
Inhaltsverzeichnis

1	Einleitung	1
1.1	Definitionen, Begriffe	1
1.2	Grundsätzliche Vorgehensweise	3
2	Intuitive Klassifikation	6
2.1	Abstandsmessung zur Klassifikation	6
2.1.1	Nächster-Nachbar-Klassifikator	7
2.1.2	Effiziente Berechnung für NN-Klassifikatoren	9
2.1.3	Klassifikation durch Vektorquantisierung	11
2.1.4	k-Means	12
2.1.5	LBG-Verfahren	14
2.1.6	Learning-Vector-Quantization	15

1 Einleitung

1.1 Definitionen, Begriffe

Perzeption hat zum Gegenstand die Eindrücke der Umwelt

Umwelt : $U = \{\rho \vec{b}(\vec{x}) \mid \rho = 1, 2, \dots\}$

(Menge aller meßbaren Größen; \vec{x} ist Zeit, Raum und \vec{b} sind Messungen physikalischer Eigenschaften, z.B. Lautstärke)

Problemkreis : $\Omega = \{\rho \vec{f}(\vec{x}) \mid \rho = 1, 2, \dots\} \subset U$

(Menge von Mustern des gleichen Typs)

Muster : Elemente eines Problemkreises

$$\vec{f}(\vec{x}) = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{bmatrix}$$

1.1 Definitionen, Begriffe

Klassifikation von (einfachen) Mustern, ist die Zuordnung eines Musters zu einer von k erlaubten Klassen Ω_λ

Beispiel: Buchstabe, isoliert gesprochenes Wort

Musterklasse od. Klasse : Zerlegung des Problemkreises in disjunkte Klassen:

$$\begin{aligned}\Omega_\lambda &\neq \emptyset \\ \Omega_\lambda \cap \Omega_\kappa &= \emptyset \quad \text{für } \lambda \neq \kappa \\ \bigcup_{\lambda=1}^k \Omega_\lambda &= \Omega\end{aligned}$$

Erweiterung um Rückweisungsklasse $\Omega_0 \Rightarrow \bigcup_{\lambda=0}^k \Omega_\lambda = \Omega$

Einfache Muster :

- dem Anwender genügt der Klassenname
- die Klassifikation ist als Ganzes möglich

1.2 Grundsätzliche Vorgehensweise

Bei allen Unterschieden zwischen verschiedenen Ansätzen gibt es einige gemeinsame Prinzipien (Niemann 83):

Postulat 1: es steht eine repräsentative Stichprobe ω zur Verfügung

$$\omega = \left\{ {}^1\vec{f}(\vec{x}), \dots, {}^N\vec{f}(\vec{x}) \right\} \subset \Omega$$

- $\omega \subset \Omega$: da alles andere irrelevant
- in der Regel klassifizierte Stichprobe
- repräsentativ: wichtig, aber schwer zu erreichen, sogar schwer zu beurteilen
- Frage nach N (Größe der Stichprobe)
 - abhängig von statistischen Eigenschaften des Problemkreises
 - oft N eher zu klein, als groß (wegen Aufwand)

1.2 Grundsätzliche Vorgehensweise

Für die Klassifikation:

Postulat 2: Ein einfaches Muster besitzt Merkmale, die für die Zugehörigkeit zu seiner Klasse charakteristisch sind.

- numerische Merkmale: Vektor reeller Zahlen
- symbolische Merkmale: Kette von Symbolen

Merkmale werden aus dem Muster durch die **Merkmalsextraktion** gewonnen, für numerische Merkmale:

$$\varphi : \varphi \left(\vec{f}(\vec{x}) \right) = \vec{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_M \end{pmatrix}$$

Dabei sollte die Dimension der Merkmalsvektoren M möglichst klein sein, u.a. da von dieser Größe der mathematische Aufwand bei der Klassifikation abhängt.

Postulat 3: Die Merkmale für Muster einer Klasse bilden einen einigermaßen kompakten Bereich. Die Bereiche für verschiedene Klassen sind getrennt.

