

Visual Computing

Objekterkennung und Bayes

Übungsblatt 3

Patrick Elsen, Dmytro Klepatskyi,
Costa Weiland, Nana Atchoukeu Chris-Mozart

Wintersemester 2018-2019
Technische Universität Darmstadt

Aufgabe 1: Gesichtserkennung

Benennen Sie das im folgenden Bild dargestellte Verfahren und erklären Sie kurz, wie dieses funktioniert.

Das auf dem Bild dargestellte Verfahren heißt *Sliding Window Approach* (Suchen über Raum und Skalierung). Das Verfahren gehört zu den *Appearance-Based Methoden*. Das Bild wird zunächst horizontal und vertikal gescannt. Dann wird das Bild um 1,2 verkleinert und wieder gescannt. Dies wird wiederholt, bis das Bild zu klein ist.

Wie setzen sich die Trainingsdaten für dieses Verfahren zusammen?

Die Trainingsdaten setzen sich aus sowohl negativen (keine Gesichter) als auch positiven Beispielen (Gesichter). Durch diese Datensammlung ist für das Lernen des Erscheinungsmodells relevant.

Aufgabe 2: Bayes Decision Theory

Was sagen prior $P(C_k)$, likelihood $P(x|C_k)$ und posteriori $P(C_k|x)$ aus? Hinweis: C_k ist eine Klasse und x ein Merkmal.

Die *a priori* oder prior Wahrscheinlichkeit sagt aus, mit welcher Wahrscheinlichkeit wir eine gewisse Klasse erwarten, ohne dass wir irgendwelche weiteren Informationen haben. Wenn man von der Buchstabenerkennung von deutschem Text ausgeht, dann kann man mit einem kleinen Ruby-Programm die relativen Wahrscheinlichkeiten der Buchstaben im Deutschen ermitteln.

```
1 def sorted_distribution(text)
2   absolute_distribution = text
3   .scan(/[a-zA-Z]/)
4   .inject({}|m,c| m[c] = (m[c]||0)+1; m}
5   total = absolute_distribution.values.sum
6   relative_distribution = absolute_distribution
```

```

7     .map{|k, v| [k, v.to_f / total]}
8     sorted = relative_distribution
9     .sort_by{|k, v| v}
10 end
11
12 # https://raw.githubusercontent.com/martinth/mobverdb/master/faust.txt
13 faust_distribution = sorted_distribution(File.read("faust.txt"))
14 puts "Faust: P(C_#{@faust_distribution.last[0]}) = #{@faust_distribution.last[1]}"

```

Dieses Programm gibt aus $P(C_e) = 0,14328213937810988$. Das bedeutet also, wenn wir deutsche Buchstaben erkennen wollen, und wir nichts weiter über den Buchstaben wissen, dann wissen wir, dass der Buchstabe mit einer Wahrscheinlichkeit von 14,3% ein kleines e sein könnte.

Wenn wir genauer über Buchstaben urteilen wollen, dann müssen wir uns Merkmale anschauen und messen. Das können zum Beispiel Dimensionen, Pixelwerte oder ähnliches sein. $P(x|C_k)$ ist die Wahrscheinlichkeit, dass ein Buchstabe der Klasse C_k das Merkmal x besitzt. Das kann man lesen als *die Wahrscheinlichkeit von x wenn C_k gegeben ist*. Dies nennt man auch *likelihood*.

Fast das Gegenteil dazu ist die *posteriori*-Wahrscheinlichkeit. Hat man einen unbekannten Buchstaben, bei dem man das Merkmal x gemessen hat, dann sagt einem $P(C_k|x)$ wie groß die Wahrscheinlichkeit ist, dass es sich hierbei um einen Buchstaben der Klasse C_k handelt.

Gegeben sei der Satz „Der Satz von Bayes ist nach dem englischen Statistiker Thomas Bayes benannt.“ Bestimmen Sie die Prior-Wahrscheinlichkeit des Buchstaben „s“ im zuvor zitierten Satz. Hinweis: Nur Buchstaben werden als relevante Zeichen für die Prior-Wahrscheinlichkeit betrachtet. Kommas, Punkte, Leerzeichen, etc. werden ignoriert.

Hier können wir wieder das Ruby-Programm verwenden. Dazu führen wir die Funktion einfach auf dem gegebenen Text aus.

```

1 quote = "Der_Satz_von_Bayes_ist_nach_dem_englischen_Statistiker_Thomas_Bayes_benannt."
2 quote_distribution = sorted_distribution(quote)
3 puts "Quote: P(C_s) = #{@quote_distribution.to_h['s']}"

```

Dabei bekommen wir den Output $P(s) = 0,09375$.

Zusätzlich ist $P(x|s) = 0,4$ und $P(x) = 0,1$ gegeben, wobei x ein beliebiges Merkmal ist. Berechnen Sie die *a-posteriori*-Wahrscheinlichkeit $P(s|x)$.

Hier kann man einfach eine Formel anwenden.

$$P(s|x) = \frac{P(x|s)P(s)}{P(x)} = \frac{0,4 \cdot 0,09375}{0,1} = 0,00375$$

Wie groß ist die Summe der Prior-Wahrscheinlichkeiten für alle relevanten Zeichen im oben zitierten Satz? Per Definition ist die Summe aller Zeichen 1.

Was nimmt der Naive Bayes Klassifikator an? Trifft diese Annahme immer zu?

Der naive Bayes-Klassifikator nimmt an, dass alle Merkmale x_0, \dots, x_n statistisch unabhängig sind. Das ist nicht immer wahr, gibt aber in der Praxis gute Resultate.

Aufgabe 3: Erkennungsarten

Wie unterscheiden sich die Begriffe Identifikation und Verifikation voneinander?

Bei der Verifikation gibt die Person ihre Identifikationsdaten (Benutzer-ID, E-Mail) dem System, und das System verifiziert anhand der biometrischen Merkmale ob die Person tatsächlich präsent ist.

Bei einer Identifikation findet das System allein anhand der biometrischen Merkmale heraus, um welche Person es sich handelt, und verifiziert diese. Das System muss die biometrischen Daten also mit denen aller Benutzer in den Referenzmerkmalsdatensätzen abgleichen.

Geben Sie für jede Erkennungsart einen Anwendungsfall an.

Der Irisscanner an der Arbeitsstelle ist ein Verifikationssystem. Hier präsentiert man seinen Arbeitsausweis, der dem System per RFID mitteilt um welche Person es sich handelt. Der Irisabgleich ist also nur dazu da, um sicherzustellen, dass es sich wirklich um die richtige Person handelt, bevor man sicherheitskritische Bereiche betreten kann.

Die Gesichtserkennung, die von der chinesischen Regierung benutzt wird, um als Teil des *Social Credit*-Programms die Aktionen der Bürger zu überwachen¹, ist ein Beispiel für ein Identifikationssystem. Hierbei werden aufgezeichnete Gesichter in Echtzeit mit einer Genauigkeit von 90% mit einer 1,2 Milliarden großen Datenbank verglichen, um die Menschen zu identifizieren und negative Aktionen zu bestrafen. Beispielsweise können Punkte abgezogen werden, wenn man bei Rot über die Ampel geht.

¹<https://www.scmp.com/node/2157883>