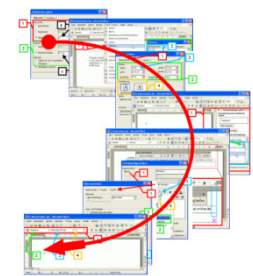


Inhaltliche Planung für die Vorlesung

- ✓ 1) Definition und Geschichte der KI, PROLOG
- ✓ 2) Expertensysteme
- ✓ 3) Logisches Schließen, Resolution
- ✓ 4) Suche und Spieltheorie
- ✓ 5) Spieleprogrammierung
- 6) General Game Playing
- 7) Reinforcement Learning und Spieleprogrammierung
- ➡ 8) **Mustererkennung**
- 9) Neuronale Netze
- 10) Optimierungen (genetische und evolutionäre Algorithmen)
- 11) Bayes-Netze, Markovmodelle
- 12) Robotik, Pathfinding



der rote Vorlesungsfaden...



Mustererkennung - Abgrenzung

- Computergrafik
Synthese von Bildern aus geometrischen Modellen
- Bildbearbeitung
Manipulation von Bildern mit fertiger Software (Photoshop, Corel, Paint, etc.)
- Digitale Bildverarbeitung
Erstellung von Software und Funktionen zur Bildbearbeitung
- Bildanalyse
Extraktion von sinnvollen Informationen aus Bildern
- Mustererkennung
Auffinden von „Mustern“ in Daten und Signalen
- Computervision
Mechanisieren von Sehvorgängen: räumliche Erfassung und Erkennen von Objekten

Bildverstehen



Mensch

- Kann sehr gut Muster (Text, Sprache, Musik, Bilder) erkennen, selbst in komplizierter Form
- Intuitiv, angeboren
- Assoziative Fähigkeiten
- Kann schlecht analytische Prozesse berechnen



Maschine

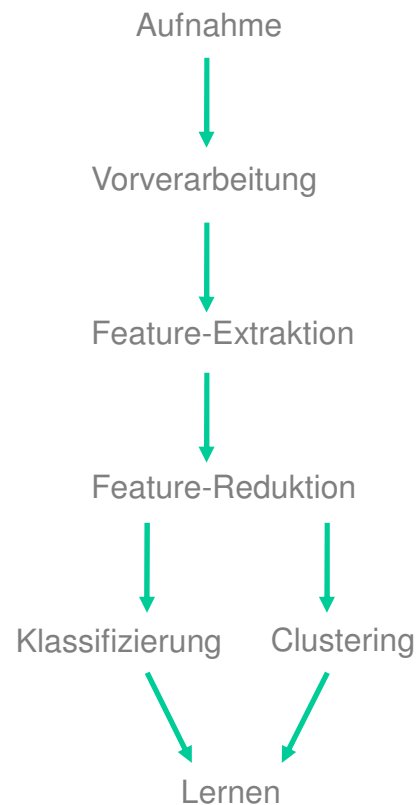
- Kann sehr gut analytische Algorithmen verarbeiten (number crunching)
- Entspricht der Konstruktion
- Kombinatorische Fähigkeiten
- Kann schlecht Muster erkennen

Was dem Menschen einfach fällt, scheint dem Rechner schwer zu fallen.
Was dem Rechner einfach fällt, scheint einem Menschen schwer zu fallen.

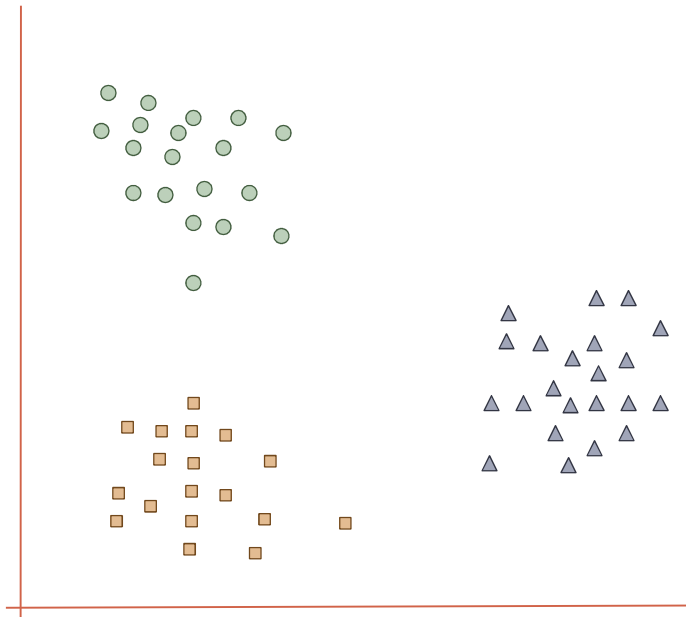
Mustererkennung

Verfahren um gemessene Objekte automatisch in Kategorien (Klassen) einzuordnen durch Erkennen von Muster.

