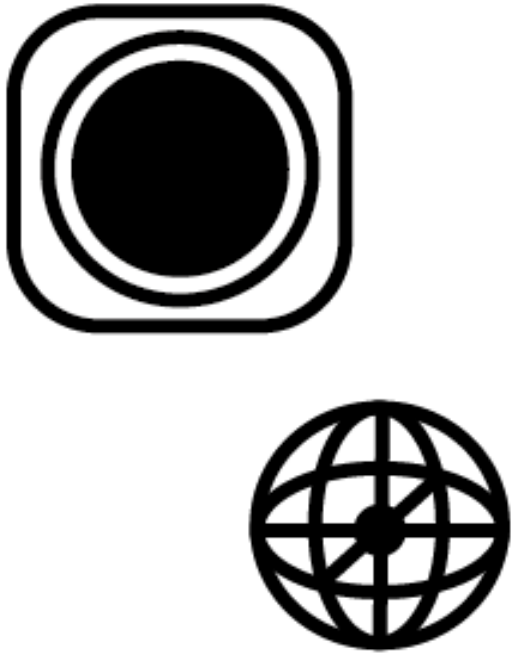
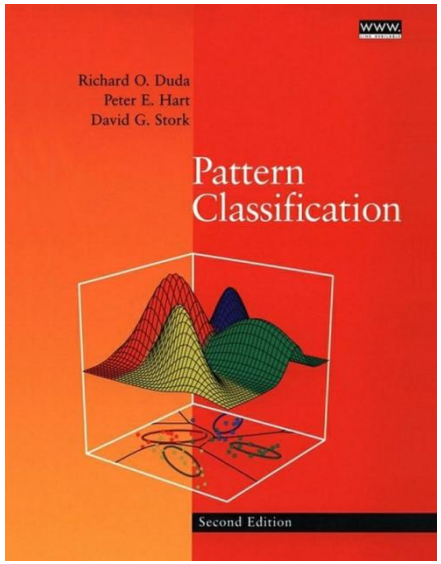


Sensordatenverarbeitung

KLASSIFIZIERUNG (11)

06.01.2025



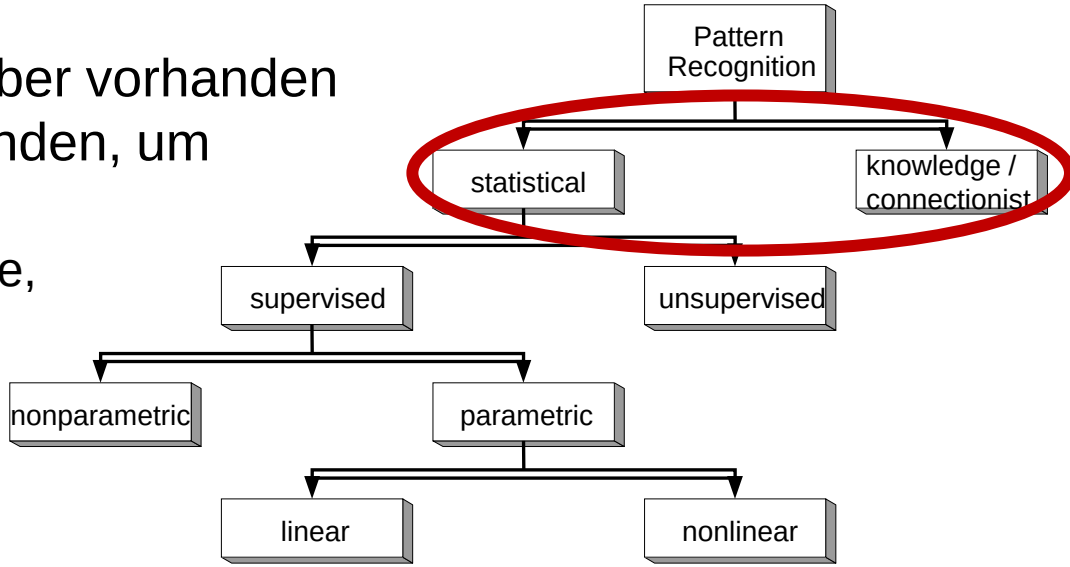
Teil b

Daten versus Wissen

- In unseren Beispielen haben wir aus annotierten Daten (jeder Fisch wurde einer Klasse zugeordnet) eine Beschreibung abgeleitet
  - Entscheidungsgrenze, Algorithmus zur Klassifizierung
- Es wurde also auf Trainingsdaten (in Fisch-Beispiel= Merkmalsvektoren [Länge, Helligkeit] und Angabe von Klasse „Lachs“ | “Barsch“) ein „Modell“ dieser Daten gelernt, die Klassifikation erfolgt durch Auswerten dieses statistischen Modells = **statistische** Mustererkennung