- geeignete Merkmale finden ist (oft) schwierig
- die Klassifikation wird relativ gut beherrscht
- lokale Optimierung gewährleistet i.A. nicht eine Optimierung des gesamten Systems

2 Intuitive Klassifikation

2.1 Abstandsmessung zur Klassifikation

quantisiere die Ähnlichkeit von Mustern mit einer Metrik. z.B.:

$$d^{(r)}(\vec{c}_i, \vec{c}_j) = \sqrt[r]{\sum_{\nu} |c_{i\nu} - c_{j\nu}|^r}$$

$$c_i, c_j \in \omega$$

$c_{i\nu}, c_{j\nu} \in \omega$ sind die Komponenten der Merkmalsvektoren

- $r = 1$ City-Block Metrik
- $r = 2$ euklidischer Abstand
- $r = \infty$ Maximum Norm

2.1.1 Nächster-Nachbar-Klassifikator

Bei gegebener klassifizierter Stichprobe klassifizieren wir eine (neues) Muster in die Klasse des nächsten Musters innerhalb dieser Stichprobe:

\vec{c} ordnen wir Ω_{κ} zu, falls

$$i = \operatorname{argmin}_{j=1,\dots,N} \{d(\vec{c}, \vec{c}_j)\} \text{ und } \vec{c}_i \in \Omega_{\kappa}$$
$$\text{und } \omega = \{\vec{c}_j | j = 1, \dots, N\}$$

Bemerkung:

- gibt gute Klassifikationsergebnis (was immer das heißen mag an dieser Stelle), wenn die Stichprobe (sehr) groß ist.
- keine Rückweisung
- einfach zu Lernen, großer Speicher- und Rechenaufwand für die Klassifikation
Es gibt Methoden zum Ausdünnen der Stichprobe

mNN-Klassifikator

Modifikation, um Ausreißer oder Störungen an der Klassengrenzen besser zu Berücksichtigen:

- Entscheidung gemäß der Majorität der m nächsten Nachbarn
- Rückweisung, falls das Maximum von Vektoren aus einer Klasse kleiner als Schwelle $l < m$

Auch hier wird i.d.R. in Gebieten des Merkmalsraumes klassifiziert, die nicht in der Stichprobe/Problemkreis enthalten sind.

2.1.2 Effiziente Berechnung für NN-Klassifikatoren

- Partial Distance Technique

```
sei  $d(\vec{c}, \vec{c}_i) := \sum_{k=1}^N (c_k - c_{ik})^2$ 
```

```
 $d_{min} := d(\vec{c}, \vec{c}_1)$ 
```

```
 $minindex := 1$ 
```

```
FOR alle  $c_i$   $i = 2 \dots I$ 
```

```
   $dist := 0$ 
```

```
  FOR  $u = 1 \dots N$  solange  $dist < d_{min}$ 
```

```
     $dist := dist + (c_k - c_{ik})^2$ 
```

```
    IF  $dist < d_{min}$ 
```

```
      THEN  $d_{min} := dist$ 
```

```
         $minindex := i$ 
```

```
return  $minindex$ 
```

2.1 Abstandsmessung zur Klassifikation

- Fast Nearest-Neighbor Search

- Vorher erstellen: Tabelle $tabd$ mit Abständen zwischen den Vektoren der Stichprobe (enthält $\frac{I(I-2)}{2}$ Einträge).

```
 $d_{min} := d(\vec{c}, \vec{c}_1)$ 
```

```
 $minindex := 1$ 
```

```
FOR alle  $c_i$   $i = 2 \dots I$ 
```

```
IF  $tabd(minindex, i) > 2d_{min}$ 
```

```
THEN continue
```

```
IF  $d(\vec{c}, \vec{c}_i) < d_{min}$ 
```

```
THEN  $d_{min} := d(\vec{c}, \vec{c}_i)$   
 $minindex := i$ 
```

```
return  $minindex$ 
```

2.1.3 Klassifikation durch Vektorquantisierung

- Idee ist hier, Merkmalsvektoren im Zentrum der Klassengebiete zu finden.
- Einfachste Variante: bestimme den Schwerpunkt jeder Klasse.
(Das gibt nur sehr einfache, insbesondere zusammenhängende Klassengebiete.)
- Vektorquantisierung dient (eigentlich) der kompakten Repräsentation oder Codierung von (hochdimensionalen) Vektoren durch Repräsentanten, den Codebook-Vektoren.

