

1 Erkennung

1.1 Das Problem der Objekterkennung

1.1.1 Herausforderungen der Erkennung

Menschliche Wahrnehmung ist Computern weit überlegen. Die Erkennung ist schwierig aufgrund von:

- **Mehrdeutigkeit & Illusionen:** (z.B. Don Quixote-Gesicht, Mona Lisa-Mosaik, ambivalente Elefantenbeine).
- **Tarnung & Verdeckung:** (z.B. Bevo Doolittles Pferde im Schnee).
- **Kontext:** Die Interpretation von Teilen hängt stark vom Gesamtkontext ab (z.B. verschwommene Straßenszene, Punktwolken-Wörter).
- **Fehler im Kontext:** M.C. Escher-Bilder sind *lokal* konsistent, aber *global* unmöglich.

1.1.2 Intuition: Wie funktioniert Erkennung?

Objekterkennung basiert auf zwei Hauptkomponenten:

1. **Lokale Beschreibung / Merkmale:** (z.B. Augen, Nase, Mund bei einem Gesicht).
2. **Globale Anordnung der Merkmale:** (z.B. relative Positionen und Größen der Merkmale zueinander).

Weitere wichtige Aspekte sind Segmentierung und Szenenkontext.

Pictorial Structure (Fischler & Elschlager, 1973) Ein klassisches Modell zur Objekterkennung, das diese Intuition formalisiert.

- **Teile (Parts):** 2D-Bildfragmente (lokale Merkmale).
- **Aufbau (Structure):** Die Anordnung der Teile, oft modelliert durch “Federn”, die die relative Position und Deformation beschränken.

Herausforderungen für dieses Modell sind **Deformationen** (Teile ändern ihre Position) und **Durcheinander** (Clutter, irrelevante Merkmale im Hintergrund).

1.2 Bayes Decision Theory (BDT)

Die BDT ist ein mathematisches Framework zur optimalen Entscheidungsfindung unter Unsicherheit. Ihr Ziel ist es, Klassifikationsentscheidungen so zu treffen, dass die **erwartete Fehlklassifikationsrate minimiert** wird. Sie nutzt dazu Wahrscheinlichkeiten als formale Beschreibung von Unsicherheit.

Die drei Kernkonzepte der BDT

1. **A Priori (Prior) $P(C_k)$:** Die Wahrscheinlichkeit einer Klasse C_k , *bevor* wir Daten x beobachtet haben. Der Prior kodiert unser Vorwissen oder unsere Erwartungen über die Häufigkeit von Klassen. Wichtig: Priors wirken wie ein “Bias” zugunsten bestimmter Klassen.
2. **Klassenspezifische Dichte (Likelihood) $p(x|C_k)$:** Die Wahrscheinlichkeit (genauer: Wahrscheinlichkeitsdichte), die Merkmalsausprägung x zu beobachten, *unter der Annahme*, dass die wahre Klasse C_k ist. Sie beschreibt also, wie typisch ein bestimmtes Datenmuster x für eine Klasse ist. Hinweis: Likelihood ist eine Funktion in x , nicht in der Klasse.
3. **A Posteriori (Posterior) $P(C_k|x)$:** Nachbeobachtete Wahrscheinlichkeit einer Klasse, *nachdem* wir x kennen. Dies ist genau die Größe, die wir für die eigentliche Entscheidung benötigen.

Bayes' Theorem (Verbindung der Konzepte)

$$P(C_k|x) = \frac{p(x|C_k) P(C_k)}{p(x)}$$

Die Evidenz

$$p(x) = \sum_j p(x|C_j) P(C_j)$$

stellt sicher, dass die Posterior-Wahrscheinlichkeiten normiert sind. Wichtig: Für die Entscheidung ist $p(x)$ oft irrelevant, da es für alle Klassen gleich ist.

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidenz}}$$

1.2.1 Bayes-Entscheidungsregel (Fehlerminimierung)

Wir wählen die Klasse mit der größten posterioren Wahrscheinlichkeit:

$$C^*(x) = \arg \max_k P(C_k|x).$$

Da der Nenner $p(x)$ konstant ist, genügt:

$$C^*(x) = \arg \max_k p(x|C_k) P(C_k).$$

Intuition: Die Likelihood misst, wie gut die Daten zu einer Klasse passen; der Prior berücksichtigt, wie plausibel diese Klasse im Voraus ist. Der Schnittpunkt der Produkte $p(x|C_k)P(C_k)$ bildet die **Entscheidungsgrenze**.

1.2.2 Likelihood Ratio Test (Spezialfall 2 Klassen)

Für zwei Klassen C_1 und C_2 :

$$\text{Entscheide } C_1 \text{ wenn: } \frac{p(x|C_1)}{p(x|C_2)} > \frac{P(C_2)}{P(C_1)}.$$

Links: datengetriebene Evidenz, Rechts: Schwellwert aus Vorwissen. So sieht man explizit, wie ein starker Prior eine Entscheidung "umkippen" kann.

