Mustererkennung in der Sprachverarbeitung - Lernhilfe

Stefan Kugele

5. Oktober 2003

Inhaltsverzeichnis

1 Grundprobleme

Ein Grundproblem ist die Segmentierung, d.h. die Einteilung der Sprachsignale in kleinste Einheiten.
Wegen der zusammenhängenden Sprachlaute während der Aussprache (*Koartikulation*) werden die einzelnen Laute stark von ihren Nachbarn beeinflusst.

1.1 Segmentierungsmöglichkeiten

Eine günstige Wahl ist die Segmentierung in *eme*.

In der deutschen Sprache gibt es in etwa 20 Konsonanten und auch 20 Vokale (mit Diphthongen).

1.2 Die deutschen Phoneme

Phoneme sind kleinste linguistische Grundeinheiten, deren Austausch ein Bedeutungsunterschied eines Wortes zur Folge hat. Ein Beispiel wären die Worte *le*b*en* und *le*g*en*. Eine akustische Realisierung eines Phonems nennt man *Phon*. Die Menge aller akustischen Realisierungen eines Phonems bezeichnet man mit *Allophon*.

Vokale und Diphthonge werden durch periodische Anregungen der Stimmbänder erzeugt. *Konsonanten* hingegen werden an einer Engstelle des *Vokaltrakts* artikuliert. Man nennt diesen Ort *Artikulationsstelle*.

2 Spektrale Darstellung

Ein *Sonagramm* ist die 2-dimensionale Darstellung der spektralen Intensität in einer Zeit-Frequenz-Ebene.

Die Darstellung kann aus einer *Filterbank* gewonnen werden, die aus vielen überlappenden Bandfiltern besteht. Die Analysebreite beträgt 300 Hz pro Bandfilter.

2.1 Kurzzeitspektren

Sprachsignale sind *nicht-stationäre* Vorgänge. Daher kann ein Sprachsignal nur in einem sehr kurzen Zeitabschnitt als hinreichend stationär betrachtet werden.

Komplexes *Kurzzeitamplitudenspektrum*:

$$F(f,t) = \int_{-\infty}^{\infty} f(\tau) \cdot w(t-\tau) \cdot e^{-j2\pi ft} d\tau$$

Multiplikation mit dem Drehzeiger $e^{j2\pi ft}$ liefert:

$$f(t) \longrightarrow \boxed{w(t) \cdot e^{j2\pi ft}} \longrightarrow F(f,t) \cdot e^{j2\pi ft}$$

2.2 Wahl der Fensterfunktion

Wichtig ist die Wahl der Fensterfunktion. Im Frequenzbereich sollte sie möglichst schmal sein, ideal ein *Rechteckfenster* und im Zeitbereich schnell abklingen. Besonders verbreitet und geeignet ist das sogenannte HAMMING-Window:

$$w(t) =$$

$$\begin{cases} 0.54 + 0.46 \cos\left(\frac{2\pi t}{T}\right) & \text{für } -\frac{T}{2} \le t \le \frac{T}{2} \\ 0 & \text{sonst} \end{cases}$$

Weiche Fenster, damit bei der kontinuierlichen Fortsetzung keine harten Sprungstellen entstehen.

2.3 Digitale Signaldarstellung

Abtastung des Analogsignals an äquidistanten Zeitpunkten $n\Delta t$.

Hierbei muss immer das *Abtasttheorem* erfüllt sein:

$$f_{tast} = \frac{1}{\Delta t} \qquad f_{tast} > 2 \cdot f_{grenz}$$

$$\begin{aligned} \text{Zeitsignal: } s(n) &= s(n\Delta t) \\ \text{Spektrum: } S(m) &= S(m\Delta f) \\ S(m) &= \sum_{n=0}^{N-1} s(n) \cdot e^{\frac{-2j\pi mn}{N}} \\ s(n) &= \frac{1}{N} \sum_{m=0}^{N-1} S(m) \cdot e^{\frac{2j\pi mn}{N}} \end{aligned}$$

mit $\Delta f = \dfrac{1}{N \cdot \Delta t}$

Das diskrete Spektrum ist ein *Linienspektrum*, zu dem nach den Gesetzen der Fouriertransformation eine *periodische* Zeitfunktion gehört. Da die Zeitfunktion abgetastet wurde, ist das Spektrum periodisch fortzusetzen.

Für das diskrete *verschobene* HAMMING-Fenster gilt:

$$h(k) = 0,54 - 0,46 \cdot \cos\left(\frac{2\pi k}{N}\right)$$

für $0 \leq k \leq N-1$.