- Alternativ könnte man Wissen über vorhanden Eigenschaften der Daten verwenden, um daraus direkt Regeln abzuleiten
  - z.B. Experte kennt Fanggebiete, Brutverhalten und andere Eigenschaften von Fischen

= **wissensbasierte Entscheidungssysteme**

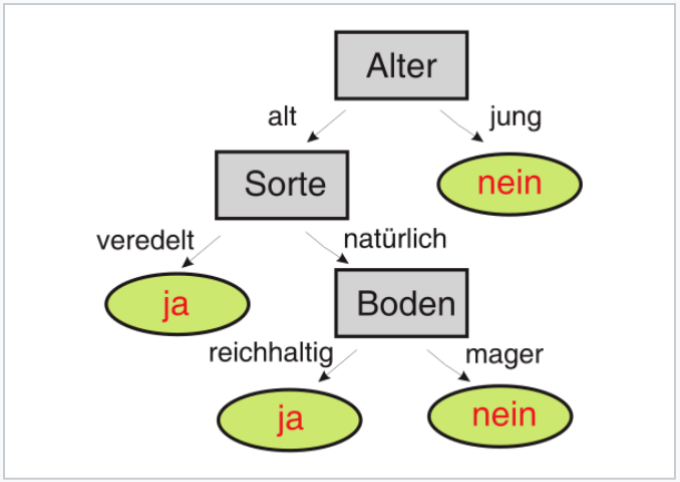


- Klassifikatoren können von Experten in Form von Regeln wissensbasiert „designed/entwickelt“ werden, zum Beispiel:
  - Fischexperte kennt die Fanggründe, Aufenthalt von Fischen, Alter der Population, daher Größe usw. :: REGEL: wenn Fisch in Bereich x gefangen, dann Klasse Barsch
  - Linguist kennt Frequenzverteilung bei Lauten:: REGEL: wenn Formante zu erkennen, dann Vokal
  - Biologe weiß, welche Bäume Früchte tragen

- Regeln können aber auch in Form von Entscheidungsbäumen anhand von Daten gelernt werden
  - automatischen Induktion von Bäumen nach Top-down Prinzip:

1. Chi-square Automatic Interaction Detectors
2. CARTs (*Classification And Regression Trees*)
3. C4.5/5.0 *(moderne Optimierungsstrategien)*

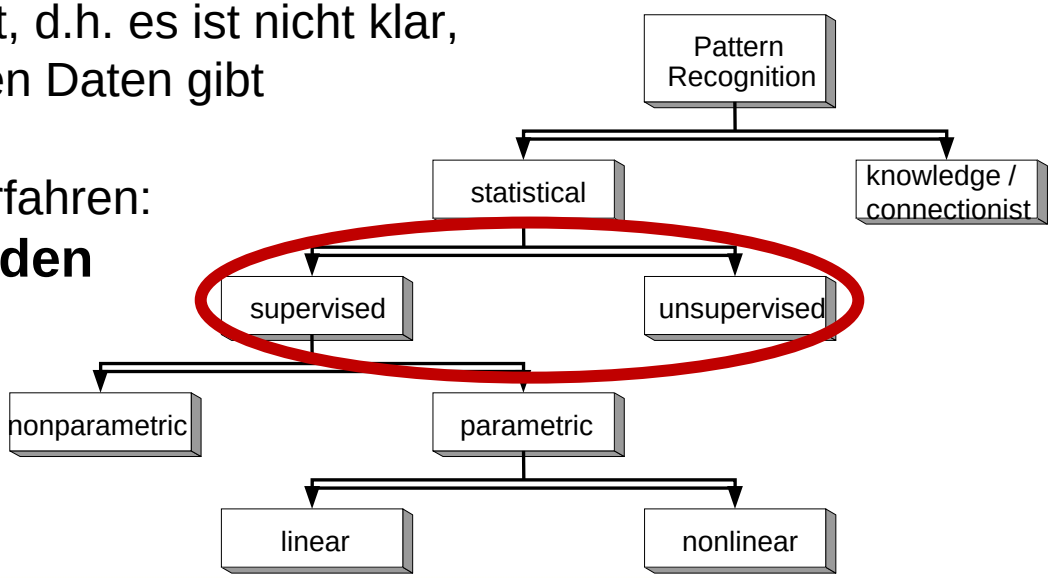
wikipedia



Binärer Entscheidungsbaum zur Vorhersage, ob ein Apfelbaum Früchte tragen wird

Grundlegende Unterscheidungen beim Lernen (statistische Verfahren):

