ALBERT-LUDWIGS-UNIVERSITÄT
FREIBURG
INSTITUT FÜR INFORMATIK

Lehrstuhl für Mustererkennung und Bildverarbeitung
Prof. Dr. Hans Burkhardt

Bestimmung und Untersuchung von Signifikanzgewichtungen für die Erkennung von handgeschriebenen Buchstaben

Diplomarbeit

Dirk Bockhorn

September 1999 – Februar 2000

Erklärung

Hiermit erkläre ich, daß die vorliegende Arbeit von mir selbständig und nur unter Verwendung der aufgeführten Hilfsmittel erstellt wurde.

Freiburg, den 25.2.2000

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die automatische Erkennung von handgeschriebenen Zeichen bzw. Zeichenketten ist ein klassisches Problem in der Mustererkennung, welches bis heute nicht vollständig gelöst wurde. Diese Arbeit beschäftigt sich mit der *On-Line Handschriftenerkennung*, insbesondere mit der Erkennung von handgeschriebenen Buchstaben und Ziffern. Hierbei werden die Buchstaben mit einem speziellen Stift auf ein Digitalisiertablett geschrieben. Das Tablett registriert die Bewegung der Stiftspitze und gibt den abgetasteten Schriftzug an das Erkennungssystem weiter. Dieses weist dann das Muster einer Zeichenklasse zu.

Systeme zur On-Line Handschriftenerkennung werden zur Zeit in einigen *Personal Digital Assistants* (PDAs) erfolgreich eingesetzt. Stiftbasierte PDAs sind kleine, handliche Computer, die ohne eine Tastatur auskommen und vollständig mittels eines speziellen Stiftes bedient werden. Die verbreitetesten PDAs beschränken sich auf Großbuchstaben und benötigen einen speziellen Zeichensatz, anstelle der gewöhnlichen Blockbuchstaben.

Das Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, das eine schreiberunabhängige Erkennung von Groß- und Kleinbuchstaben sowie Ziffern ermöglichen soll. Das Verfahren vergleicht das unbekannte Testmuster mit einer Menge von Referenzbuchstaben. Das Zeichen, welches den ähnlichsten Referenzbuchstaben repräsentiert, wird dem Testmuster zugewiesen. Der Vergleich wird durch eine zeitliche Angleichung der beiden Muster realisiert. Diese Angleichung wird mit Hilfe des *Dynamic Time Warping*-Algorithmus (DTW) gelöst.

In dieser Arbeit wird eine Methode beschrieben, die den DTW-Algorithmus so erweitert, daß signifikante Regionen in einem Buchstaben durch Einführung von Modellgewichten stärker berücksichtigt werden als weniger bedeutende Regionen. Zum einen wird eine Formulierung des DTW-Algorithmus entwickelt, die es erlaubt, verschiedene Zeitpunkte in einem Muster unterschiedlich zu gewichten. Zum anderen wird ein Verfahren zur Bestimmung der Modellgewichte vorgestellt.

Die meisten Probleme in der Mustererkennung lassen sich durch ein einheitliches Schema beschreiben (siehe Abbildung 1.1). In einem ersten Schritt werden die Objekte, die erkannt werden sollen, digitalisiert. In dem vorliegen Fall sind das die geschriebenen Buchstaben. Dann findet eine Vorverarbeitung der aufgenommen Daten statt. Anschließend werden die für das Problem charakteristischen Merkmale extrahiert. Im letzten Schritt, der eigentlichen Klassifikation, wird das aufgenommene Muster einer eindeutigen Bedeu-

tungsklasse zugeordnet.

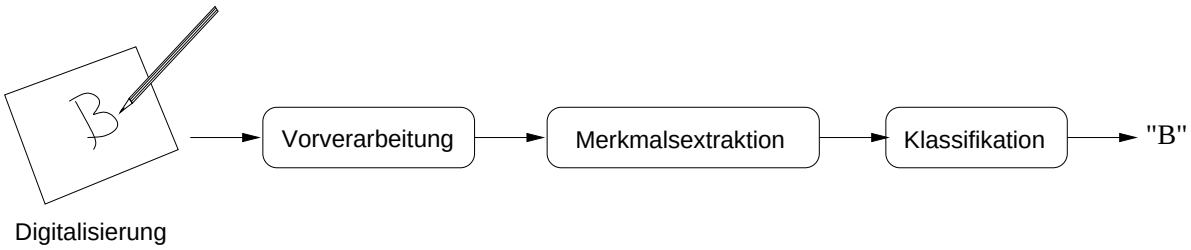


Abbildung 1.1: Allgemeine Schema zur Mustererkennung

Diesem Schema folgt auch der Aufbau dieser Arbeit. Im folgenden Kapitel werden einige Begrifflichkeiten erläutert und das zu lösende Problem von anderen Problemstellungen in der Handschriftenerkennung abgegrenzt. Das dritte Kapitel beschäftigt sich mit der Extraktion von geeigneten Merkmalen. In Kapitel 4 werden der zur Klassifikation herangezogene DTW Algorithmus und die entwickelten Erweiterungen beschrieben. In Kapitel 5 werden die Methoden zur Berechnung der Referenzmodelle vorgestellt. Eine Untersuchung der statistischen Verteilung der Merkmale und die Präsentation der Klassifikationsergebnisse schließt sich in Kapitel 6 an. Abschließend wird in Kapitel 7 die Arbeit zusammengefaßt und ein Ausblick auf mögliche Verbesserungen gegeben. Im Anhang werden die Klassifikationsergebnisse detailliert präsentiert.

