6 Mustererkennung bei Sprachsignalen

Erkennung gesprochener Sprache

6.1 Einleitung

Probleme bei der Erkennung gesprochener Sprache

- Vielzahl möglicher **Aussprachevarianten** ein und derselben Spracheinheit
  - verschiedene Sprecher
  - ein Sprecher je nach Kontext
- **Wortgrenzen** nur schwer zu finden
  - Die Sprache ist fließend und es gibt keine Pausen zwischen den Wörtern.
- **Mehrdeutigkeiten**
  - Mann man
  - mehr Meer
  - Rad Rat
  - bewusster leben bewusst erleben
- **Störungen**
  - lautes Hintergrundrauschen
  - Ventilatoren (kann durch einfache Filterung eliminiert werden)
  - Umgebungsgeräusche in einer Cafeteria (schwieriger)
- **Komplexität**
  - Die Suche nach der am besten zum Sprachsignal passenden Wortfolge wird schnell umfangreich.
- **Grammatikunterschiede** zur geschriebenen Sprache

Ziel

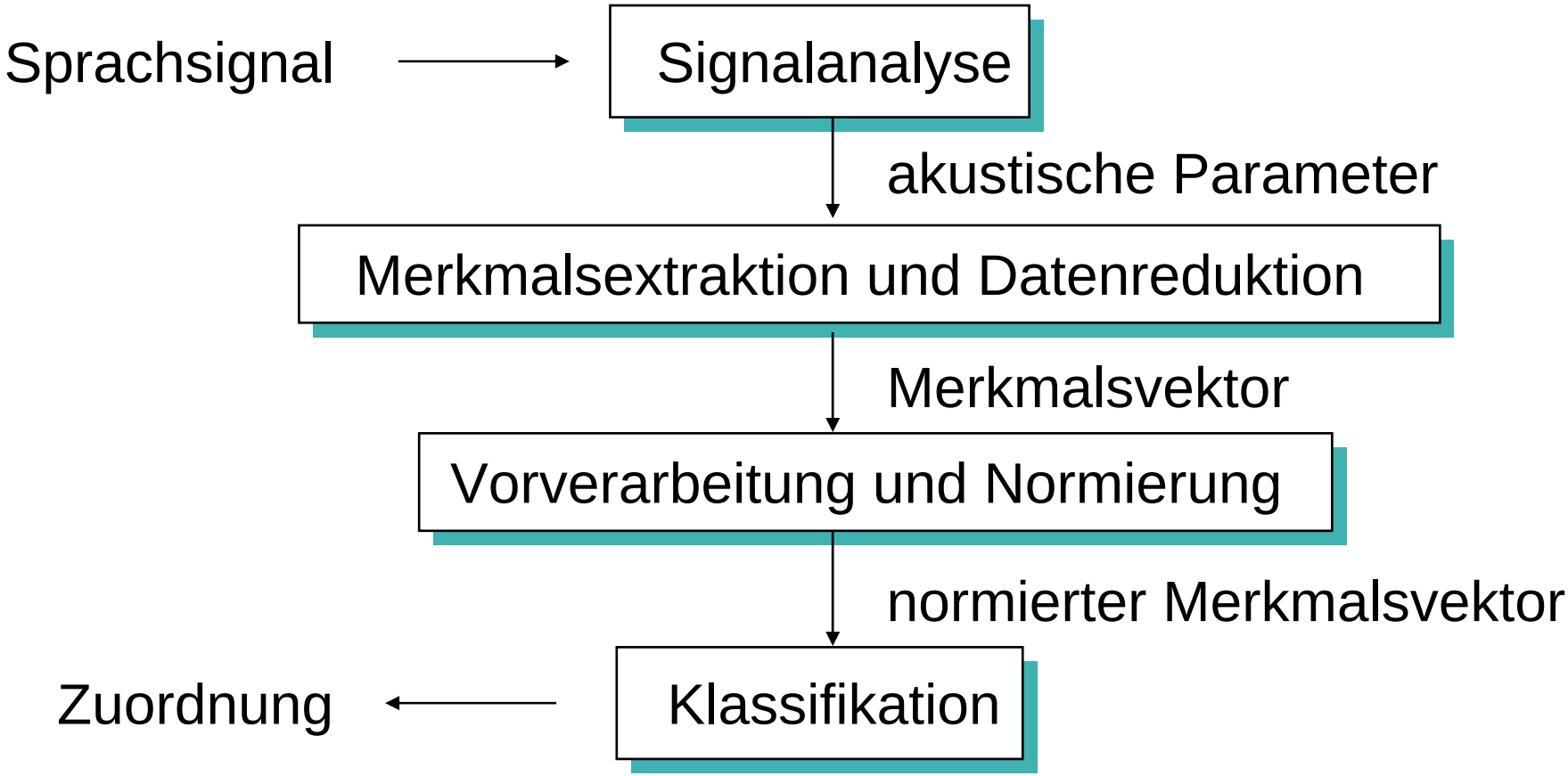
- Das Ziel der Mustererkennung im Rahmen der Sprachverarbeitung besteht darin, ein kontinuierliches Signal in Abschnitte einzuteilen, die einer bestimmten Klasse bzw. einem bestimmten Muster zugeordnet werden können.
- Klassen bzw. Muster können u.a. repräsentieren:
  - Phoneme
  - Wortformen
  - Wortformfolge
  - Personen (bei der Sprechererkennung)

Eigenschaften von Spracherkennungssystemen

- Einzelworterkennung
- Wortkettenerkennung
- Erkennung fließender Sprache

- sprecherabhängig
- sprecherunabhängig

Mustererkennung



(Merkmalsvektor - Klasse = Phonem, Wortform, Sprecher)

6.2 Klassifikation (von Sprachmustern)

Methoden zur Klassifikation

- lineare Klassifikation
- Abstandsklassifikatoren
- statistische Klassifikation
- …
- nichtlineare Zeitanpassung
- Hidden - Markov - Modelle

Klassifikation

Gegeben: Menge von **Klassen**, die aus **bekannten Mustern** bestehen

Aufgabe: neues Muster → Klasse

6.2.1 Numerische Klassifikation

Bezeichnungen

Merkmalsvektor: $\mathbf{x} = (x_1, x_2, ..., x_N) \in R^N$

Menge der Klassen, wobei jede Klasse aus einem oder mehreren Mustern besteht $M = \{K_1, K_2, \dots K_L\}$

$K_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{iN_i}\}$ $\mathbf{x}_{ij} = (x_1^{ij}, x_2^{ij}, ..., x_N^{ij})$