- Überwachtes (supervised) Lernen:
  - Klassenzuordnungen der Daten für die Lernphase sind bekannt
  - Für jedes Trainingsbeispiel kennt man die Klasse
- Unüberwachtes (unsupervised) Lernen:
  - Klassenzuordnungen der Daten sind nicht bekannt ODER
  - Klassen sind generell unbekannt, d.h. es ist nicht klar, ob oder welche Klassen es in den Daten gibt
  - Aufgabe der unüberwachten Verfahren: **Strukturen in den Daten finden**



Strukturen finden: Clustering

- Clustering ist ein typisches **unüberwachtes Lernverfahren**
- **Ziel des Clustering:** automatische Zuordnung der Daten zu Gruppen (wir nennen sie hier nicht „Klassen“, denn die Klassenzugehörigkeit ist nicht bekannt – und kann daher nicht überprüft werden. Es besteht die Hoffnung, dass die gefundenen Gruppen den Klassen entsprechen)
- **Cluster** (Informatik und Statistik):
  gefundene *Gruppe* von Datenobjekten mit ähnlichen Eigenschaften
- **Clusteranalyse**: Verfahren zur Berechnung einer solchen Gruppierung
- **Wichtigstes Clustering-Verfahren**: k-Means Algorithmus
- **Kernidee**:
  - Objekte im selben Cluster verfügen über "ähnliche" Eigenschaften
  - Objekte mit unterscheidbaren Eigenschaften landen in verschiedenen Clustern.

K-Means-Algorithmus

Der *K-Means-Algorithmus:*

100 Fische

- Ziel: Bestimme eine Gruppenstruktur von einer Menge von *N* Samples im *D*-dimensionalen Raum 2 Merkmale (Länge, Hautfarbe) → D=2
- Die Anzahl der Gruppen (Klassen) *K* wird vorgegeben Barsche, Lachse → K=2
- Kriterium der Ähnlichkeit: *Minimierung des quadratischen Fehlers*,
- Fehler = Abstand zwischen einem Sample und dem *Mittelpunkt seiner Gruppe* im D-dimensionalen Raum1 Lachs: L=10cm, Hell=5; Durchschnittslachs 12,7
  - D.h. wenn $x_n^{(k)}$ $(n=1, ..., N_k)$ die Sample der Gruppe *k* bezeichnet und $\mu^{(k)}$ den Mittelwert der Gruppe *k*, dann minimiere

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N_K} \left\| x_n^{(k)} - \mu^{(k)} \right\|^2$$

- Problem: Die $\mu^{(k)}$ hängen selber von der Gruppenzuordnung ab!
  - Beste Zuordnung kann man nicht so einfach ausrechnen (ist ein NP-hartes Problem)
  - Lösung: Verwende einen iterativen Algorithmus

Länge und Helligkeit von Durchschnittslachs ist nicht bekannt!

Animation K-means (k=4)

k-Means Clustering Algorithmus

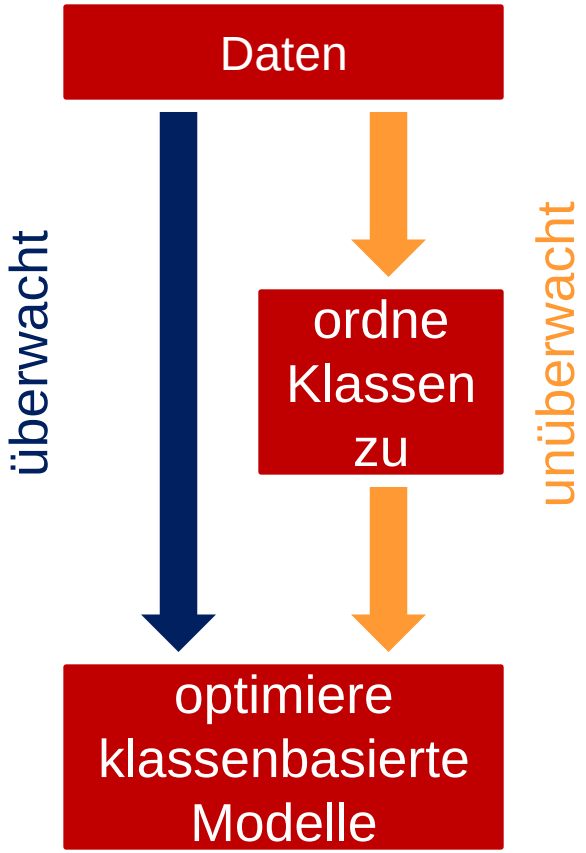
Schritt 1	**Initialisierung** Gegeben festes k und Datensamples $x_1, \dots, x_N$, Initialisiere die Mittelwerte der k Klassen
Schritt 2	**Nearest-Neighbor Klassifikation** Ordne jedes Sample x_i der Klasse zu, deren Mittelwert $\mu_{f(i)}$ es am nächsten liegt
Schritt 3	**Update** Rechne auf Basis dieser Zuordnung neue Mittelwerte μ_i für die Klassen aus
Schritt 4	**Iteration** Falls Abbruchbedingung nicht erfüllt, gehe zu Schritt 2 sonst stoppen

