Automatische Spracherkennung

2 Grundlagen: Mustererkennung

Soweit vorhanden ist der jeweils englische Fachbegriff, so wie er in der Fachliteratur verwendet wird, in Klammern angegeben.

Beispiele von Computerkommandos und Skizzenanweisungen sind in `Schreibmaschinenschrift` wiedergegeben.

Wiederholung: Spracherkenner als Blockschaltbild:

Grundprinzipien der Mustererkennung (pattern match)

Ganz allgemein versteht man unter Mustererkennung die Aufgabe, aus dem Signal einer bestimmten physikalischen Quelle eine Kategorisierung vorzunehmen.

In der Sprecher-/Sprach-Erkennung: Ein Sprachsignal wird einer bestimmten Kategorie (Laut, Silbe, Wort, Äußerung, Person) zugeordnet.

D.h. der Input des Mustererkenners ist immer eine physikalische Funktion, Output ist entweder eine symbolische Information (Kategorie) oder eine Gewichtung aller möglichen Ergebnisse (vgl. den Term $p(s|W)$ im statistischen Ansatz der Spracherkennung).

Prinzipielle Vorgehensweise der Mustererkennung in der ASR

(Fast) immer wird das Sprachsignal in Merkmale umgerechnet, eventuell vektorquantisiert (Datenreduktion, Generalisierung) und anschließend mit gespeichertem Wissen verglichen. Deshalb auch der englische Begriff 'pattern match' (= Mustervergleich). Die Vergleichswerte werden anschließend durch eine sog. Entscheidungsfunktion ausgewertet.

Skizze

```
Inputsignal  Mustererkennung  Kategorien
```

Beispiele für Mustererkennung

- Spracherkennung: Input Sprachsignal, Output Text
- Sprechererkennung: Input Sprachsignal, Output Person
- Sprecherverifikation: Input Sprachsignal + Name, Output Yes/No
- Handschrifterkennung: Input Schrift-Graphik, Output Text
- Unterschrifterkennung: Input Unterschriftgraphik + Name, Output Yes/No

Mustervergleich und Entscheidungsfunktion

Mustererkennung zerfällt (fast) immer in zwei Verarbeitungs- stufen: Mustervergleich/-bewertung und Entscheidungsfunktion.

Skizze

```
Blockschaltbild Vergleich + Entscheidung
```

a) Mustervergleich/-bewertung

Der Mustervergleich/-bewertung führt immer zu kontinuierlichen Werten für mehrere Kategorien. Z.B. liefern zwei HMM, welche verschiedene Kategorien repräsentieren (z.B. Wörter) zum selben Input verschiedene Wahrscheinlichkeiten, dass dieses Inputsignal von diesen Kategorien produziert

wurde.
Im Allgemeinen erhält man also für jede mögliche Kategorie (Wortliste, Phonemliste etc.) einen kontinuierlichen Wert auf einer Skala.

Skizze

```
Mustervergleich führt zu kontinuierlichem Abstandsmessungen
```

b) Entscheidungsfunktion

Die Entscheidungsfunktion legt fest, wie die Werte des Mustervergleichs interpretiert werden und beeinflusst damit entscheidend die Art und Weise der Mustererkennung.

Beispiel:

Ein Algorithmus liefert zu einem geg. Sprachsignal für 6 Vokalkategorien Werte zwischen 0 und 1 je ähnlicher das Sprachsignal der betreffenden Kategorie war.

Skizze

```
A 0.17   O 0.88
E 0.05   U 0.09
I 0.56   @ 0.34
```

Die Entscheidungsfunktion könnte z.B. sich für die Kategorie mit dem maximalen Wert entscheiden (Max-Entscheidung).

Das bedeutet hier die Entscheidung für Kategorie O

Wenn wir jedoch andere Vergleichswerte haben wie z.B.:

```
A 0.17   O 0.88
E 0.85   U 0.09
I 0.86   @ 0.87
```

führt die Max-Entscheidung zwar zum selben Ergebnis, aber bei genauer Betrachtung ist das nur ein Zufallsereignis, weil die Werte von E,I,O und @ sehr eng beieinander liegen.

Eine intelligentere Entscheidungsregel wäre in diesem Fall:
Entscheidung für die Kategorie, deren Wert deutlich über dem Mittelwert der anderen Kategorien liegt.
Wenn keine Kategorie dies erfüllt, wie hier in unserem Beispiel, sollte also keine Entscheidung oder sogar eine Zurückweisung erfolgen.