Muster: Merkmale (**Features**), die alle Objekte einer Klasse gemeinsam haben und deutlich von Merkmalen anderer Klassen unterscheiden



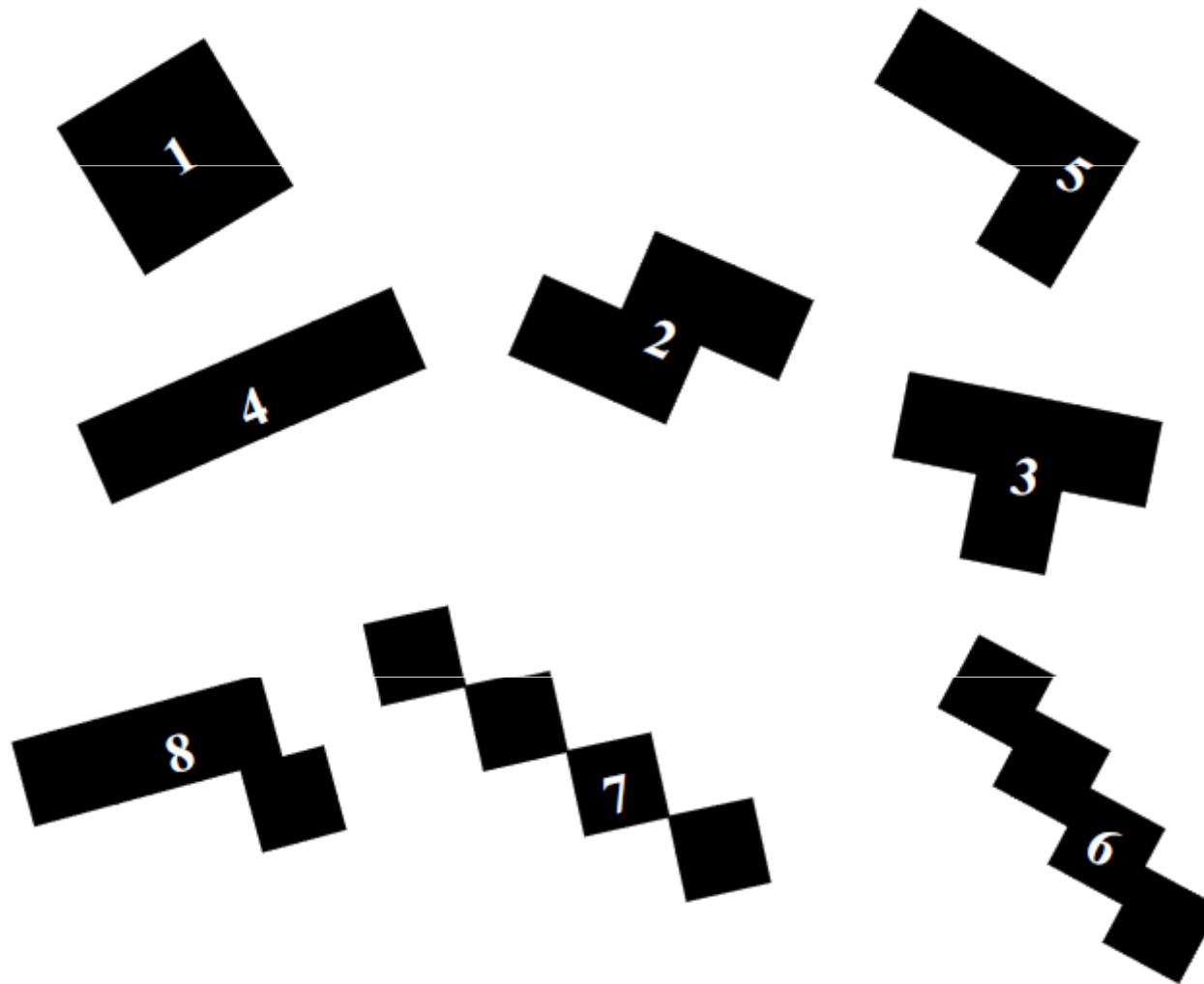
Merkmalsvektor



Jedes Objekt wird durch einen **n-dimensionalen Merkmalsvektor** im **n-dimensionalen Merkmalsraum** repräsentiert.