Klassifikator: $c : \{\mathbf{x}\} \rightarrow \{K_i : i = 1, ..., L\}$

Beispiel

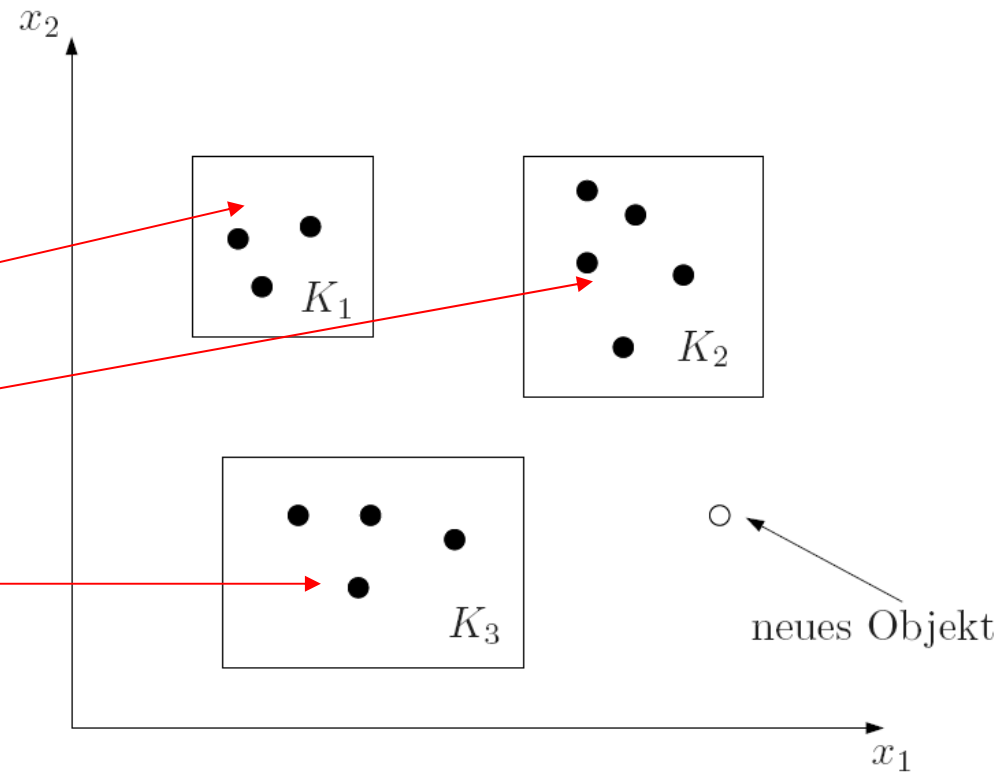
$N = 2$

$L = 3$

$N_1 = 3$

$N_2 = 5$

$N_3 = 4$



Lineare Klassifikation

$$\mathbf{x} = (x_1, x_2, \ldots, x_N) \in R^N$$

Entscheidungsfunktionen: $e_i : R^N \rightarrow R, \quad i = 1, \ldots, L$

$$e_i(\mathbf{x}) = w_0^j + w_1^j x_1 + w_2^j x_2 + \ldots + w_N^j x_N, \quad w_j^j \in R, \quad j = 0, \ldots, N$$

$e_i(\mathbf{x}) = 0$ beschreibt eine Hyperebene im R^N

Für einen linearen Klassifikator gilt:

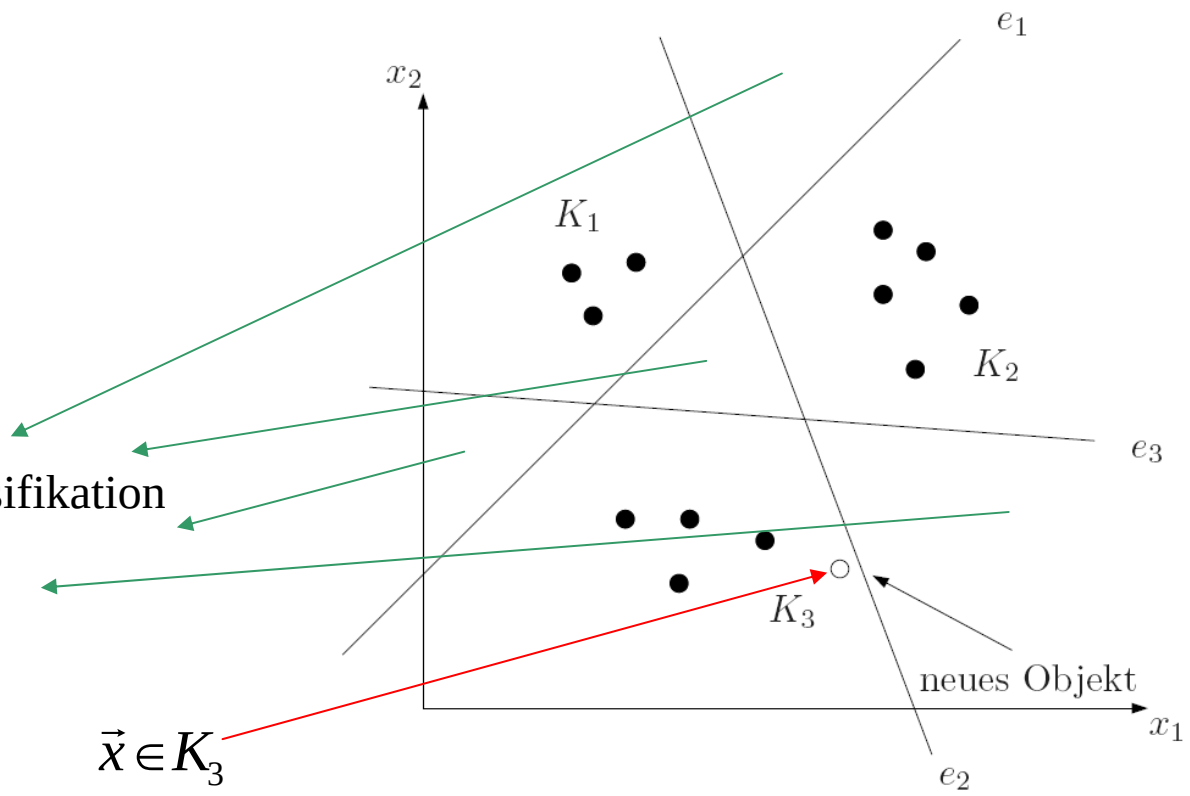
$$c(\mathbf{x}) = K_i \leftrightarrow (e_i(\mathbf{x}) > 0) \wedge (e_j(x) < 0, \quad \textit{für alle} \quad j = 1, \ldots, L, \quad j \neq i)$$