Mögliche Abbruchbedingungen:

- Feste Anzahl Iterationen
- Der Fehler *J* fällt unter einen vorher gewählten Schwellwert
- Die Klassenzuordnung ändert sich nicht mehr oder nur noch wenig

Überwachtes Lernen (Training)

- Überwachtes Lernen / Training:
  Die zugehörige Klasse ist für jedes Sample aus dem *Trainingsdatensatz* bekannt.
  (z.B. für jeden Fisch weiß man, ob Barsch oder Lachs)
  - Vorteil: einfaches Training (sehen wir später)
  - Nachteil: Klassenzugehörigkeiten oft nur mit hohem Aufwand (manuell) zu bekommen
    - (man muss jeden Fisch vorher ansehen und markieren bzw. in getrennte Becken bringen)
- Unüberwachtes Lernen / Training:
  - Zuordnung der Trainingsdaten ist nicht bekannt (z.B. alle Fische gemeinsamen in einem Becken)
  - Die Anzahl und Typen der Klassen wird vermutet (welche Fischarten sind im Becken)
  - Jedes Sample (Fisch) wird mittels Clustering z.B. k-means) einer Klasse zugeordnet



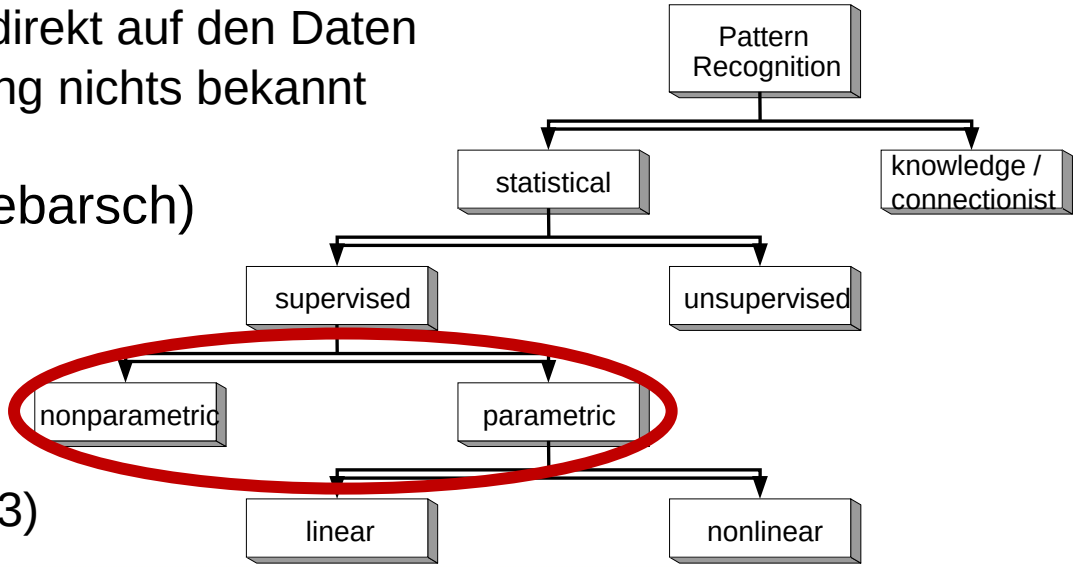
⇒ Aus unüberwachtem Lernen wird durch Anwendung des Clustering (automatische Zuordnung von Klassen) überwachtes Lernen

Lernen (Nicht-)Parametrischer Modelle

- Parametrische versus nichtparametrische Verfahren/Modelle:
  - **Parametrische Verfahren / Modelle**
    - Annahme: Daten folgen gewissen Wahrscheinlichkeitsverteilung
    - Schätze die **numerischen Parameter** dieser Verteilung
    - Vorteil: oft einfache Berechnung und Interpretierbarkeit der Parameter
  - **Nichtparametrische Verfahren / Modelle**
    - Es werden keine Annahme über die Verteilung gemacht
    - Die Daten sprechen für sich selbst, d.h. die Klassenzuordnung erfolgt direkt auf den Daten
    - Vorteil: wenn über Verteilung nichts bekannt