Das Kurzzeitsprektrum zum Zeitpunkt n ist:

$$S_n(m) = \sum_{k=0}^{N-1} s(k) \cdot h(k) \cdot e^{\frac{-2j\pi mk}{N}}$$

3 Grundlagen der Mustererkennung

Betrags- und Leitungsspektren sind phasenunabhängig.

$$\overset{Signal}{\longrightarrow} \boxed{\text{Wandelung}} \overset{Messdaten}{\longrightarrow} \boxed{\text{Vorver.}}$$

$$\overset{Muster\ \underline{x}}{\longrightarrow} \overset{Klassifikator}{\boxed{\text{Entsch.fkt.}}} \longrightarrow Klasse$$

Aufgabe der Vorverarbeitung ist die Auswahl von wichtigen bzw. die Reduktion von irrelevanten Merkmalen.

Der *Merkmalsvektor* $\underline{x}$ hat das folgende Aussehen:

$$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ n_N \end{pmatrix} = (x_1, x_2, \ldots, x_n)^T$$

Wobei N die Anzahl der Dimensionen und K die Anzahl der Klassen ist.

3.1 Klassifikator

Klassifikator: $\underline{x} \longrightarrow k$

Man unterscheidet zwischen einem *entscheidungstheoretischen* und einem *strukturalistischen* Ansatz. Die *Entscheidungsfunktion* wird mit $d(\underline{x})$ bezeichnet.

3.2 Lernvorgang

In einer Lernphase sollen Entscheidungsfunktionen selbsttätig optimiert werden können.
Man unterscheidet zwischen

- fest eingestellten Systemen und
- adaptiven Systemen

Weiter kann ein System *überwacht*, also mit einem Lehrer, oder *nicht-überwacht* sein.

3.3 Fehlerrate

Die *mittlere Fehlerrate* lässt sich mit Hilfe der Quellenstatistik abgeschätzt und minnimiert werden.

4 Klassifikation anhand von Entscheidungsfunktionen

Die Muster einer Klasse sind immer nur ähnlich und unterliegen einer gewissen Streuung da:

(a) eigentlich konstante Muster vorliegen, aber zusätzlich eine Streuung bei der Produktion statt findet

(b) konstante Muster, aber eine Unschärfe bei deren Messung

Beides liegt bei der Sprachverarbeitunf vor. Im Merkmalsraum liegen daher „Ballungen“ oder „*Cluster*“ vor.

4.1 Linearer Klassifikator

Entscheidungsfunktionen können durch einen positiven oder hohen Wert die Zugehörigkeit zu einer Klasse anzeigen oder die Rolle einer *Trennfunktion* spielen.

Für 2 Dimensionen ($N = 2$) gilt folgendes:

$$d(\underline{x}) = w_0 + w_1 x_1 + w_2 x_2$$

$\underline{x} \in \omega_1$ falls $d(\underline{x}) > 0$ und $\underline{x} \in \omega_2$ falls $d(\underline{x}) < 0$. Die Trenngerade hat die Form: $w_0 + w_1 x_1 + w_2 x_2 = 0$

Dies kann auf N Dimensionen ausgeweitet werden, wobei dann $d(\underline{x}) = 0$ *Hyperebenen* als Trennflächen sind.

$$d(\underline{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N$$

$$= w_0 + \sum_{n=1}^{N} w_n x_n = w_0 + \underline{w}^T \underline{x}$$

In der *erweiterten Form*

$$\underline{x} = (1, x_1, x_2, \ldots, x_N)^T$$

$$\underline{w} = (w_0, w_1, w_2, \ldots, w_N)^T$$

$$d(\underline{x}) = \sum_{n=0}^{N} w_n x_n = \underline{w}^T \underline{x}$$

4.1.1 Erweiterung auf mehrere Dimensionen

Hierbei werden 3 Fälle unterschieden:

Fall 1 Jede Klasse soll von allen übrigen Klassen getrennt (separiert) werden:

$$d_i(\underline{x}) = \underline{w}_i^T \underline{x} \text{ für } i = 1 \ldots K$$

$$\underline{x} \in \omega_i \text{ wenn } d_i(\underline{x}) > 0$$

$$\wedge \forall j = 1 \ldots K, j \neq i : d_j(\underline{x}) < 0$$

Dieser Fall ist sehr streng und geht nur für bereits Gelerntes.

$$\underline{d}(\underline{x}) = \begin{pmatrix} d_1 \\ \vdots \\ d_{i-1} \\ d_i \\ d_{i+1} \\ \vdots \\ d_K \end{pmatrix} = \begin{pmatrix} < 0 \\ \vdots \\ < 0 \\ > 0 \\ < 0 \\ \vdots \\ < 0 \end{pmatrix}$$

