten von k -Means

Nachteil von k -Means mit zufälliger Initialisierung:

Algorithmus findet evtl. nicht die richtigen Cluster oder keine optimale Lösung

⇒ **Lösungsmöglichkeit: andere Varianten zur Initialisierung der Cluster**

¹Arthur und Vassilvitskii: "k-means++: the advantages of careful seeding", 2007

Varianten von k -Means

Nachteil von k -Means mit zufälliger Initialisierung:

Algorithmus findet evtl. nicht die richtigen Cluster oder keine optimale Lösung

⇒ **Lösungsmöglichkeit: andere Varianten zur Initialisierung der Cluster**

z.B. k -Means++¹ (default-Initialisierung in scikit-learn):

1. Wähle zunächst zufällig ein Sample als Cluster-Mittelpunkt $\underline{\mu}_i$
2. Berechne Abstand aller Samples zu $\underline{\mu}_j$
3. Wähle ein Sample als weiteren Mittelpunkt $\underline{\mu}_j$, die Auswahlwahrscheinlichkeit wird dabei mit dem Abstand zu $\underline{\mu}_i$ gewichtet, größere Abstände werden bevorzugt
4. Wiederhole 2. & 3., bis k Mittelpunkte bestimmt sind
5. Führe jetzt normalen k -Means-Algorithmus aus

¹Arthur und Vassilvitskii: "k-means++: the advantages of careful seeding", 2007

k -Means: jedes Sample \underline{x}_n gehört zu exakt einem Cluster \Rightarrow harte Entscheidung

Konzept von Fuzzy c -Means:

\underline{x}_n wird mehreren Clustern zugeordnet \Rightarrow weiche Entscheidung

Funktionsprinzip:

- \underline{x}_n gehört zu jeder der c Klassen mit einer Wahrscheinlichkeit P_j
- Abstandsberechnung zwischen \underline{x}_n und $\underline{\mu}_j$ wird mit P_j gewichtet
- Stärke der Gewichtung wird über einen Fuzziness-Parameter kontrolliert
- Meistens wird am Ende eine harte Entscheidung über die maximale Mitgliedswahrscheinlichkeit aller Samples \underline{x}_n getroffen

²J.C. Bezdek: "Pattern recognition with fuzzy objective function algorithms.", Plenum Press, New York 1981.

Clustering-Methoden für eine unbekannte Klassenanzahl

Bei Methoden wie k -Means oder Fuzzy c -Means geht man davon aus, die richtige Anzahl der Cluster zu kennen.

Nachteile:

- in der Realität ist diese Anzahl oft unbekannt
- falsch gewählte Clusterzahl vermindert Performance des Systems

⇒ **Bei unbekannter Clusterzahl: passendere Methoden verwenden**

Beispiele für Methoden für unbekannte Anzahl an Clustern:

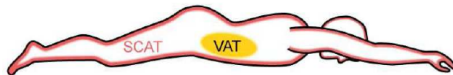
- Mean-Shift Clustering³
- DBSCAN⁴ (**D**ensity-**b**ased **s**pacial **c**lustering of **a**pplications with **n**oise)
- OPTICS⁵ (**O**rdering **p**oints **t**o **i**dentify the **c**lustering **s**tructure)

³Y. Cheng: "Mean Shift, Mode Seeking, and Clustering", 1995

⁴Ester et al: "A density-based algorithm for discovering clusters in large spatial databases with noise"

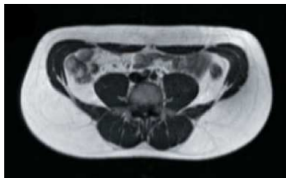
⁵Ankerst et al.: "OPTICS: Ordering Points To Identify the Clustering Structure", 1999

Beispiel 6.1: k -means für Fett-Segmentierung in MRT-Bildern⁶

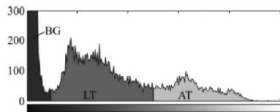


- LT lean tissue (muscle)
- AT adipose tissue (fat)
- SCAT subcutaneous adipose tissue (fat directly under skin)
- VAT visceral adipose tissue (fat inside the body)