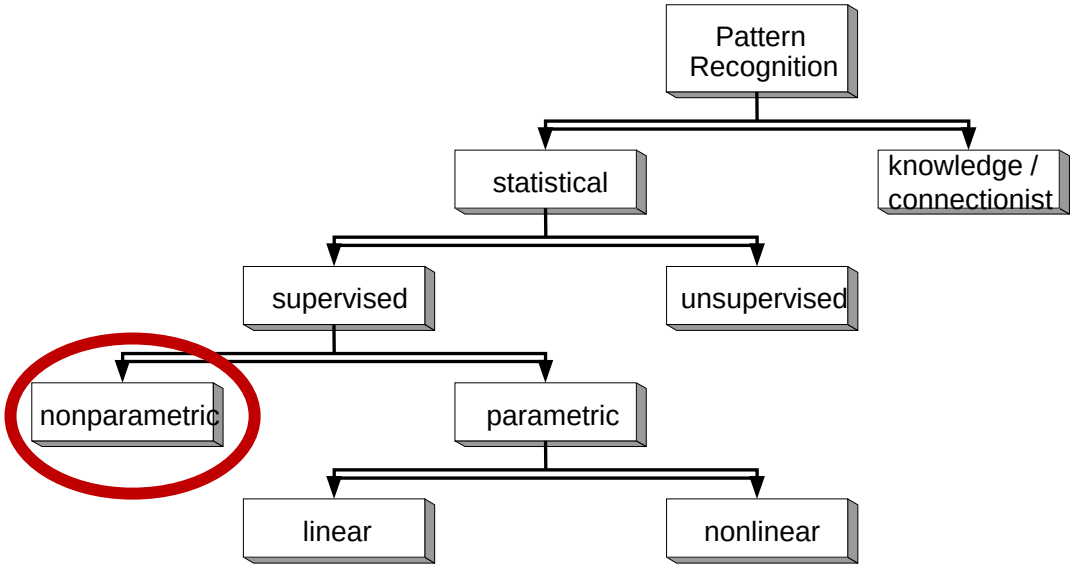
Einführungsbeispiel (Lachs vs Seebarsch)

- Nichtparametrisch:
  - kNN (Lösung 4)
  - Histogramme (Lösung 1, 2)
- Parametrisch:
  - Entscheidungsgerade (Lösung 3)



Nichtparametrische Verfahren

- Parzen Windows
- K-Nearest Neighbors
- Histogramme



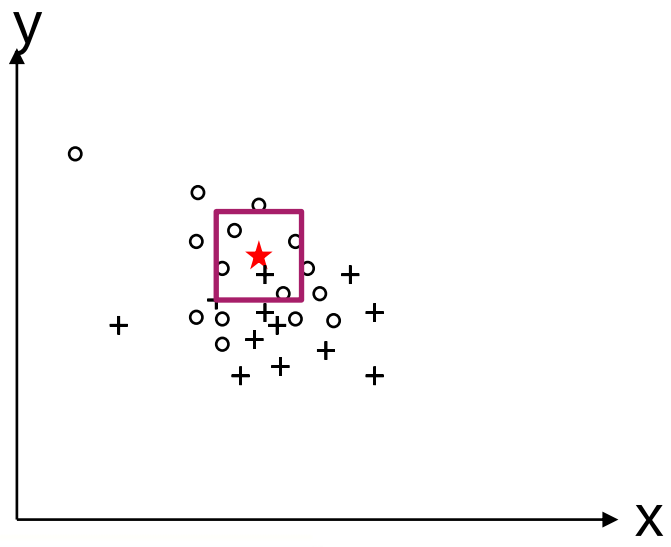
Nichtparametrisches Verfahren: Parzen Windows

- Es wird keine Annahme über die Verteilung getroffen, d.h. die Klassifikation wird direkt aus den Daten abgeleitet.
- Zwei Klassen c_1 und c_2 (Kreuze und Kreise) in einem zweidimensionalen *Merkmalsraum* (= *Feature-Raum*).
- *Features (Merkmale*) sind Datenpunkte (Vektoren), die aus einer beliebigen Vorverarbeitung entstanden sind
- Zu welcher Klasse gehört der rote Stern?