Fall 2 Jede Klasse soll von jeder anderen Klasse paarweise separiert werden.
Entscheidungsfunktion $d_{ij}(\underline{x})$ trennt Klasse i von Klasse j

$$d_{ij}(\underline{x}) = \underline{w}_{ij}^T \underline{x}$$

$$\underline{x} \in \omega_i \text{ wenn } d_{ij}(\underline{x}) > 0 \; \forall j = 1 \ldots K, j \neq i$$

$$\underline{w}_{ij} = (w_{0ij}, w_{1ij}, \ldots, w_{Nij})^T$$

$$d_{ij}(\underline{x}) = -d_{ij}(\underline{x})$$

Es gibt also $\frac{K(K-1)}{2}$ Entscheidungsfunktionen.

$$\underline{D} = (d_{ij}) = \begin{pmatrix} & & * & & \\ > 0 & > 0 & \ldots & > 0 & > 0 \\ & & * & & \end{pmatrix}$$

Fall 3 Ein Muster gehört zur Klasse i, wenn die Entscheiduzngsfunktion $d_i(\underline{x})$ einen größeren Wert hat, als alle anderen Entscheidungsfunktionen. Es gibt somit K Entscheidungsfunktionen.

$$d_i(\underline{x}) = \underline{w}^T \underline{x}$$

$$\underline{x} \in \omega_i \text{ wenn } d_i(\underline{x}) > d_j(\underline{x}) \forall j = 1 \ldots K, j \neq i$$

Trennfunktion: $d_i(\underline{x}) - d_j(\underline{x}) = 0$.
In diesem Fall entstehen keine undefinierten Gebiete, wie es bei den Fällen 1 und 2 passieren kann.

$$\underline{d}(\underline{x}) = \begin{pmatrix} d_1 \\ \vdots \\ d_i \\ \vdots \\ d_K \end{pmatrix}$$

$$\underline{x} \in \omega_i \text{ wenn } d_i(\underline{x}) = \max_K(d_j)$$

$\underline{d}(\underline{x})$ wird auch als Schätzvektor bezeichnet.

Wenn eine Musterkonfiguration nach Fall 3 trennbar ist, so ist sie auch nach Fall 2 trennbar.

4.2 Polynomklassifikatoren

Eine einfache Verallgemeinerung linearer Entscheidungsfunktionen erhält man durch einen *Polynomansatz*.

Quadratische Entscheidungsfunktionen für 2-dimensionale Muster ($N = 2$)

$$\begin{aligned} d(\underline{x}) &= w_0 + w_1 x_1 + w_2 x_2 + \\ &\quad w_{12} x_1 x_2 + w_{11} x_1^2 + w_{22} x_2^2 \end{aligned}$$

Bei N Dimensionen

$$\begin{aligned} d(\underline{x}) &= w_0 + \sum_{j=1}^{N} w_j x_j + \\ &\quad \sum_{j=1}^{N-1} \sum_{k=j+1}^{N} w_{jk} x_j x_k + \sum_{j=1}^{N} w_{jj} x_j^2 \end{aligned}$$

Diese Gleichung hat T Terme:

$$T = \binom{N+G}{G} = \binom{N+G}{N} = \frac{(N+G)!}{N!G!}$$

4.3 Generalisierte Entscheidungsfunktionen

Für kompliziertere Gebietsaufteilungen kann der Lineare Klassifikator verallgemeinert werden, indem beliebige, einwertige, reelle Funktionen von $\underline{x}$ verwendet werden.

$$d(\underline{x}) = w_0 + w_1 f_1(\underline{x}) + w_2 f_2(\underline{x}) + \ldots + w_R f_R(\underline{x})$$

$$d(\underline{x}) = \sum_{r=0}^{R} w_r f_r(\underline{x}) \quad \text{mit } f_0(\underline{x}) = 1$$

Für die Klasse i gilt:

$$d_i(\underline{x}) = \sum_{r=0}^{R} w_{ir} f_r(\underline{x}) \quad \text{mit } f_0(\underline{x}) = 1$$

Transformierter Merkmalsvektor $\underline{x}^*$

$$\underline{x}^* = \begin{pmatrix} 1 \\ f_1(\underline{x}) \\ \vdots \\ f_R(\underline{x}) \end{pmatrix}$$

$$d(\underline{x}) = \underline{w}^T \underline{x}^*$$

Im transormierten Merkmalsraum $\underline{x}^*$ findet man wieder den Linearen Klassifikator vor. Wichtig ist, dass auch allgemeine Entscheidungsfunktionen auf einen Linearen Klassifikator zurückgehührt werden können.