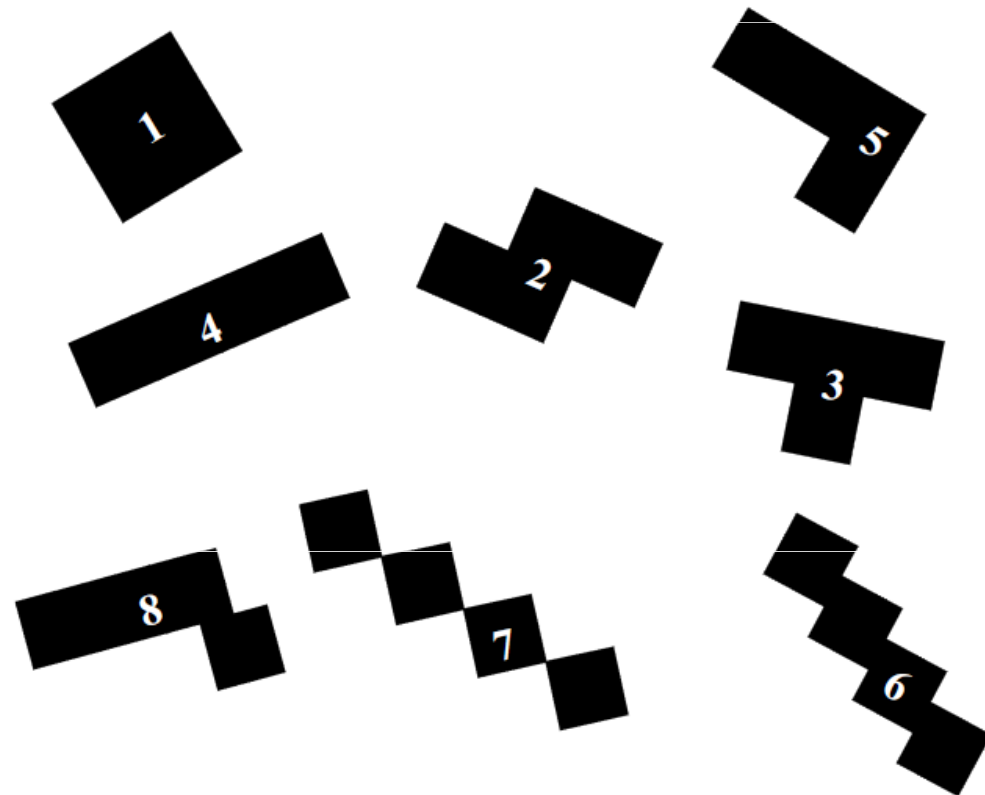
Ähnliche Objekte sollten räumlich im Merkmalsraum beieinander liegen, sodass diese Cluster (Klassen) bilden.