Algorithmus:

- Wähle ein (quadratisches/kreisförmiges/...) Fenster der Größe *V*.
- Zähle die Anzahl der *Samples (Muster)* s_k jeder Klasse *k*, die in das Fenster *V* fallen.
- Beispiel hier: 4 Kreise, 1 Kreuz -> roter Stern wird als „Kreis“ klassifiziert
- Formale Regel:

$$P(\text{Sample } x \text{ gehört zu } c_k) = \frac{s_k}{\sum_\kappa s_\kappa}$$



k-Nearest Neighbors (kNN)

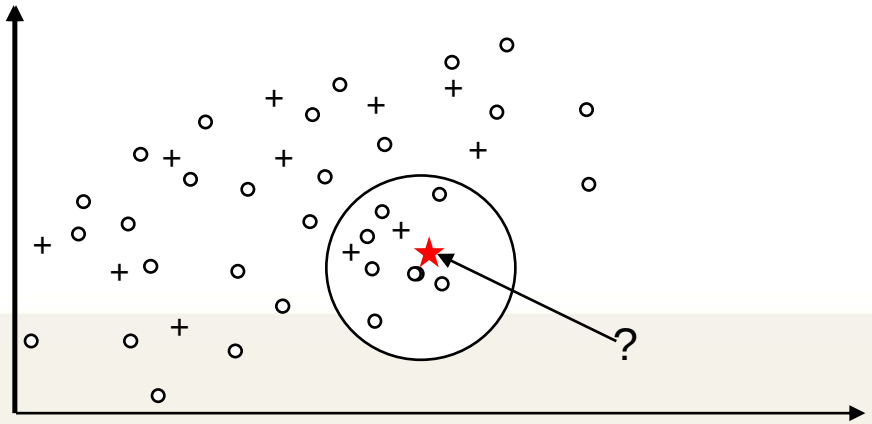
Problem Parzen-windows: Klassifikation ist von der Fenstergröße abhängig

- Fenster zu klein: erratische Abschätzung, fragmentierter Merkmalsraum, Ausreißer werden nicht erkannt; Zu großes Fenster: schlechte Auflösung
- Lösung: Fenstergröße von Datendichte abhängig machen ODER betrachte nicht festen Fenster, sondern feste Zahl n*ächster Nachbarn*

Beschreibung des Algorithmus (K-Nearest Neighbors):

- Finde *k* nächste Nachbarn des zu klassifizierenden Samples *„roter Stern“*.
- Bestimme die häufigste Klasse unter den *k* Nachbarn
- Weise *„roter Stern“* dieser Klasse zu (oder gib Wahrscheinlichkeit an)
- Beispiel: *k=9, 7x* Kreis, *2x* Kreuz => klassifiziere *„roten Stern“* als Kreis

Auch der kNN-Algorithmus hat Probleme: man braucht ein geeignetes *k* !



Histogramme zur Dichteschätzung

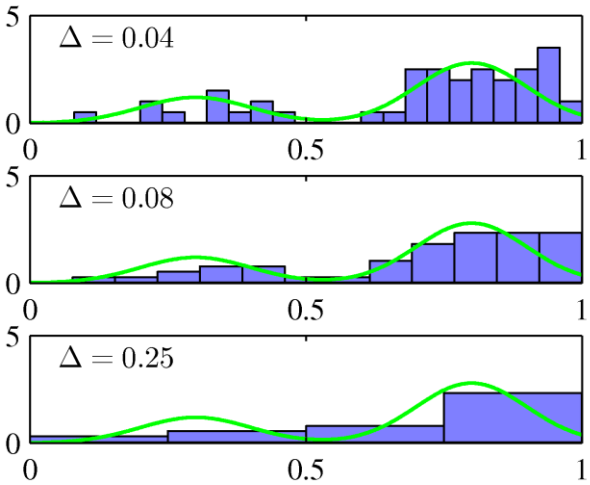
Ansatz: Betrachte nicht die Daten, sondern bilde *Histogramm aus* Datenpunkten. Die Größe der einzelnen *bins* muss vorher festgelegt werden.