4.4 Trennbarkeit des Linearen Klassifikators (K=2)

Einteilung in 2 Klassen mei M Mustern, die durch N Merkmale bestimmt sind.

Für M Muster gobt es 2^M Klasseneinteilungen. Die Wahrscheinlichkeit, dass eine willkürliche Klasseneinteilung mit dem Linearen Klassifikator realisiert werden kann beträgt:

$$p = \frac{\text{trennbare}}{\text{mögliche}}$$

$$p = \begin{cases} 2^{(1-M)} \sum_{i=0}^{N} \binom{M-1}{i} & \text{für } M \geq N+1 \\ 1 & \text{für } M \leq N+1 \end{cases}$$

Wenn p klein, also unwahrscheinlich zu trennen ist, aber dennoch getrennt werden konnte, so liegt keine *global position* vor, sondern eine spezielle Verteilung (*Cluster*) vor.

Das *Trennvermögen* wird mit

$$2(N+1) \approx 2N$$

berechnet, wobei N die Zahl der Freiheitsgrade beträgt.

Die Trennbarkeit kann erhöht werden, indem die Zahl der Dimensionen erhöht wird. Es dürfen jedoch keine linearen Abhängigkeiten entstehen, da sonst die Punkte auf einer *Hyperebene* liegen.

5 Abstands-Klassifikatoren

Das Muster $\underline{x}$ soll dann zur Klasse i gehören, wenn es zu dieser Klasse im Merkmalsraum den geringsten geometrischen Abstand hat.

5.1 Minimum-Abstands-Klassifikator

Im einfachsten Fall wird für die Abstandsmessung ein einziges Muster, der so genannte *Prototyp* herangezogen. Für die Klasse i wird der Prototyp $\underline{m}_i$ durch Mittelung aller vorhandener Muster gewonnen:

$$\underline{m}_i = \frac{1}{M_i} \sum_{k=1}^{M_i} \underline{x}_{ki}$$

M_i ist die Anzahl der Muster der Klasse i.

Der quadratische EUKLID'sche Abstand ϱ_i eines unbekannten Musters $\underline{x}$ zu $\underline{m}_i$ berechnet sich nach:

$$\begin{aligned} d_i(\underline{x}) &= \underline{m}_i^T \underline{m}_i - 2\underline{m}_i^T \underline{x} \\ &= \sum_{n=1}^{N} m_{ni}^2 - 2\sum_{n=1}^{N} m_{ni} x_n \end{aligned}$$

Der Klassifikator benötigt einen *Minimum-Detektor*.

Für mehrere Klassen $i = 1 \ldots K$ gilt:

$$\underline{x} \in \omega_i \text{ wenn } d_i(\underline{x}) < d_j(\underline{x})$$

$$\forall j = 1 \ldots K, j \neq i$$

$$\underline{d}(\underline{x}) = \begin{pmatrix} d_1 \\ \vdots \\ d_{i-1} \\ d_i \\ d_{i+1} \\ \vdots \\ d_k \end{pmatrix} = \begin{pmatrix} * \\ \vdots \\ * \\ min \\ * \\ \vdots \\ * \end{pmatrix}$$

Die Trenngerade ist die Mittelsenkrechte zwischen den Prototypen.

5.1.1 Nächster-Nachbar-Klassifikator

Ein beliebiges Muster $\underline{x}$ wird klassifiziert, indem unter den gespeicherten Prototypen der nächste Nachbar gesucht wird, d.h. derjenige Prototyp, der den kleinsten Abstand zu dem unbekannten Muster hat.

Der kleinste EUKLID'sche Abstand ϱ_i^2 zu allen M_i Prototypen $\underline{z}_{ri}$ einer Klasse i ist:

$$\varrho_i^2 = \min_r (\underline{x} - \underline{z}_{ri})^T (\underline{x} - \underline{z}_{ri}) \quad r = 1 \ldots M_i$$

$$d_i(\underline{x}) = \min_r \left(\underline{z}_{ri}^T \underline{z}_{ri} - 2\underline{z}_{ri}^T \underline{x} \right) \quad r = 1 \ldots M_i$$

$$\underline{x} \in \omega_i \text{ wenn } d_i(\underline{x}) < d_j(\underline{x})$$

$$\forall j = 1 \ldots K, j \neq i$$

Es ergeben sich *stückweise lineare* Trennfunktionen. Es sind somit auch komplizierte Gebietsaufteilungen möglich. Ein Nachteil ist, dass zu viele Prototypen gespeichert und auch berechnet werden müssen.