2.1.4 k-Means

Teile jede Klasse Ω_{κ} in L_{κ} (Teil-)Gebiete, die jeweils durch ihren Schwerpunkt oder Mittelpunktvektor $\vec{\mu}_{\kappa}^l$ repräsentiert werden.

Im restlichen Abschnitt lassen wir die Klassenindizes κ zur Vereinfachung weg:

- sei $\Omega = \{\vec{c}_i | i = 1, \dots, N\}$ die (einzige) Klasse
- sie wird in L disjunkte Teilgebiete R^l zerlegt:
$$\Omega = R^1 \dot{\cup} R^2 \dot{\cup} \dots \dot{\cup} R^L$$
- jedes Teilgebiet R^l wird durch seinen Schwerpunkt repräsentiert:

$$\vec{\mu}^l = \frac{1}{|R^l|} \sum_{\vec{c}_i \in R^l} \vec{c}_i$$

- Ziel ist Minimierung des Quantisierungsfehlers

$$\epsilon = \frac{1}{N} \sum_{l=1}^L \sum_{\vec{c}_i \in R^l} d(\vec{c}_i, \vec{\mu}^l)$$

2.1 Abstandsmessung zur Klassifikation

wähle aufgrund von Vorwissen oder zufällig (z.B. die ersten L Vektoren der Stichprobe mit der Größe N) initiale Mittelpunkte $\vec{\mu}^l, l = 1, \dots, L$

$\epsilon^0 := \infty$; der Quantisierungsfehler in der Iteration 0

$t = 0$; Iterationszähler

$t := t + 1, \quad \epsilon^{(t)} := 0$

FOR alle Gebiete $l = 1, \dots, L$

$N^l := 0; \quad \hat{\vec{\mu}}^l := \vec{0}$

FOR alle Vektoren \vec{c}_i der Stichprobe

bestimme $\vec{\mu}^l$ mit minimalem Abstand zu \vec{c}_i

$\epsilon^{(t)} := \epsilon^{(t)} + d(\vec{c}_i, \vec{\mu}^l)$

berechne neuen Schätzwert für den Mittelpunkt $\hat{\vec{\mu}}^l := \vec{\mu}^l + \vec{c}_i$

$N^l := N^l + 1$

$\epsilon^{(t)} := \epsilon^{(t)} / N$

FOR alle Gebiete $l = 1, \dots, L$

$\vec{\mu}^l := \hat{\vec{\mu}}^l / N^l$

UNTIL $(\epsilon^{(t-1)} - \epsilon^{(t)}) / \epsilon^{(t)} \leq \epsilon$

2.1.5 LBG-Verfahren

bestimme Anzahl der Gebiete je Klasse mit dem LBG-Verfahren auf der Basis von K-means

Anzahl der Gebiete ist $L = 1$, wobei das Gebiet R_1 durch

$$\vec{\mu}_1 = \frac{1}{N} \sum_{i=1}^N \vec{c}_i \text{ repräsentiert wird}$$

FOR alle Gebiete R^l (oder für die \tilde{L} Gebiete mit größtem Quantisierungsfehler)

zerlege R^l in 2 Gebiete mit den Mittelpunkten $\vec{\mu}^l + \vec{\delta}$ und $\vec{\mu}^l - \vec{\delta}$ ($\vec{\delta}$ ist zufälliger betragsmäßig kleiner Vektor)

$$L := L * 2 \quad (L := L + \tilde{L})$$

wende k-means zur Optimierung von L Gebieten an

UNTIL Quantisierungsfehler $\epsilon^{(t)} \leq \epsilon$ ODER $L \geq$ maximale Klassenanzahl

Wenn die Mittelpunktvektoren gelernt sind, erfolgt die Klassifikation von \vec{c} nach Ω_{κ} , falls

$$\vec{\mu}_{\kappa}^l = \min_{\lambda, j} d(\vec{\mu}_{\lambda}^j, \vec{c})$$

2.1.6 Learning-Vector-Quantization

betrachte beim Lernen die gesamte Stichprobe (nicht Klassen unabhängig)

LVQ1-Algorithmus

wähle auf der Basis einer klassifizierten Stichprobe L Vektoren $\vec{\mu}_l$ als initiale Mittelpunkte, d.h. für jeden Vektor $\vec{\mu}_l$ ist die zugehörige Klasse bekannt (z.B. berechne je Klasse ω_κ L/K Mittelpunkte $\vec{\mu}^l$ mit Hilfe des k-means-Algorithmus aus Abschnitt ??)