- Zu kleine bins: Viel erratische Struktur, die in den eigentlichen Daten nicht vorhanden ist.
- Zu große Klassen: Struktur geht verloren.
- Skalierungsprobleme im hochdimensionalen Raum (Anzahl der bins einer festen Größe steigt exponentiell)

Es gibt aber auch Vorteile:

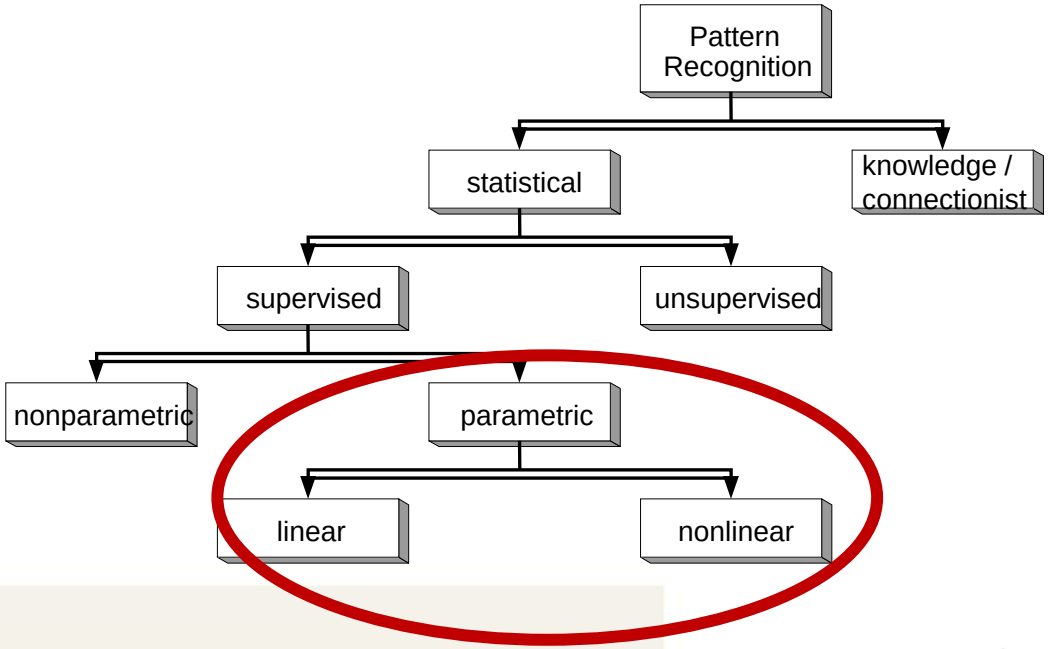
- Rechenaufwand und Speicheraufwand für Daten sind reduziert
- Gut geeignet, wenn Daten sequentiell hinzugefügt werden.

Beispiel: Dichteschätzung mit Histogrammen, grün eingezeichnet ist die wahre Dichte. Der optimale Wert für die Größe der *bins* liegt im Beispiel etwa bei 0.08.



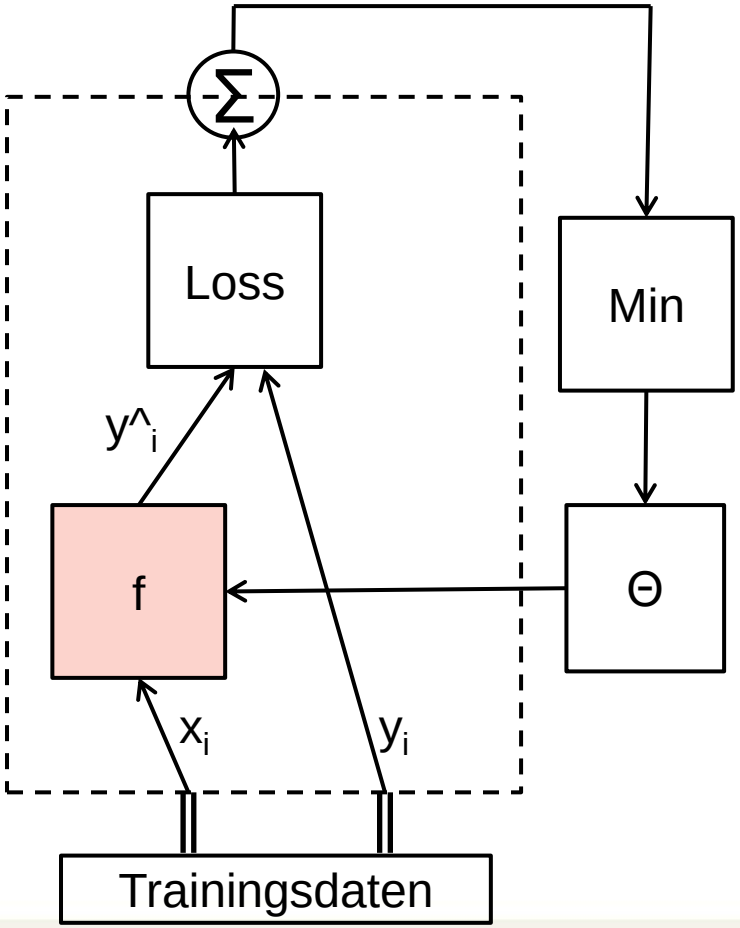
Parametrische Verfahren

- Lineare Klassifikatoren
- Bayes Entscheidungstheorie
- Gaussklassifikatoren
- Support Vektor Maschinen (SVMs)
  (werden aus Zeitgründen nicht besprochen)