5.1.2 k-Nächster-Nachbar-Regel

Die *k-Nächste-Nachbar-Regel* kommt den optimalen Klassifikator (BAYESschen Klassifikator) noch näher. Das Risiko berechnet sich zu:

$$2R_{Bayes} \geq R_{1-NN} \geq R_{3-NN} \geq \ldots$$

$$\ldots \geq R_{k-NN} \geq R_{Bayes}$$

5.2 Gewichtetet Euklidscher Abstand

Bei der Nächsten-Nachbar-Regel könnten falsche Zuordnungen durch ungöünstige Skalierungen entstehen.

Der sogenannte „Gewichtete EUKLIDsche Abstand“ eines Musters $\underline{x}$ zu einem Prototypem $\underline{z}$ ist:

$$\varrho^2 = \sum_{n=1} N g_g^2 (x_n - z_n)^2 = (\underline{x} - \underline{z})^T G (\underline{x} - \underline{z})$$

mit der Gewichtsmatrix:

$$\underline{G} = \begin{pmatrix} g_1^2 & & 0 \\ & \ddots & \\ 0 & & g_n^2 \end{pmatrix}$$

Festlegung der Gewichte:
Steuung σ_n der Merkmale x_n (bzw. Varianz σ_n^2) ist gleich:

$$\underline{G} = \begin{pmatrix} \frac{1}{\sigma_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_n^2} \end{pmatrix}$$

5.3 Mahalanobis-Abstand

Das Abstandsmaß wird so verallgemeinert, dass ungleiche Streuungen in der hauptachsen der Musterverteilungen ausgeglichen werden. Die Gewichtung wird auf der Grundlage der inversen *Kovarianzmatrix* durchgeführt.

Gegeben sei der Mittelwert $\underline{m}$:

$$\underline{m} = \mathbb{E}[\underline{x}] = \frac{1}{M} \sum_{k=1}^{M} \underline{x}_k$$

Die Kovarianzmatrix ist definiert als:

$$\underline{C} = \mathbb{E}[(\underline{x} - \underline{m})(\underline{x} - \underline{m})^T]$$

Nach Umformung:

$$\underline{C} = \mathbb{E}[\underline{x}\underline{x}^T] - \underline{m}\underline{m}^T$$

Empirisch berechnet sich die Kovarianzmatrix zu:

$$\underline{C} = \frac{1}{M} \left(\sum_{k=1}^{M} \underline{x}_k \underline{x}_k^T \right) - \underline{m}\underline{m}^T$$

Die Kovarianzmatrix ist *reell, symmetrisch,* und *positiv definit.*

$$\underline{C} = \sigma_{\mu\nu} = \begin{pmatrix} \boxed{\sigma_{11}} & \sigma_{12} & \dots & \sigma_{1N} \\ \sigma_{21} & \boxed{\sigma_{22}} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1} & \dots & \dots & \boxed{\sigma_{NN}} \end{pmatrix}$$

Die Diagonalelemente $\boxed{\sigma_{ii}}$ entsprechen den Varianzen. Die nicht-diagonal Elemente σ_{ij} sind die Kovarianzen. Die N *Eigenvektoren* der Kovarianzmatrix $\underline{C}$ zeigen die Richtungen der Hauptachsen der Verteilung an, während die Eigenwerte $\lambda_n \;\; n = 1 \dots N$ gleich der Varianzen entlang der Hauptachsen entsprechen.

Wird die inverse Kovarianzmatrix $\underline{C}^{-1}$ als Gewichtsmatrix verwendet, so ergibt sich der sogenannte MAHALANOBIS-*Abstand* zu einem Mittelpunksvektor $\underline{m}$ mit:

$$\varrho^2 = (\underline{x} - \underline{m})^T \underline{C}^{-1} (\underline{x} - \underline{m})$$

Die Konturen gleichen Abstand sind *schiefliegende Ellipsen.*

5.3.1 Mahalanobis-Abstand-Klassifikator

Für jede Klasse i wird eine inverse Kovarianzmatrix bestimmt. Der quadratische Abstand ϱ_i^2 zum Mittelpunkt $\underline{m}_i$ der Klasse i berechnet sich zu:

$$\varrho_i^2 = (\underline{x} - \underline{m}_i)^T \underline{C}^{-1} (\underline{x} - \underline{m}_i)$$

$$\underline{x} \in \omega_i \text{ wenn } \varrho_i^2 < \varrho_j^2 \; \forall j = 1 \dots K, j \neq i$$