Kapitel 2

Grundlagen

Dieses Kapitel führt grundlegende Begrifflichkeiten und Problemstellungen in der Handschriftenerkennung ein. Im ersten Teil werden einige Besonderheiten von Handschrift aufgezeigt. Darauf folgt eine Charakterisierung verschiedener Probleme der Handschriftenerkennung. Eine kurze Beschreibung existierender Verfahren schließt das Kapitel ab. Einen guten Überblick über den Stand der Technik in der On-Line Handschriftenerkennung gibt der Übersichtsartikel von C. Tappert *et al* [19].

2.1 Handschrift

Geschriebene Sprache besteht aus einem Alphabet von Zeichen oder Buchstaben sowie Satzzeichen. Die Unterschiede zwischen verschiedenen Ausprägungen desselben Zeichens sind nicht so stark, wie die Unterschiede zwischen verschiedenen Zeichen. Ohne diese Eigenschaft wäre keine Erkennung möglich. Das gilt sowohl für den Menschen als auch für den Computer. Ausnahme zu dieser Regel sind zum Beispiel die *0* und das *O*, die beide gleich oder zumindest sehr ähnlich geschrieben werden. In diesem Fall läßt sich aber meist durch den Kontext entscheiden, ob es sich um eine Ziffer oder um einen Buchstaben handelt [19].

Handschrift besteht aus einer zeitlichen Sequenz einzelner Striche, wobei ein Strich (engl.: stroke) einen zusammenhängenden Teil eines Schriftzuges vom Aufsetzen (pen down) des Stiftes bis zum Absetzen (pen up) bezeichnet. Zeichen werden üblicherweise vollständig geschrieben, bevor das nächste Zeichen geschrieben wird. Doch manchmal kommt es zu *verzögerten Strichen* (eng.: delayed strokes), wenn zum Beispiel *t*-Striche oder *i*-Punkte nachträglich gesetzt werden. Zeichen werden in einer bestimmten Reihenfolge räumlich angeordnet (z.B. von links nach rechts).

Die relative Größe und Position der Buchstaben ist ein wichtiges Merkmal. Ein Schreiber orientiert sich dabei im wesentlichen an vier Referenzlinien. Großbuchstaben werden auf der Grundlinien geschrieben und reichen bis zur Oberlängenlinie. Die meisten Kleinbuchstaben sind nur halb so hoch wie ein Großbuchstabe. Einige haben jedoch Oberlängen und reichen bis zur Oberlängenlinie, und/oder besitzen Unterlängen, die unterhalb der Grundlinie liegen.

2.2 Charakterisierung von Handschriftenerkennungsproblemen

2.2.1 Schwierigkeitsstufen

Die Problemarten der Handschriftenerkennung lassen sich hauptsächlich durch das verwendete Alphabet und den Schreibstil charakterisieren. Die Größe des Alphabets kann sehr stark variieren. Das chinesische Grundalphabet, zum Beispiel, besteht aus 3000–5000 Symbolen, während das lateinische Alphabet 26 Buchstaben (52 mit Kleinbuchstaben und ohne Umlaute) umfaßt. Sollen nur Zahlen erkannt werden, so besteht das Alphabet nur aus den zehn Ziffern.

Diese Arbeit beschäftigt sich mit der Erkennung von Buchstaben. Man unterscheidet zwischen Buchstaben- und Worterkennung. Bei der Worterkennung muß das unbekannte Wort in einzelne potentielle Buchstaben segmentiert werden, die anschließend von einem Buchstabenerkenner bewertet werden. Diese strikte Trennung von Buchstabensegmentierung und -erkennung ist allerdings nicht bei allen Schreibstilen möglich.

In Abb. 2.1 sind die verschiedenen Schreibstile, sortiert nach Schwierigkeitsgrad, dargestellt. Die Schreibstile sind im einzelnen:

- gerahmte einzelne Zeichen (engl.: boxed discrete characters): Die Buchstaben müssen einzeln in die dafür vorgesehen Kästchen geschrieben werden. Eine Segmentierung ist hier nicht notwendig, da dies bereits vom Schreiber erledigt wird.

- einzelne Zeichen mit Abstand (engl.: spaced discrete characters): Hier muß eine Segmentierung stattfinden. Diese ist meist einfach, da ein Leerraum zwischen den Zeichen existiert.