1.2.3 Prior und Likelihood in der Praxis

Die Trennung zwischen Datenmodell (Likelihood) und Vorwissen (Prior) ermöglicht robuste Modelle:

- **Spracherkennung:** Likelihood = akustisches Modell; Prior = Sprachmodell. Das Sprachmodell zieht uns aus akustischen Mehrdeutigkeiten heraus.
- **Bildverarbeitung:** Likelihood = Appearance Model; Prior = Kontextwissen über sinnvolle Objekte am Ort.

1.3 Naive Bayes Klassifikator

1.3.1 Das Problem hoher Dimensionen

Für d Merkmale:

$$x = (x_1, \dots, x_d)$$

müssten wir die gemeinsame Dichte $p(x|C_k)$ im d -dimensionalen Raum schätzen. Dies benötigt exponentiell viele Daten: **Fluch der Dimensionalität**. Schon wenige Dutzend Dimensionen machen klassische Dichteschätzung praktisch unmöglich.

Die Naive-Unabhängigkeitsannahme Um das Problem zu umgehen, wird angenommen:

$$p(x|C_k) = p(x_1, \dots, x_d|C_k) \approx \prod_{i=1}^d p(x_i|C_k).$$

Das heißt: Die Merkmale sind *bedingt unabhängig* gegeben die Klasse. Diese Annahme ist oft falsch, aber in vielen Anwendungen erstaunlich effektiv, weil:

- Abhängigkeiten zwischen Merkmalen sich teilweise gegenseitig ausgleichen,
- Die Entscheidung meist nur eine Rangordnung braucht (MAP), nicht exakte Wahrscheinlichkeiten.

1.3.2 Naive Bayes Entscheidungsregel

Durch Einsetzen der Zerlegung:

$$C^*(x) = \arg \max_k P(C_k) \prod_{i=1}^d p(x_i|C_k).$$

Lernen bedeutet:

1. Schätzen der Priors $P(C_k)$ (Häufigkeiten).
2. Schätzen der eindimensionalen Dichten $p(x_i|C_k)$ für alle Merkmale und Klassen.

Log-Space (Numerische Stabilität) Da Produkte kleiner Wahrscheinlichkeiten numerisch instabil sind:

$$C^*(x) = \arg \max_k \left[\log P(C_k) + \sum_{i=1}^d \log p(x_i|C_k) \right].$$

Logarithmen ersetzen Multiplikationen durch Additionen, ohne die Entscheidung zu verändern.

1.4 Fallstudie: Gesichtsdetektion

Ein Beispiel für **Appearance-Based Methods**: Modelle werden direkt aus (großen) Bilddatensammlungen gelernt.

1.4.1 Sliding Window Ansatz

- Ein Fenster (z.B. 19×19 Pixel) wird über das gesamte Bild geschoben.
- Bei jeder Position wird entschieden: “Gesicht” (C_1) oder “Kein Gesicht” (C_2).
- Das Bild wird skaliert (z.B. Faktor 1.2 verkleinert) und der Vorgang wiederholt sich, um Gesichter aller Größen zu finden.

Fallstudie: Schneiderman & Kanade (1998) Ein sehr erfolgreicher Gesichtsdetektor, der auf Naive Bayes basiert.

- **1. Repräsentation (Merkmale x_i): Wavelet-Koeffizienten** an bestimmten Frequenzen, Orientierungen und **Positionen** (f_i, u_i, v_i). Dies kodiert sowohl lokale Merkmale (Kanten) als auch deren globale Anordnung (Position).
- **2. Trainingsdaten:**
 - *Positive Beispiele* (C_1): Tausende von Bildern, die Gesichter enthalten (normalisiert).
 - *Negative Beispiele* (C_2): Tausende von Bildern, die *keine* Gesichter enthalten.
- **3. Klassifikator & Lernen:** Naive Bayes. Das “Lernen” besteht darin, die Wahrscheinlichkeiten $P(x_i|C_{\text{face}})$ und $P(x_i|C_{\text{non-face}})$ für jedes Merkmal x_i zu schätzen. Dies geschieht durch **Zählen (Erstellen von Histogrammen)** in den positiven und negativen Trainingsdatensätzen.
- **Multi-View:** Um Gesichter aus verschiedenen Winkeln zu erkennen, werden separate Detektoren trainiert (z.B. Frontal, Linksprofil, Rechtsprofil) und die Ergebnisse kombiniert.

1.5 Erkennungsarten (Biometrie)

Gesichtserkennung ist ein biometrisches Verfahren. Man muss zwischen verschiedenen Aufgaben unterscheiden:

- **Detektion:** (Face vs. Non-Face) Ist überhaupt ein Objekt (Gesicht) vorhanden?
- **Verifikation (1:1):** “Bin ich das?” (z.B. Smartphone entsperren).
 - Eine Person gibt ihre Identität an (z.B. Nutzer-ID).
 - Das System vergleicht die aktuelle Probe **nur mit dem einen** gespeicherten Template dieser ID.
 - Ausgabe: Ja / Nein.
- **Identifikation (1:n):** “Wer ist das?” (z.B. Überwachung).
 - Eine Person zeigt nur ihr Merkmal (Gesicht).
 - Das System vergleicht die Probe mit **allen n** Templates in der Datenbank.
 - Ausgabe: Eine Kandidatenliste (die m besten Treffer, $m \ll n$).