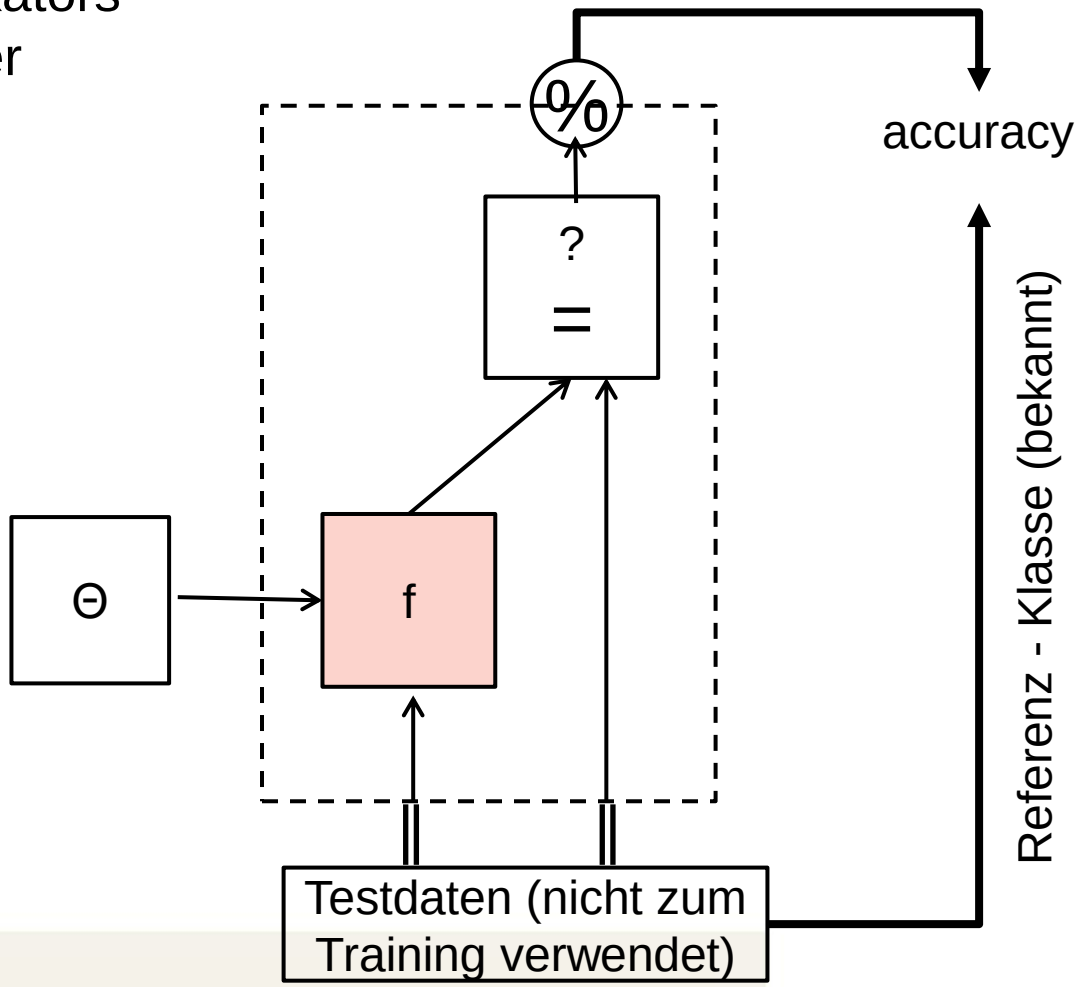
ML: Trainieren eines Klassifikators

- Parametrische Verfahren – Machine Learning (ML) = Parameter Θ lernen
- Wende auf jedes Datensample die Operation im gestrichelten Kasten an



ML: Evaluieren des Klassifikators

- Nach dem Training wird auf jedes Datensample der (ungesehenen) Testdaten (Klasse bekannt) das gelernte Modell angewendet und die Performanz des Klassifikators (z.B. Akkuratheit) anhand der Referenz ermittelt



ML: Anwendung

- Wende das Modell mit Parameter Θ auf die Daten der Anwendung an