Beispiel

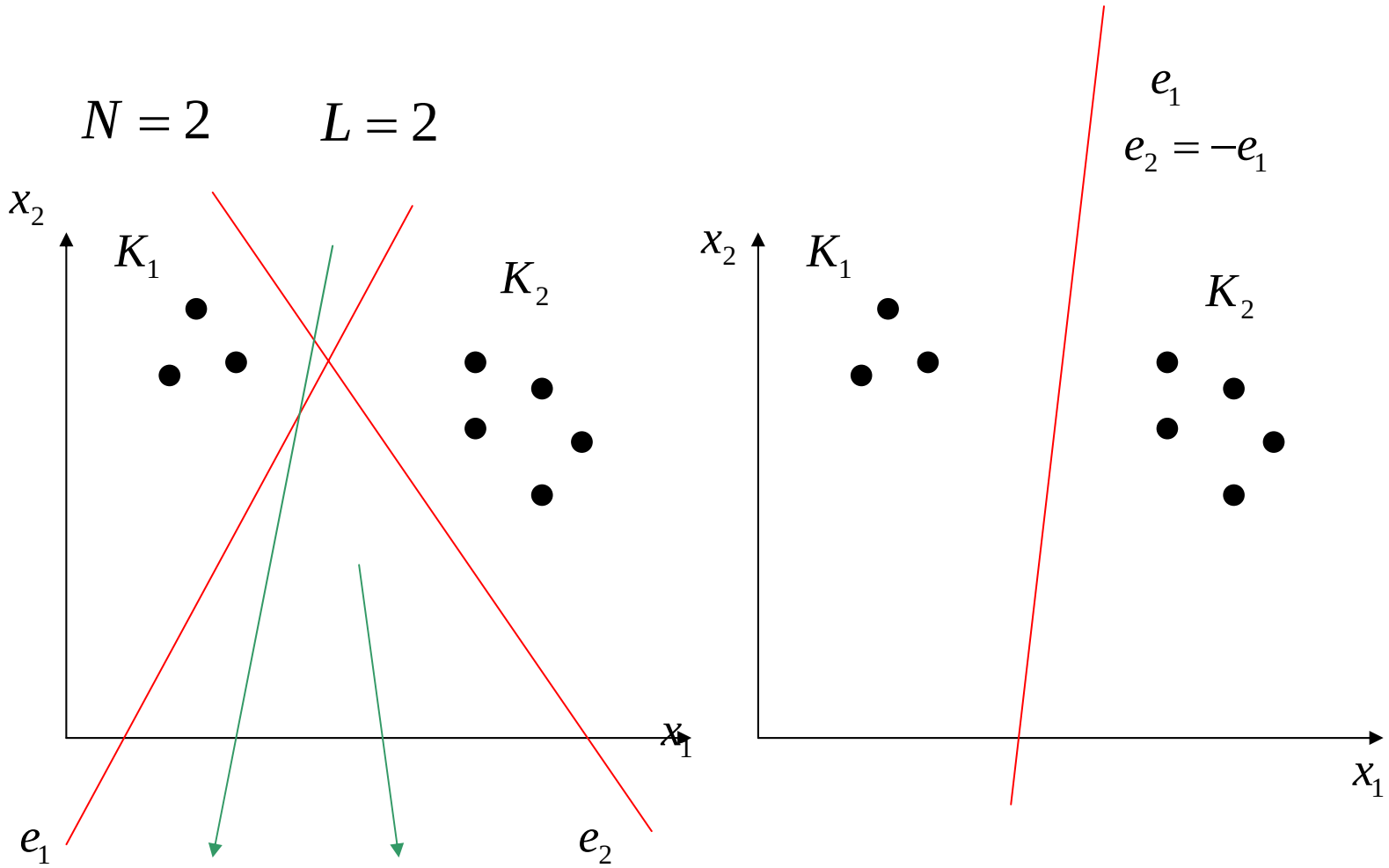
$N = 2$

$L = 3$

hier ist keine Klassifikation möglich



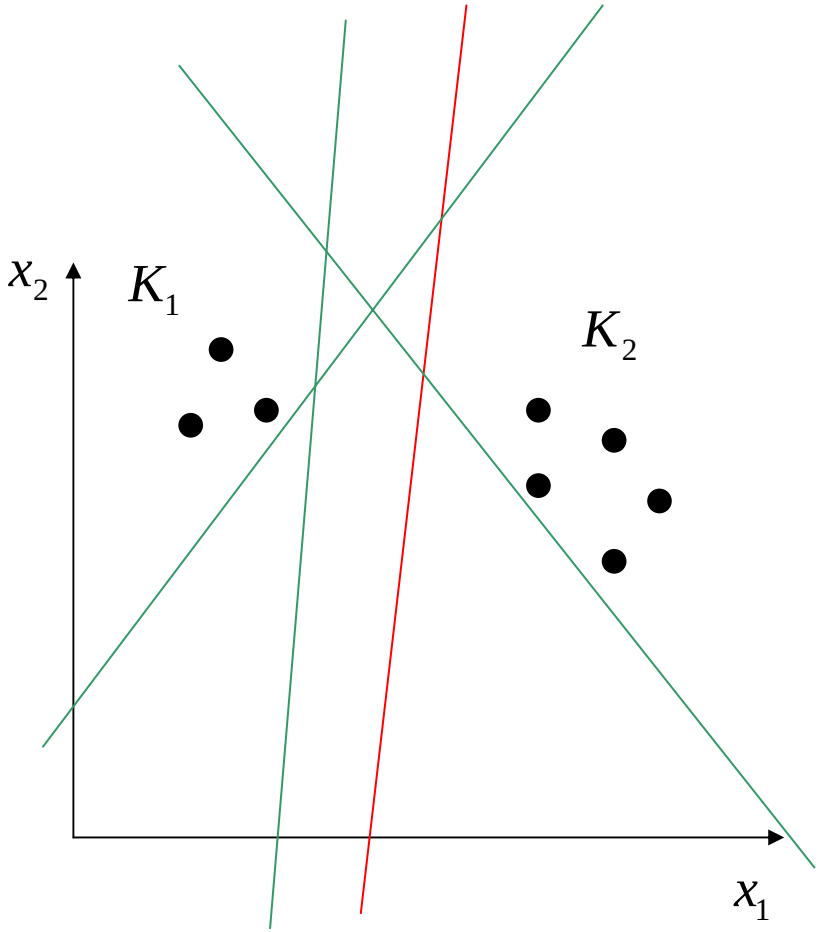
Beispiel



hier ist keine Klassifikation möglich

eine trennende Gerade
Klassifikation immer möglich

Beispiel



Problem:

Welches ist die beste trennende Gerade?

Support Vector Machines

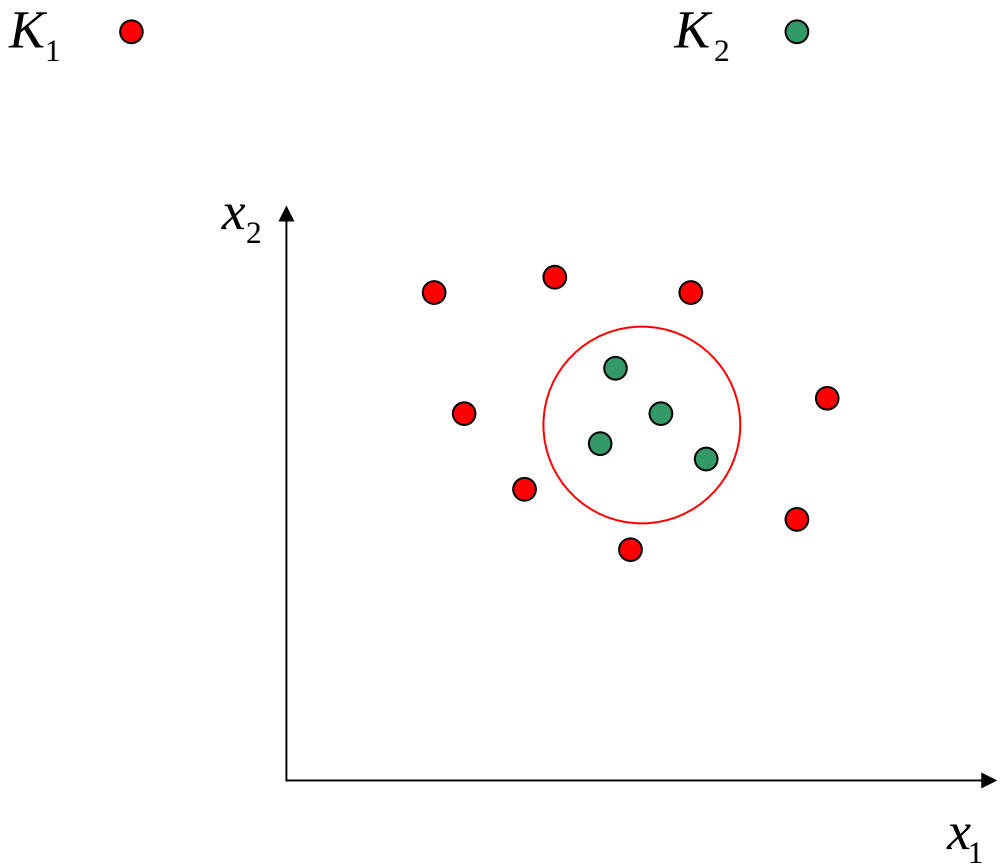
Bemerkungen

$$\vec{x}_{pq} \in K_p \quad \Rightarrow \quad e_i(\vec{x}_{pq}) \begin{cases} > 0 & \text{falls } p = i \\ < 0 & \text{falls } p \neq i \end{cases}$$