Features



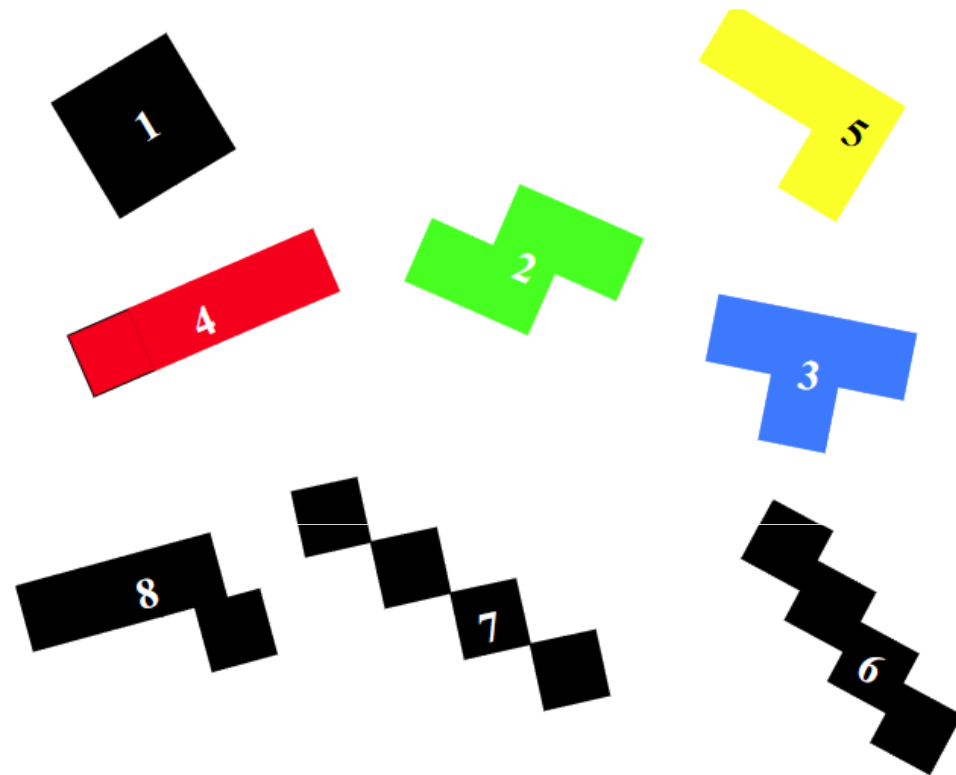
Features

object	area	perimeter
1	4	8
2	4	10
3	4	10
4	4	10
5	4	10
6	4	13
7	4	16
8	4	11



Features

object	area	perimeter	colour
1	4	8	black
2	4	10	green
3	4	10	blue
4	4	10	red
5	4	10	yellow
6	4	13	black
7	4	16	black
8	4	11	black



kNN – k-nächste-Nachbarn

Der KNN-Klassifikation ist der denkbar einfachste Klassifikator:

1. Trainingsphase: Lege alle Datenpunkte in einer Datenbank ab („lazy learning“)
2. Testphase: Neue Datenpunkte werden folgendermaßen klassifiziert:
 1. Berechne Abstand mit allen Trainingspunkten
 2. Nehme k-nächsten Nachbarn für weitere Auswahl
 3. Führe Mehrheitsentscheid aus über die Klassen der k Nachbarn

Abstandsmaß:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Euklidische Distanz

Vorteile:

- Einfach
- Sehr gute Ergebnisse

$$d(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}$$

Mahalanobis Distanz

Nachteile:

- Teuer
- Wie geeignetes k finden?