Iterationszähler $t = 0$

wähle zufällig \vec{c} aus der Stichprobe aus

bestimme $\vec{\mu}^l$ mit kleinstem Abstand zu \vec{c}

IF $\vec{\mu}^l$ und \vec{c} sind aus derselben Klasse

THEN $\vec{\mu}^l := \vec{\mu}^l + \alpha(t)[\vec{c} - \vec{\mu}^l]$

ELSE $\vec{\mu}^l := \vec{\mu}^l - \alpha(t)[\vec{c} - \vec{\mu}^l]$

$t := t + 1$

UNTIL $t > \beta * L$

LVQ2-Algorithmus

initialisiere L Vektoren $\vec{\mu}_l$ (analog zu LVQ1)

Iterationszähler $t = 0$

wähle zufällig \vec{c}_i aus der Stichprobe aus

bestimme die beiden nächsten Nachbarn $\vec{\mu}_l$ und $\vec{\mu}_k$

IF $\vec{\mu}_l$ gehört zur richtigen Klasse UND $\vec{\mu}_k$ zur falschen Klasse UND

$$\min \left(\frac{m(\vec{c}_i, \vec{\mu}_l)}{m(\vec{c}_i, \vec{\mu}_k)}, \frac{m(\vec{c}_i, \vec{\mu}_k)}{m(\vec{c}_i, \vec{\mu}_l)} \right) > \frac{1-w}{1+w} \quad \text{mit} \quad 0.2 < w < 0.3$$

THEN $\vec{\mu}_l := \vec{\mu}_l + \alpha(t)[\vec{c}_i - \vec{\mu}_l]$

$$\vec{\mu}_k := \vec{\mu}_k - \alpha(t)[\vec{c}_i - \vec{\mu}_k]$$

$t := t + 1$

UNTIL $t > \beta * L$

LVQ3-Algorithmus

initialisiere L Vektoren $\vec{\mu}_l$ (analog zu LVQ1)	
Iterationszähler $t = 0$	
wähle zufällig \vec{c}_i aus der Stichprobe aus	
bestimme die beiden nächsten Nachbarn $\vec{\mu}_l$ und $\vec{\mu}_k$	
IF	$\vec{\mu}_l$ gehört zur richtigen Klasse UND $\vec{\mu}_k$ zur falschen Klasse UND $\min \left(\frac{m(\vec{c}_i, \vec{\mu}_l)}{m(\vec{c}_i, \vec{\mu}_k)}, \frac{m(\vec{c}_i, \vec{\mu}_k)}{m(\vec{c}_i, \vec{\mu}_l)} \right) > \frac{1-w}{1+w} \quad \text{mit} \quad 0.2 < w < 0.3$
THEN	$\vec{\mu}_l := \vec{\mu}_l + \alpha(t)[\vec{c}_i - \vec{\mu}_l]$ $\vec{\mu}_k := \vec{\mu}_k - \alpha(t)[\vec{c}_i - \vec{\mu}_k]$
IF	$\vec{\mu}_l$ und $\vec{\mu}_k$ gehören zur richtigen Klasse
THEN	$\vec{\mu}_l := \vec{\mu}_l + \varepsilon \alpha(t)[\vec{c}_i - \vec{\mu}_l]$ $\vec{\mu}_k := \vec{\mu}_k + \varepsilon \alpha(t)[\vec{c}_i - \vec{\mu}_k]$
$t := t + 1$	
UNTIL $t > \beta * L$	

mit $0.1 < \varepsilon < 0.3$