$$q = 1, \ldots, N_p$$

- Die Festlegung der Entscheidungsfunktionen geschieht in einer Lernphase und ist nicht immer möglich.
- Anstelle der Hyperebenen können auch kompliziertere Flächen betrachtet werden.

Beispiel – nichtlineare Trennung



Minimum – Distance – Klassifikator

$$c(\mathbf{x}) = K_m \leftrightarrow m = \underset{i=1,\ldots,L}{\operatorname{argmin}} \left\{ \sum_{n=1}^{N} (x_n - f_n^i)^2 \right\}$$

$$\mathbf{f}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{x}_{ij} = (f_1^i, \cdots, f_n^i)$$

Mittelwertsvektor der Klasse K_i

Nearest – Neighbour – Klassifikator

$$c(\mathbf{x}) = K_m \leftrightarrow \exists m' \in \{1,\ldots,N_m\}, \quad mit:$$

$$\sum_{k=1}^{N}(x_k - x_k^{mm'})^2 = \min_{i=1,\ldots,L;\, j=1,\ldots,N_i} \left\{ \sum_{k=1}^{N}(x_k - x_k^{ij})^2 \right\}$$

6.2.2 Statistische Klassifikation

Wahrscheinlichkeiten

$P(\mathbf{x}|i)$ bedingte Wahrscheinlichkeit, dass ein zur Klasse K_i gehörendes Objekt den Merkmalsvektor **x** liefert

$P(i|\mathbf{x})$ bedingte Wahrscheinlichkeit für das Vorliegen einer Klasse K_i unter der Bedingung, dass der Merkmalsvektor **x** vorliegt

P_i Wahrscheinlichkeit des Auftretens eines Objektes der Klasse K_i

c_{pq} Kosten (Bestrafung) für den Fall, dass ein Objekt der Klasse K_p fälschlicherweise der Klasse K_q zugeordnet wird

Bayes – Klassifikator

$$c(\mathbf{x}) = K_m \leftrightarrow m = \underset{k=1,\ldots,L}{\operatorname{argmin}} \{d_k(\mathbf{x})\}$$

$$d_k(\mathbf{x}) = \sum_{l=1}^{L} c_{kl} p_l p(\mathbf{x} \mid l)$$

Spezialfall:

$$c_{pq} = \begin{cases} 0 \text{ falls } p = q \\ 1 \text{ falls } p \neq q \end{cases}$$

$$\sum_{l=1}^{L} c_{kl} \cdot p_l \cdot p(\mathbf{x} \mid l) = \sum_{l=1, l \neq k}^{L} p_l \cdot p(\mathbf{x} \mid l) = \sum_{l=1}^{L} p_l \cdot p(\mathbf{x} \mid l) - p_k \cdot p(\mathbf{x} \mid k)$$

Konstante

zu maximieren

Maximum – Likelihood – Klassifikator

$$c(\mathbf{x}) = K_m \leftrightarrow m = \underset{k=1,\ldots,L}{\operatorname{argmax}} \{ p(\mathbf{x} \mid k) p_k \}$$

Bayes

$$p(k \mid \mathbf{x}) = \frac{p_k \cdot p(\mathbf{x} \mid k)}{p(\mathbf{x})}$$

$$c(\mathbf{x}) = K_m \leftrightarrow m = \underset{k=1,\ldots,L}{\operatorname{argmax}} \{ p(k \mid \mathbf{x}) \}$$

Problem

Spracherkennung

$$\hat{W} = \underset{W \in V^*}{\operatorname{argmax}} P(W \mid \mathbf{X})$$

optimale Wortfolge

$\hat{W} = w_1 w_2 \ldots w_K \quad w_i \in V$

Wortfolge

Merkmalssequenz

$\mathbf{X} = \mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_T$

$$\hat{W} = \underset{W \in V^*}{\operatorname{argmax}} P(\mathbf{X} \mid W) \cdot P(W)$$

akustisches Modell
z.B.: HMM

Sprachmodell

Merkmalssequenzen in der Spracherkennung

- Merkmalssequenzen variieren in zweifacher Hinsicht
  - Länge der Merkmalssequenz
  - Wert des Merkmals