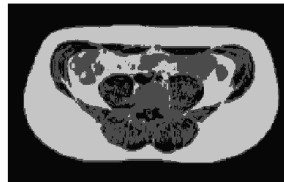
original MRI image



gray value histogram



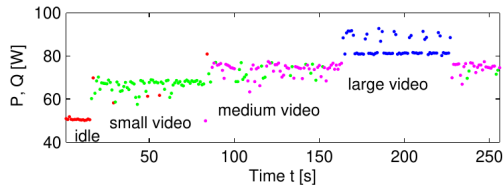
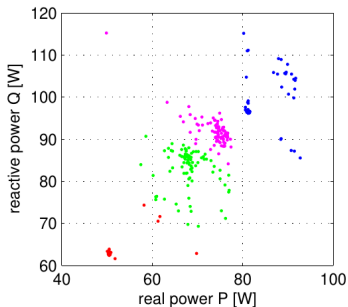
clustering by k -means



⁶Vorlesungsunterlagen "Detection and Pattern Recognition", Prof. B. Yang, ISS, Uni Stuttgart

Beispiel 6.2: Zustandsidentifikation eines PCs⁷

Vereinfachte Aufgabe zur Zustandsüberwachung: Identifikation der vier Zustände “Idle” sowie Abspielen eines Video mit “niedriger Auflösung”, “mittlerer Auflösung” oder “hoher Auflösung”



⁷Vorlesungsunterlagen “Detection and Pattern Recognition”, Prof. B. Yang, ISS, Uni Stuttgart

Vor- und Nachteile von Clustering-Methoden

Vorteile:

- + intuitive Methode
- + keine Labels benötigt
- + geeignet, um mit relativ geringem Aufwand einen ersten Überblick über große Datensets zu erhalten

Nachteile:

- komplexe Grenzen zwischen Clustern kaum möglich
- sensitiv gegenüber Outliern
- schlechte Initialisierung der Cluster \Rightarrow evtl. falsche Clusterbildung
- Evaluation der Ergebnisse: ohne Expertenwissen schwierig

6.2 - Komponentenanalyse

Grundgedanke Komponentenanalyse

Grundprinzip: Finde eine Repräsentation eines Datensets, mithilfe derer sich die Daten in niedrigerer Dimension darstellen lassen, aber trotzdem noch alle relevanten Informationen vorhanden sind

Anwendungsgebiete:

- Reduktion des Merkmalsraums (vgl. Kapitel 3)
- Blinde Quellentrennung⁸: ordne verschiedene Anteile eines gemischten Signals ihren ursprüngliche Quellen zu (z.B. Sprecherlokalisierung)
- Visualisierung höher-dimensionaler Daten in 2-D oder 3-D (z.B. t-SNE⁹)

⁸Choi et al.: "Blind source separation and independent component analysis: A review", 2004

⁹Van der Maaten und Hinton: "Visualizing Data Using t-SNE", Journal of Machine Learning Research, 2008

Funktionsprinzip:

Mithilfe einer Hauptachsentransformation wird ein Merkmalsraum so transformiert, dass sich korrelierte Merkmale durch ihre Linearkombinationen ausdrücken lassen.

Die PCA ist bereits aus Kapitel 3 zur Feature-Reduktion mittels Feature-Transformation bekannt. Grundsätzlich ist eine solche Feature-Transformation ohne Berücksichtigung von Klassenlabels auch eine Form des unüberwachten Lernens.

Independent Component Analysis

Grundidee: Aufteilung multivariater Signale in additive, unabhängige, nicht-gaußförmige Signalkomponenten

Optimierung: Maximiere statistische Unabhängigkeit der geschätzten Komponenten

Verbreitete Maße für die Unabhängigkeit der Komponenten:

1. minimale Transinformation (engl. mutual information), z.B. mithilfe von Entropie-Maximierung oder Kullback-Leibler-Divergenz
2. maximale non-Gaussianity, z.B. mittels Kurtosis oder Negentropie

Anwendung: beispielsweise im Bereich der blinden Quellentrennung, aber auch für unüberwachte Lernaufgaben^{10,11}

¹⁰M.S. Barlett: "Face image analysis by unsupervised learning", 2001

¹¹Polder et al.: "Estimation of compound distribution in spectral images of tomatoes using independent component analysis", 2003