Grundproblem bei der Bewertung von Sprachmustern

Im Gegensatz zu statischen Mustern wie Bildern, Unterschriften, etc. handelt es sich bei Sprache um einen zeitlich veränderlichen Prozess. Das bedeutet, dass zusätzlich zur Variabilität *innerhalb* der Merkmalsvektoren (intrinsic variability) noch die *zeitliche* Variabilität ihrer Abfolge in der Zeit hinzukommt. Anschaulich heißt das nichts anderes, als dass jede gesprochene Variante eines gleichen Wortes verschiedene Längen haben kann, und zusätzlich noch unterschiedliche Dehnungen und Kompressionen innerhalb der Musterfolge aufweist. Aus diesem Grund kann man die Sprachmustererkennung als zweifachen Prozess sehen: einerseits sucht man ähnliche Muster (Merkmalsvektoren), andererseits sucht man auch noch die ähnlichste Abfolge dieser Muster in der Zeit.

Skizze

```
Mehrere Musterwörter im Vergleich
```

Wichtigste Arten der Sprach-Mustererkennung in ihrer historischen Reihenfolge:

1. Einfacher Mustervergleich (linearer Klassifikator)
   Die direkte Methode: für jedes Wort, das im Erkennerwortschatz vorkommt, werden ein oder mehrere Muster aus der Trainingsstichprobe extrahiert und die zugehörigen Merkmalsvektorfolgen abgespeichert. Bei der Erkennung wird das unbekannte Wort ebenfalls in eine Merkmalsvektorfolge verarbeitet und dann mit jedem Muster verglichen. Unterschiedliche Längen werden durch geeignete, ziemlich brutale Methoden auf eine Normlänge gebracht (in der Regel die des Musters). Das ähnlichste Muster repräsentiert das erkannte Wort (bei angenommener Max-Entscheidungsfunktion). Die Normierung auf gleiche Länge erfolgt entweder durch Abschneiden oder Ergänzen von Nullvektoren oder durch lineare Interpolation des Pfades im Vektorraum, den die Vektoren aufspannen.

Skizze

`Lineare Interpolation`

Der eigentliche Vergleich eines einzelnen Merkmalsvektors $\vec{v}$ mit dem korrespondierenden Mustervektor $\vec{m}$ erfolgt über ein geeignetes Abstandsmaß im Vektorraum. Meistens ist dies der *Euklidische Abstand* $d(\vec{v}, \vec{m})$ der beiden Vektoren $\vec{v}$ und $\vec{m}$ im Raum, also die geometrische Entfernung ihrer Endpunkte:

$d(\vec{v}, \vec{m}) = \sqrt{\sum_{n=1}^{N}(v_n - m_n)^2}$

Das schaut komplizierter aus als es ist. Für den zweidimensionalen Fall $N = 2$ wird die Formel zur bekannten 'Dreiecksformel':

$d(\vec{v}, \vec{m}) = \sqrt{(v_1 - m_1)^2 + (v_2 - m_2)^2}$

Skizze

`Rechtwinkliges Dreieck, Euklid`

Der Abstand D zweier *Vektorfolgen* A und B der fixen Länge K (also K Vektoren) ist dann einfach die Summe über alle einzelnen Abstände:

$D(A, B) = \sum_{k=1}^{K} d(\vec{v_k}, \vec{m_k})$

2. Dynamic Time Warping, DTW (Dynamische Programmierung, DP)
   Methode zwei Signale durch nicht-lineare Zeitverzerrung so aufeinander abzubilden, dass der 'Abstand' der beiden Signale minimal wird.

Skizze

DTW

Verwendung der DTW als Mustererkennung: hintereinander Abbildung auf alle gespeicherten Muster; Vergleich der Summe aller lokalen Abstände korrespondierender Merkmalsvektoren entlang des optimalen Abbildungspfades.
Diese Methode ist immer noch weit verbreitet, da sie einfach zu implementieren und wenig rechenaufwändig ist. Sie wird im Teil 3 dieser Veranstaltung ausführlich besprochen.

3. Hidden Markov Modelle (HMM)
   Statistische Modelle, die für eine zeitliche Abfolge von Merkmalsvektoren die Wahrscheinlichkeit berechnen, dass diese Folge von einer bestimmten Kategorie (auf die das Modell trainiert wurde) erzeugt wurde (Produktionsmodell).
   Die Kategorie (das Sprachsegment) wird dabei durch eine Markov-Kette von Zuständen modelliert, die jeweils eine statistische Verteilungsfunktion für die Produktion einer Art von Merkmalsvektoren enthalten, die für diese Kategorie charakteristisch sind. Zwischen den Zuständen des HMM existieren statistisch gewichtete Übergänge, welche die zeitliche Abfolge der Merkmalsvektoren modellieren sollen.
   Der Mustervergleich erfolgt kurz gesagt folgendermaßen: Eine Merkmalsvektorfolge eines unbekannten Musters wird so auf den Markov-Prozess verteilt, dass das insgesamte Produkt aus Übergangswahrscheinlichkeiten längs des Pfades durch die Zustände sowie der sog. Emmissionswahrscheinlichkeiten für jeden Merkmalsvektor in seinem zugeordneten Zustand ein Maximum ergeben. Die Merkmalsvektorfolge wird mit jedem HMM, also mit jeder Kategorie, verglichen und diejenige mit der maximalen Gesamtwahrscheinlichkeit ausgewählt (bei einer Max-Entscheidung!).