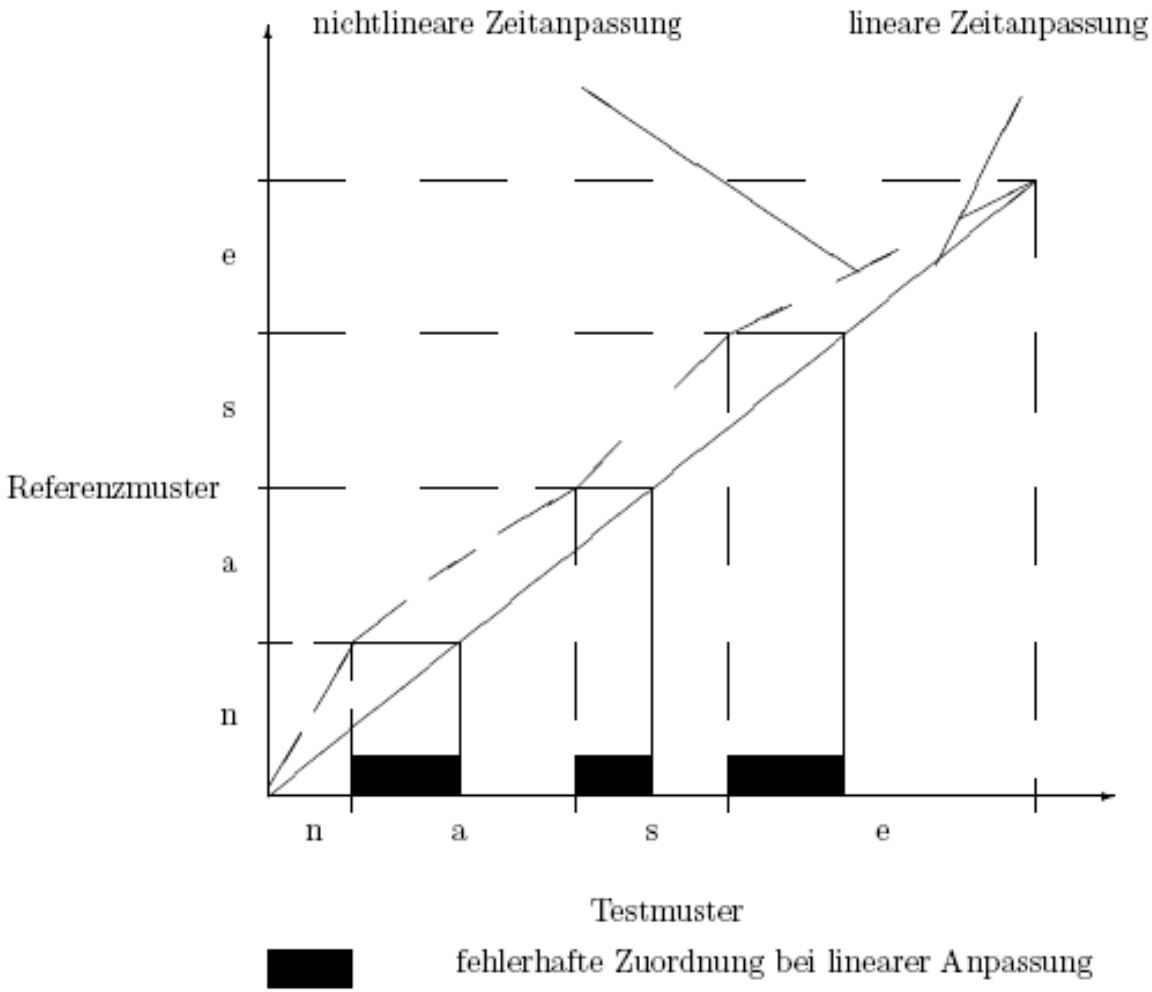
6.3 Nichtlineare Zeitanpassung

6.3.1 Einzelworterkennung

Problem

- Eine Schwierigkeit bei der Spracherkennung besteht darin, z. B. zwei gleiche Wortformen, die zu verschiedenen Zeitpunkten und eventuell von verschiedenen Sprechern ausgesprochen wurden und sich daher bezüglich des zeitlichen Aufbaus unterscheiden, aneinander so anzupassen, dass mit einer Abstandsmessung die (erwartete) Gleichheit erkannt wird.
- **Lineare Zeitanpassung**
  - Die Zeitachsen der beiden Sprachmuster werden als Ganzes gedehnt oder gestaucht, so dass beide Sprachmuster dieselbe Länge haben.
- **Nichtlineare Zeitanpassung**
  - Hier werden Teile innerhalb eines Musters gedehnt, andere gestaucht oder unverändert gelassen.

Problem


nichtlineare Zeitanpassung
lineare Zeitanpassung
e
s
Referenzmuster
a
n
n
a
s
e
Testmuster
fehlerhafte Zuordnung bei linearer Anpassung

Formale Beschreibung

$$\mathbf{P} = \mathbf{p}(1), \mathbf{p}(2), \ldots, \mathbf{p}(I)$$

Folge von Merkmalsvektoren, die ein Testmuster beschreiben

$$\mathbf{Q} = \mathbf{q}(1), \mathbf{q}(2), \ldots, \mathbf{q}(J)$$

Beschreibung eines Referenzmusters (Klasse)

P und **Q** haben i.a. unterschiedliche Länge.

$$\mathbf{p}(i) = \left(p_1(i), p_2(i), \ldots, p_N(i)\right)$$

$$\mathbf{q}(j) = \left(q_1(j), q_2(j), \ldots, q_N(j)\right)$$

Optimierungsproblem

Gesucht:

$$W = w(1), w(2), \ldots, w(l), \ldots, w(L)$$

$$w(l) = \big(i(l), j(l)\big) \qquad i(l) \in \{1, \ldots, I\} \qquad j(l) \in \{1, \ldots, J\}$$

$$w(l_1) \neq w(l_2) \quad l_1 \neq l_2 \qquad w(1) = (1,1) \qquad w(L) = (I, J)$$

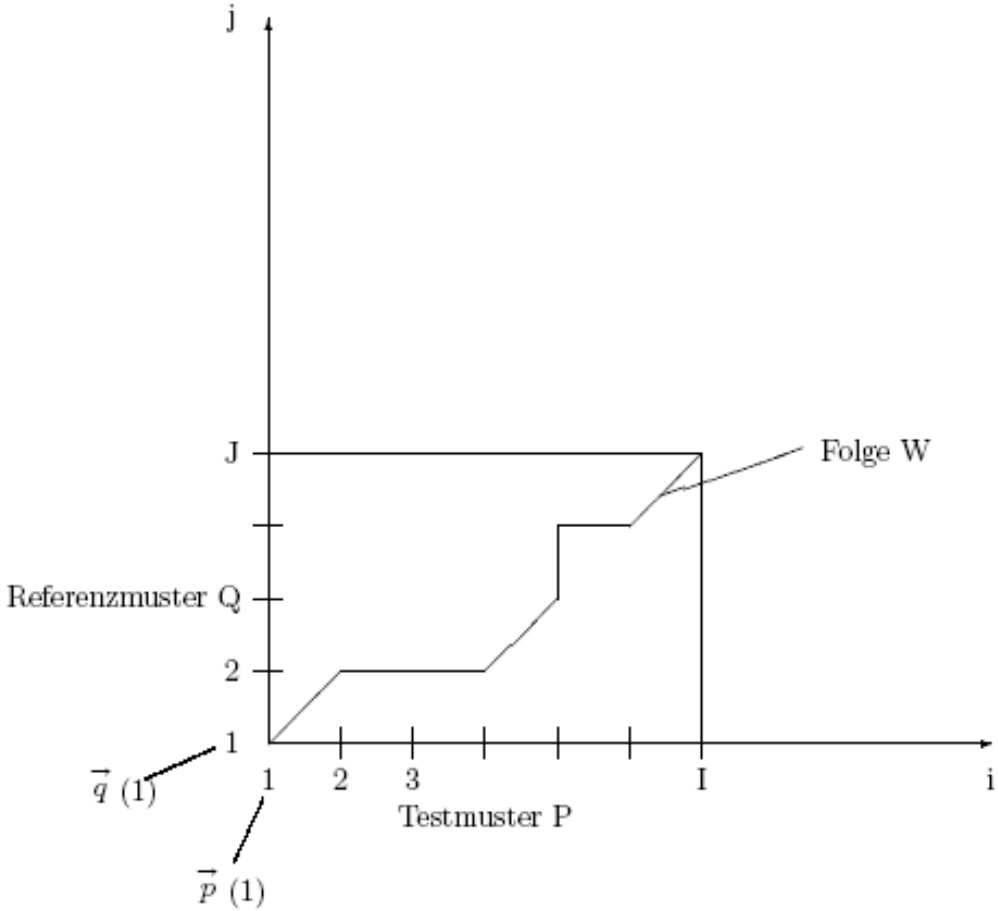
$$i(l-1) \leq i(l) \leq i(l-1) + 1$$

$$j(l-1) \leq j(l) \leq j(l-1) + 1$$

$$\sum_{l=1}^{L} d(w(l)) \rightarrow \min$$

$$d((i, j)) = \sum_{k=1}^{N} \big(p_k(i) - q_k(j)\big)^2$$

Folge W


j
J
Referenzmuster Q
2
1
Folge W
$\overrightarrow{q}$ (1)
1
2
3
I
i
Testmuster P
$\overrightarrow{p}$ (1)