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

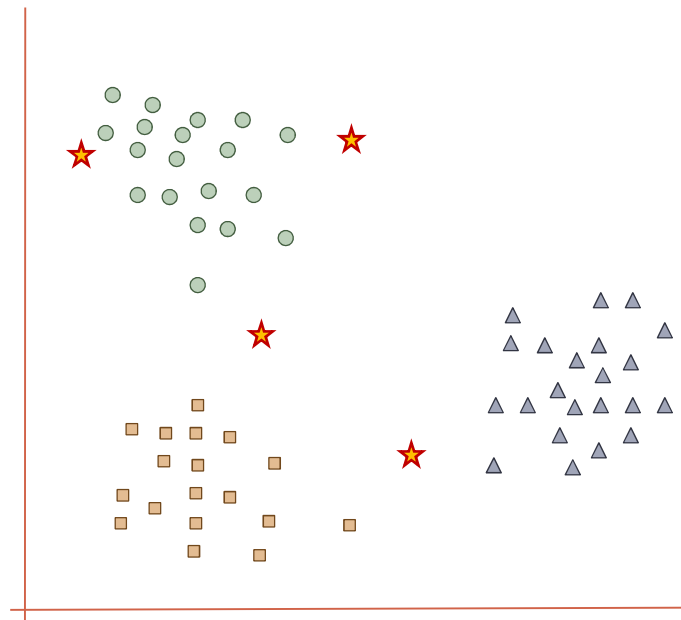
Manhattan Distanz

kMeans

K-Means ist ein iteratives **Clustering-Verfahren** zur Klassifikation

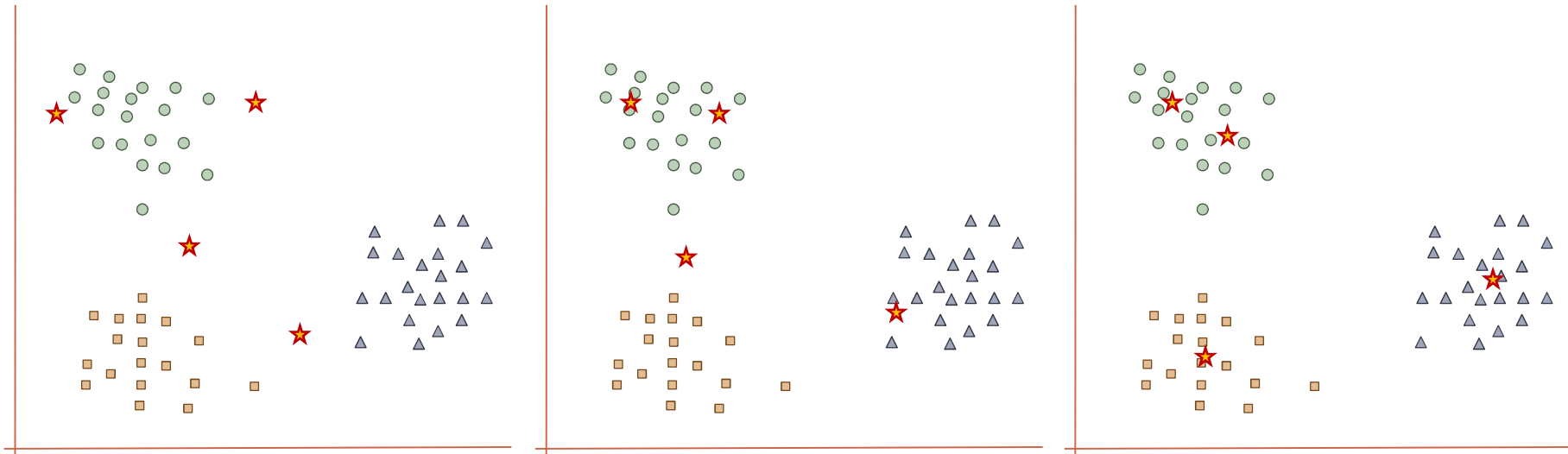
Trainingsphase:

1. Lege alle Datenpunkte in einer Datenbank ab
2. Wähle k Prototypen und lege sie zufällig in den Merkmalsraum
3. Weise jedem Datenpunkte den nächsten Prototypen zu (Partitionierung)



kMeans

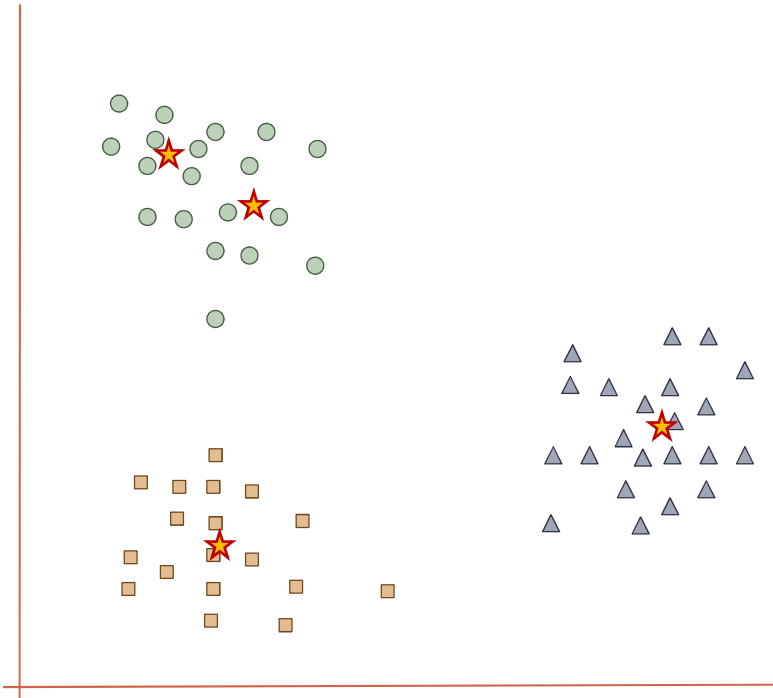
4. Errechne für jeden Prototypen das Clusterzentrum (Schwerpunkt) und setze den Prototypen darauf
5. Goto 3)



kMeans

Testphase:

1. Berechne Abstände zu Prototypen und wähle nächsten Prototyp



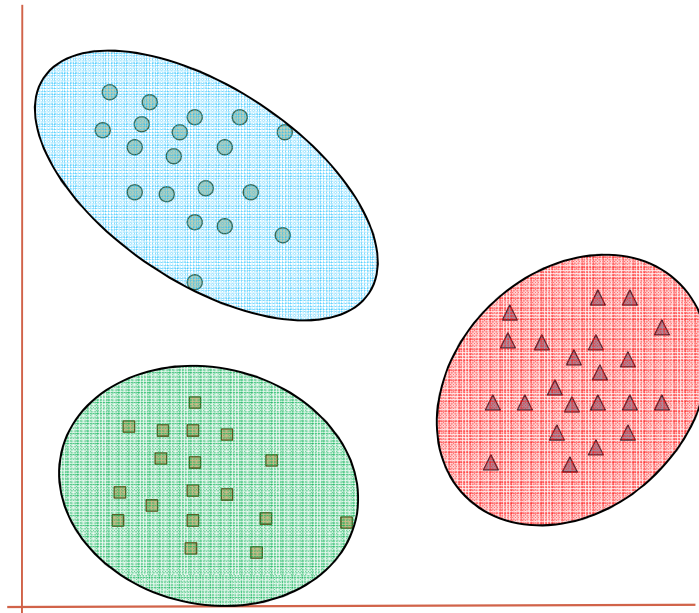
Vorteile:

- Einfach
- schnell

Nachteile:

- Wann mit Schleife enden?
- Konvergiert nicht immer (!)
- Wie geeignetes k finden?

Gauß-Klassifikator



Modelliere Cluster mit n-dimensionalen Normalverteilungen (Gauß-Glocken)

Verteilungsfunktion der einfachen Normalverteilung

$$P(x) = N(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

Mehrdimensionale Verallgemeinerung

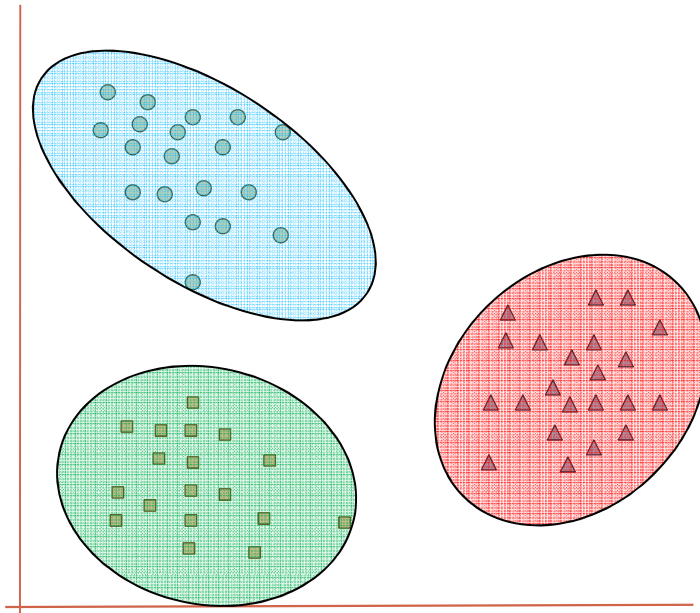
$$P(x|c_i) = N^n(\mu, \sigma^2) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^\top \Sigma_i^{-1} (x - \mu_i)\right)$$

$$\Sigma_i = \frac{1}{n} \sum_{j=1}^n (x_j - \mu_i)(x_j - \mu_i)^\top$$

Kovarianzmatrix des Clusters i

$$\mu_i = \frac{1}{n} \sum_{i=1}^n x_i \quad x_i \in c_i$$

Gauß-Klassifikator



1. Trainingsphase:
 1. Berechne Normalverteilung für alle Klassen
2. Testphase: Neue Datenpunkte x werden folgendermaßen klassifiziert:
 1. Berechne Wahrscheinlichkeit von $P(x|c_i)$ für alle Klassen c_i
 2. Wähle die Klasse mit der höchsten Wahrscheinlichkeit