   Skizze

HMM mit drei Zuständen + Merkmalsvektorfolge

HMM sind derzeit das erfolgreichste Modell der ASR. Wird in Teil 3 detailiert besprochen.

4. Künstliches Neuronales Netz (Artificial Neural Network ANN)
   Ein ANN ist ein *Funktionen-Approximator*, der beliebige, auch nicht-lineare funktionale Zusammenhänge modellieren kann.
   Ein ANN besteht aus sehr vielen, sehr primitiven mathematischen Operatoren und soll die neuronale Verarbeitung in biologischen Systemen simulieren. Allerdings bestehen erhebliche Unterschiede der in der Technik gängigen Modelle gegenüber dem Nervensystem in einem lebenden Organismus. Daher die korrekte Nomenklatur als 'Künstliches Neuronales Netz'.
   Die zentralen Rechenelemente in einem ANN sind Multiplikation ('Gewichtung') in den Verknüpfungen der Neurone und Addition in den Neuronen selber. Hinzu kommt noch eine nicht-lineare Begrenzung der Ausgänge aller Neuronen, um ein Aufschaukeln ('Epilepsie') des Netzes zu vermeiden.

   Skizze

`Einzelnes künstliches Neuron, Einfaches Netzwerk`

ANN werden (u.a.) nach ihrer Topologie in verschiedene Klassen eingestuft. Die wichtigsten Merkmale sind:

- Anzahl der 'hidden layer', d.h. der verborgenen Schichten
  Je höher die Anzahl der 'hidden layer' (und deren Neuronenzahl) in einem ANN ist, desto höher ist seine Mächtigkeit, nichtlineare Zusammenhänge zu modellieren (sog. 'Deep Neuronal Network, DNN). In der Praxis ist dies allerdings durch die Rechenungenauigkeit und die damit verbundene Fehlerfortpflanzung mit Problemen verbunden, die durch gezielte Manipulation der 'hidden layer' vermieden werden können. Die Entwicklung von erfolgreichen DNN haben in der ASR in den Jahren 2005-2013 einen großen Performanzgewinn gebracht; alle modernen ASR-Systeme arbeiten heute mit DNN. Theoretisch ist ein einzelner sehr großer hidden layer genauso mächtig; es ist nicht ganz klar, warum DNN erfolgreicher sind.
- 'feed-forward' versus 'recursive'
  Feed-Forward ANN haben keine 'Rückleitung' von den Ausgängen eines Layers zu den Eingängen eines tieferen Layers. Rekursive ANN können solche Rückleitungen haben

und damit alle Probleme, die bei potentiell selbstverstärkenden Systemen auftreten können (Instabilität). In der Literatur ist man sich uneinig, ob rekursive ANN einen Vorteil gegenüber einfachen Feed-Forward Topologien bringen. Auf jeden Fall sind sie mathematisch schwieriger zu interpretieren.

- Zeitliche Dynamik
  Je nachdem ob zeitliche Vorgänge innerhalb des Netzes eine Rolle spielen oder nicht, unterscheidet man zwischen einfachen Perceptrons (keine Dynamik) und z.B. 'time delay neuronal network' oder 'pulse coded neuronal network'.

Das ANN kann als Mustererkennung genutzt werden, wenn als Ausgangsfunktion eine Entscheidungsfunktion gefordert wird (Ja/Nein). Auch bei diesem Ansatz ist allerdings der Ausgang des Netzwerks eine kontinuierliche Bewertungsfunktion, deren Output von einer nachgeschalteten Entscheidungsfunktion verarbeitet werden muss.
Es lässt sich zeigen, dass ANN unter bestimmten Voraussetzungen am Ausgang Schätzwerte für sog. *a-posteriori Wahrscheinlichkeiten* darstellen; das ist die Wahrscheinlichkeit, dass eine Klasse m für einen gegebenene Merkmalsvektor am Eingang zuständig ist (s. Bourlard & Morgan, 1994).
ANNs werden in der ASR zumeist nur in sog. hybriden Verfahren verwendet. Das bedeutet, dass man das ANN zur Bewertung einzelner Merkmalsvektoren einsetzt, aber die zeitliche Modellierung, also den Ablauf der Merkmalsvektoren einem traditionellen HMM überlässt. Auf die Grundfunktionen von ANN (Training, Test) werden wir in Abschnitt C kurz eingehen.