Bemerkungen

- Die Folge *W* hat die Aufgabe, gleiche akustische Ereignisse beider Sprachmuster möglichst optimal einander zuzuordnen.
- Der Punkt (i,j) kann von den Punkten (i -1,j), (i -1,j -1), (i,j -1) erreicht werden.
- enthält der Weg W einen horizontalen Abschnitt von (i -1,j) nach (i,j), so bedeutet dies, dass ein Merkmalsvektor (j) des Referenzmusters auf zwei aufeinanderfolgende Merkmalsvektoren (i -1,i) des Testmusters abgebildet wird
- enthält der Weg W einen vertikalen Abschnitt von (i,j - 1) nach (i,j), so bedeutet dies, dass zwei aufeinanderfolgende Merkmalsvektoren (j -1, j) des Referenzmusters auf einen Merkmalsvektor (i) des Testmusters abgebildet wird

Bemerkungen

- Zeitanpassung heißt in den letzten beiden Fällen, dass an bestimmten Stellen Lautereignisse eingefügt werden. Das angepasste Sprachmuster ist also stets länger als das ursprüngliche Sprachmuster (kein Lautereignis wird ausgelassen)
- verläuft der Weg W von (i -1, j -1) nach (i,j) (also diagonal), so bedeutet dies, dass zwei aufeinanderfolgende Abschnitte beider Muster zugeordnet werden können (keine Zeitanpassung)

Lösung – Dynamische Optimierung

Das Optimierungsproblem lässt sich gut durch Dynamische Optimierung lösen. Dazu führen wir eine Partialsumme D(i, j) ein, die als Summe der Abstände längs des optimalen Weges W von Anfangspunkt (1,1) zu einem beliebigen Punkt (i,j) definiert ist.

$$D(1,1) = d((1,1))$$

$$D(i,\ j) = d((i,\ j)) + \min\left\{D(i-1,\ j), D(i,\ j-1), D(i-1,\ j-1)\right\}$$

$$D(i,\ j) = \infty \quad i \notin \left\{1, \ldots, I\right\} \text{ oder } j \notin \left\{1, \ldots, J\right\}$$

Mit Hilfe dieser Rekursionsgleichung ist es möglich die Summe D(I,J) zu berechnen. Damit erhält man auch die optimale Folge *W*.

Mustererkennung

Vom Testmuster *P* wird mit allen Referenzmustern die Partialsumme *D*(*I,J*) berechnet. Das Referenzmuster, mit dem sich der kleinste Ergebniswert ergibt, wird dem Testmuster zugeordnet. Eventuell werden auch mehrere Referenzmuster zugeordnet (*D* unterscheidet sich wenig) oder keins (falls *D* zu groß).

Beispiel

...									
8									
10									
10,2									
10,1									
9,9									
10,1									
10,5									
0,1	0,0								
	0,1	10	9,5	10	9,9	10	8	6	...

$\mathbf{q}(1) = q_1(1)$

$\mathbf{p}(1) = p_1(1)$

$D(1,1) = |0.1 - 0.1|$

$d((i, j)) = |p_1(i) - q_1(j)|$

Beispiel

...									
8									
10									
10,2									
10,1									
9,9									
10,1									
10,5	↕ 10,4	↗ 0,5							
0,1	0	→ 9,9							
	0,1	10	9,5	10	9,9	10	8	6	...

$$D(1,2) = d(1,2) + D(1,1)$$

$$D(2,1) = d(2,1) + D(1,1)$$

$$D(2,2) = d(2,2) + \min\{D(1,1), D(2,1), D(1,2)\}$$

Beispiel

...									
8									
10									
10,2									
10,1									
9,9	↑ 30,2	↑ 0,7	↗ 1,0	→ 1,1					
10,1	↑ 20,4	↑ 0,6	↗ 1,1	→ 1,2					
10,5	↑ 10,4	↗ 0,5	→ 1,5	→ 2,0					
0,1	0,0	→ 9,9	→ 19,3	→ 29,2					
	0,1	10	9,5	10	9,9	10	8	6	...

$$D(2,3) = d(2,3) + \min\{D(1,2), D(2,2), D(1,2)\}$$

Beispiel

0,1		12,7	12,7	12,7	↗ **2,8**
9,8		2,8	2,8	↗ **2,8**	12,3
10		**2,6**	→ **2,6**	2,6	12,5
10,3		2,8	3,0	3,2	13,3
...					
	...	10	10	10	0,1

Beispiel

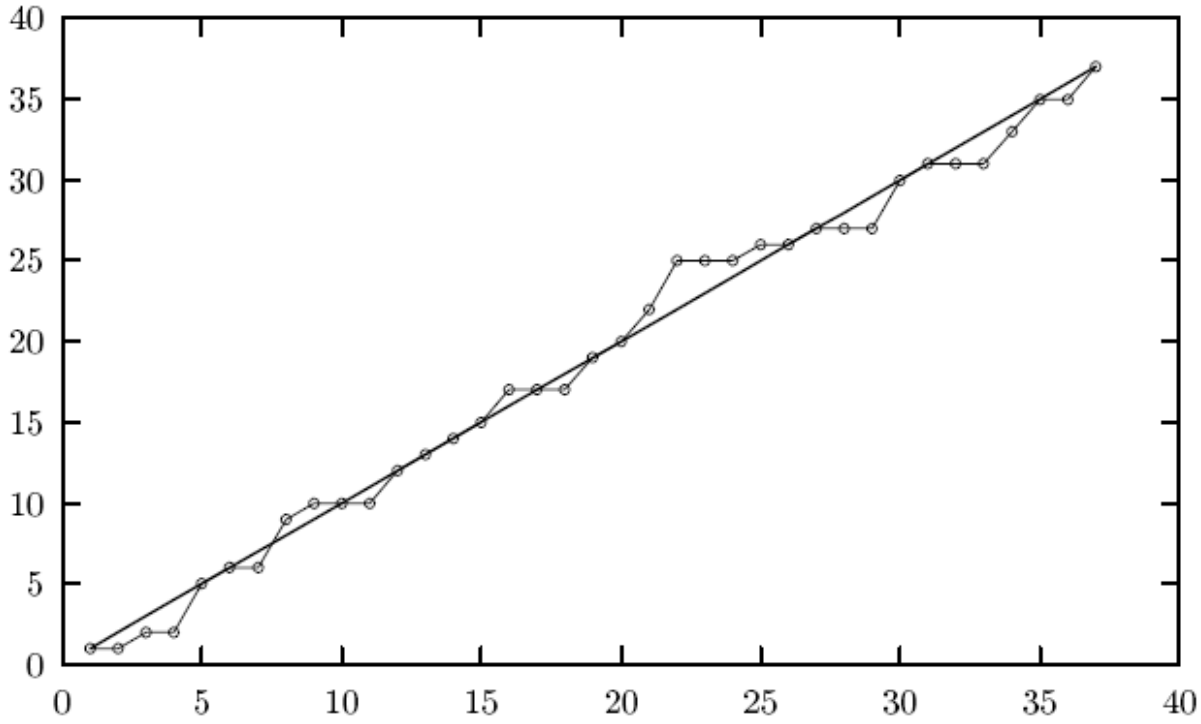


Abbildung 5.2: Optimaler Pfad für den Vergleich zweier Äußerungen des Wortes *Donau* von einem Sprecher

Beispiel

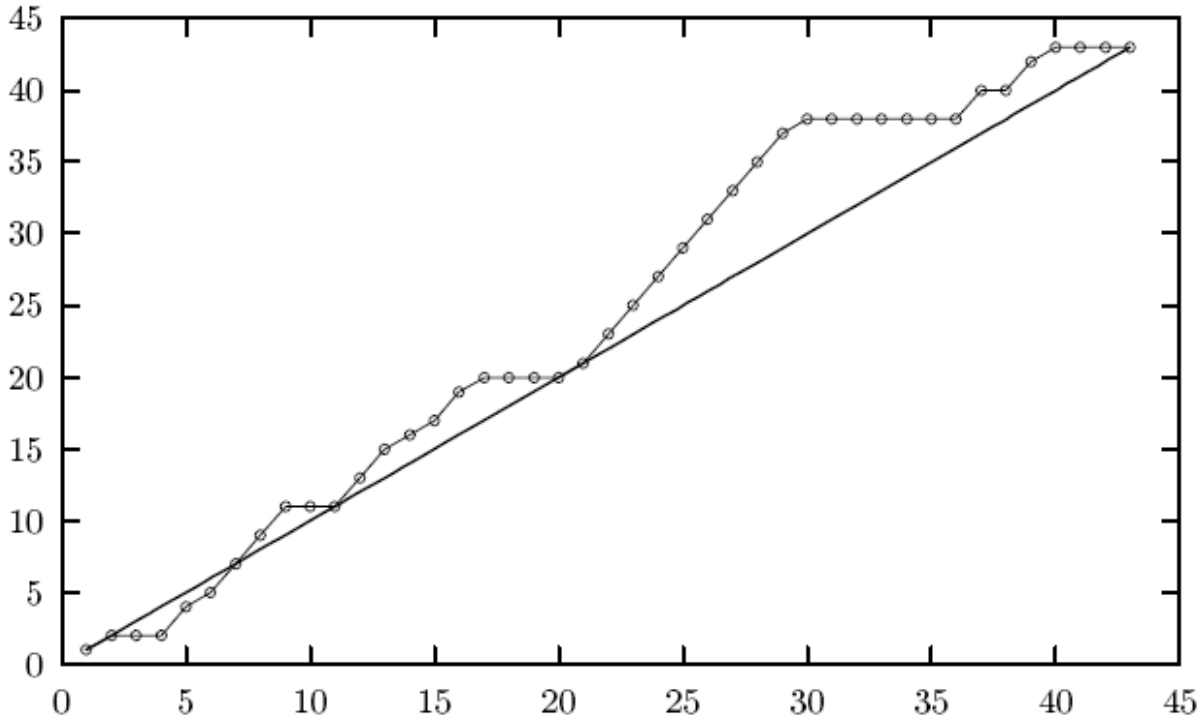


Abbildung 5.3: Optimaler Pfad für den Vergleich zweier Äußerungen des Wortes *Donau* von zwei Sprechern

6.3.2 Word spoting

Word spoting

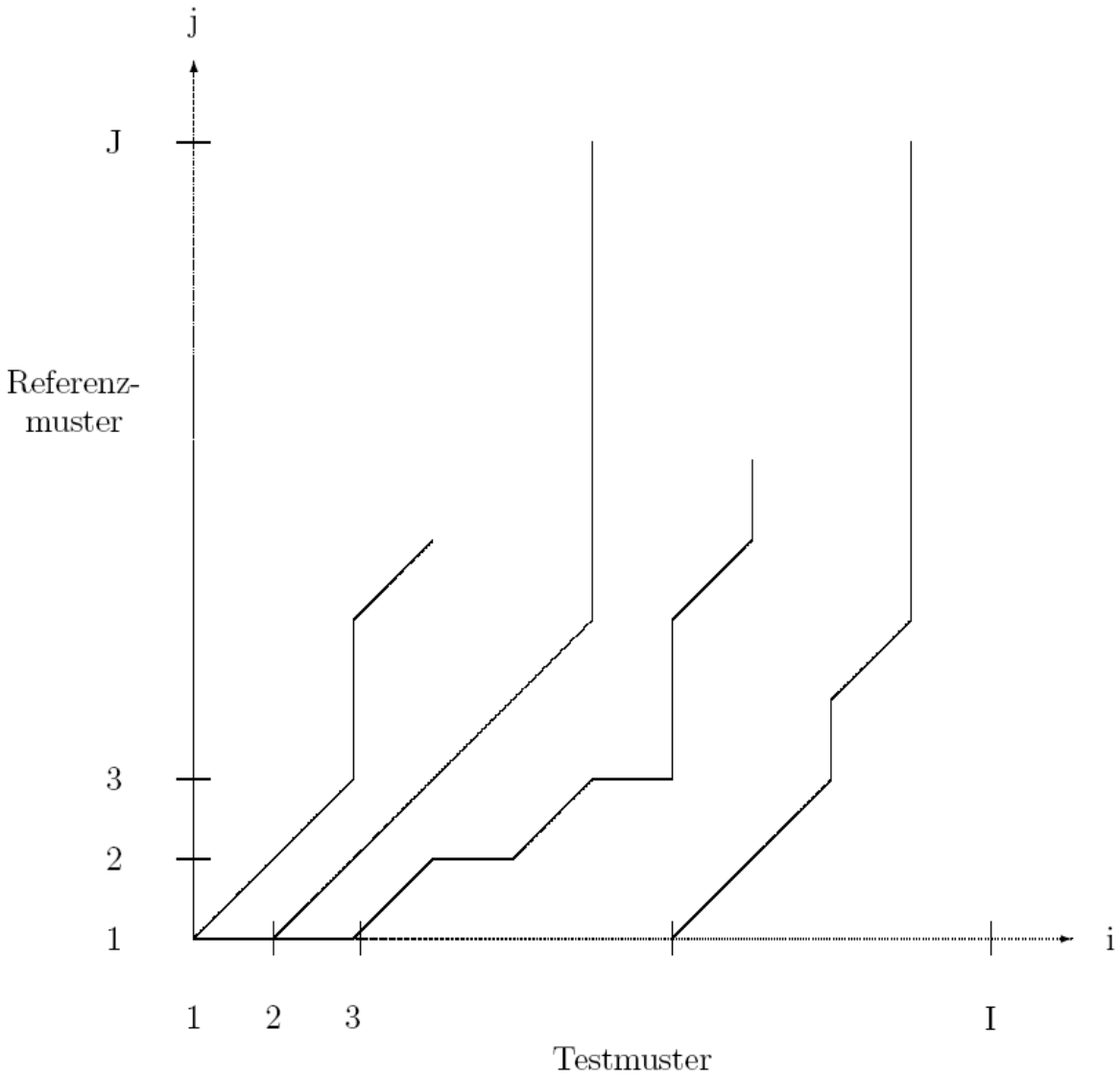
- hier wird ein langes Sprachsignal daraufhin untersucht, ob ein vorliegendes Wort (Referenzmuster) in diesem Sprachsignal enthalten ist
- Drei Probleme sind dabei zu lösen:
  - die Bestimmung der Wortgrenzen
  - die nichtlineare Zeitanpassung
  - die eigentliche Erkennung

Lösungsschritte

- Jeder Zeitpunkt *i* des Testmusters muss für den Beginn eines Wortes in Betracht gezogen werden.
- Wir betrachten deshalb mehrere Wege *W* bei der nichtlinearen Zeitanpassung.
- Die Wege *W* beginnen bei einem Punkt (*i,*1)

  i = 1, 2, … (erste Zeile) und enden in einem Punkt (i_1,*J*) i_1 = 1, 2,… der obersten Zeile.
- Es werden wieder die Summen *D*(i_1,*J*) berechnet.
- Ein Weg kann abgebrochen werden, wenn die Summe *D* zu groß wird.
- Falls mehrere Wege vollendet werden, so wird derjenige mit der kleinsten Summe *D* ausgewählt.

Word spoting



6.3.3 Wortkettenerkennung

Testmuster und Referenzwortschatz

$$\mathbf{P} = \mathbf{p}(1), \mathbf{p}(2), \ldots, \mathbf{p}(I)$$

Folge von Merkmalsvektoren, die ein Testmuster beschreiben

$$\mathbf{p}(i) = \left(p_1(i), p_2(i), \ldots, p_N(i)\right)$$

Referenzwortschatz:

$$\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_L$$

L Referenzmuster (Wörter)
J(*l*) sei die Anzahl der Zeitsegmente des Wortes *l* aus den Referenzwortschatz
(*l* = 1, 2,…,*L*)

$$\mathbf{Q}_l = \mathbf{q}(1,l), \mathbf{q}(2,l), \ldots, \mathbf{q}(J(l),l)$$

$$\mathbf{q}(j,l) = \left(q_1(j,l), q_2(j,l), \ldots, q_N(j,l)\right)$$

Lösung

$$W = w(1), w(2), \ldots, w(m), \ldots, w(M)$$

$$w(m) = \big(i(m),\, j(m), l(m)\big)$$

$$i(m) \in \{1, \ldots, I\}$$

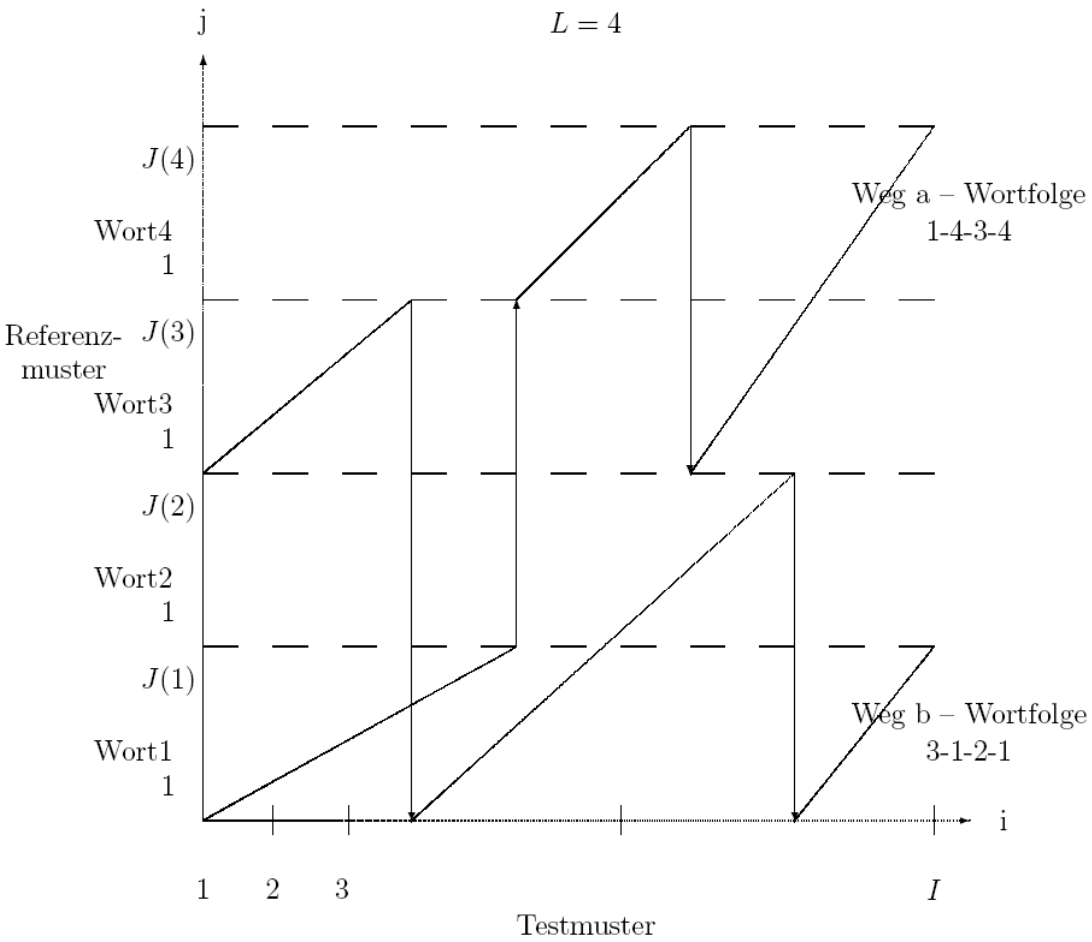
$$j(m) \in \{1, \ldots, J(l(m))\}$$

$$l(m) \in \{1, \ldots, L\}$$

Weg in der Punktmenge

$$\{(i, j, l):\ i = 1, \ldots, I \quad j = 1, \ldots, J(l) \quad l = 1, \ldots, L\}$$

Lösung



Wir müssen alle Kombinationen der *L* Referenzmuster betrachten, wobei ein Muster auch mehrmals auftreten kann.

Einschränkungen an *W*

Anfang: $w(1) = (1,1,l(1))$ $l(1) \in \{1,\ldots,L\}$

Ende: $w(M) = (I, J(l(M)), l(M))$ $l(M) \in \{1,\ldots,L\}$

Übergangsregeln im Innern eines Referenzmusters:

Der Vorgänger des Punktes (i, j, l) $i > 1$ $j > 1$

ist einer der drei Punkte $(i-1, j, l)$

$(i-1, j-1, l)$

$(i, j-1, l)$

Einschränkungen an *W*

Übergangsregeln an den Grenzen der Referenzmuster:

Der Vorgänger des Punktes $(i,1,l)\ i>1$

ist einer der Punkte $(i-1, J(l^*), l^*)\ l^* \in \{1,\ldots,L\}$

$$(i-1,1,l)$$

Optimierungsbedingung:

$$\sum_{m=1}^{M} d(w(m)) \rightarrow \min$$

$$d((i,j,l)) = \sum_{k=1}^{N} \left(p_k(i) - q_k(j,l)\right)^2$$

Lösung des Optimierungsproblems

Wieder führen wir eine Partialsumme D(i,j,l) ein, die als Summe der Abstände längs des optimalen Weges W von Anfangspunkt w(1) zu einem beliebigen Punkt (i,j,l) definiert ist.

$$D(1,1,l(1)) = d((1,1,l(1)))$$

$$D(i,\ j,l) = d((i,\ j,l)) + \min\{D(i-1,\ j,l), D(i,\ j-1,l), D(i-1,\ j-1,l)\}$$

falls $i > 1 \quad j > 1$

$$D(i,1,l) = d((i,1,l)) + \min\{D(i-1,1,l), \{D(i-1, J(l^*), l^*) : l^* \in \{1,\ldots,L\}\}\}$$

falls $i > 1$

Mit Hilfe dieser Rekursionsgleichung ist es möglich die Summe D(I,J(l(M),l(M)) zu berechnen. Damit erhält man auch die optimale Folge *W*.