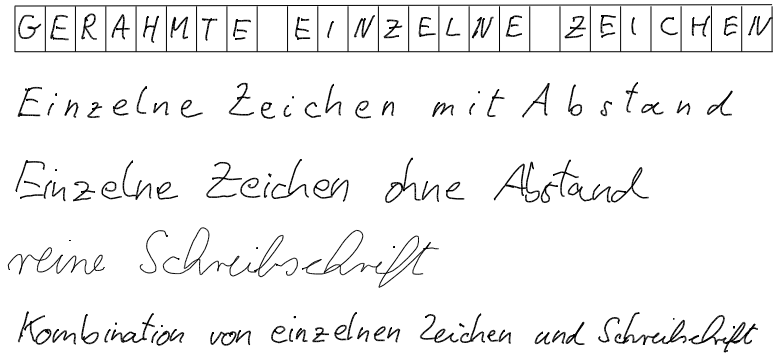


Abbildung 2.1: Verschiedene Typen von Handschriften (erhöhter Schwierigkeitsgrad von oben nach unten)

- einzelne Zeichen ohne Abstand (engl.: run-on discretely written characters): Die letzten drei Probleme machen eine Segmentierung im Zusammenspiel mit dem Erkenner notwendig. Dabei ist dieser Fall noch am einfachsten zu handhaben, da das Ende eines Zeichens auch gleichzeitig das Ende eines Strichs sein muß.

- reine Schreibschrift (engl.: pure cursive script writing): Da mehrere Zeichen mit einem Strich geschrieben werden können, ist die Segmentierung von Schreibschrift ein schwieriges Problem.

- Kombination von einzelnen Zeichen und Schreibschrift (engl.: mixed cursive, discrete and run-on discrete): Für die Segmentierung gilt dasselbe, wie für das vorherige Problem. Hinzu kommt die hohe Variabilität der einzelnen Buchstaben.

Die Unterscheidung ähnlicher Formen verschiedener Buchstaben ist ein schwieriges Problem für die automatische Erkennung. Ähnliche Buchstaben sind z.B. *U-V, C-L, C-c, I-1, 2-Z, a-d* und *n-h*. Die Buchstabenpaare *P-p* und *Y-y* lassen sich meist anhand der relativen Position zur Grundlinie unterscheiden [19].

2.2.2 Off-Line vs On-Line

In der Handschriftenerkennung wird grundsätzlich zwischen zwei verschiedenen Arten von Daten unterschieden: *Off-Line* und *On-Line* Daten.

(a) On-Line (b) Off-Line

Abbildung 2.2: Die Buchstaben wurden (a) auf ein Blatt, das auf ein Digitalisiertablett angebracht wurde, geschrieben und (b) mittels eines Scanners aufgenommen und binärisiert.

Off-Line Daten

Die Off-Line Daten werden typischerweise nach dem Schreiben aufgenommen. Die Buchstaben und/oder Ziffern werden auf Papier geschrieben und anschließend, etwa mit einem Scanner, digitalisiert. Diese Methode der Erfassung von Handschriften findet man in der Praxis z.B. bei automatischen Schecklesegeräten, wo handbeschriebene Überweisungsformulare ausgewertet werden oder bei Briefsortieranlagen.

Die so aufgenommen Daten werden üblicherweise in einer binären oder grauwertigen Bildmatrix gespeichert.

On-Line Daten

On-Line Daten werden während des Schreibens aufgenommen. Dabei werden die (x, y)-Koordinaten einer speziellen Stiftspitze mit einer gegebenen Abtastrate aufgenommen. Moderne Digitalisiertabletts nehmen nicht nur die Koordinaten auf, wenn sich die Stiftspitze auf der Schreibunterlage befindet (*pen down*-Bewegung), sondern auch wenn sie nur knapp darüber „schwebt"(*pen up*-Bewegung). Neben den x und y Koordinaten stehen je nach Tablett noch weitere Werte zur Verfügung: der Druck, mit dem auf die Schreibunterlagen gedrückt wird, der Winkel des Schreibgerätes zur Schreibebene und die Drehung um die Achse des Stiftes.

Die On-Line aufgenommenen Schriftzüge werden als Funktion der Zeit $\mathbf{x}(t)$ im Kontinuierlichen, bzw. im diskreten Fall als Vektor $\mathbf{X} = (X_1, X_2, \ldots, X_N)$ beschrieben. Im folgenden werden nur die absoluten x- und y-Koordinaten verwendet: $\mathbf{x}(t) = (x(t), y(t))$, bzw. $\mathbf{X} = ((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N))$.

Off-Line/On-Line - Was ist besser ?

Im allgemeinen sind On-Line Daten für die automatische Erkennung besser geeignet. So sind in Off-Line Daten keine temporalen oder dynamischen Informationen enthalten, wie die Anzahl der Striche, die Reihenfolge der Striche, die Schreibrichtung der einzelnen Striche oder die Geschwindigkeit, mit der ein Strich geschrieben wurde. Zudem kann häufig auf zusätzliche Information wie dem gemessenen Druck auf die Schreibunterlage oder dem Winkel zwischen dem Schreibutensil und der Schreibebene zugegriffen werden. Die zeitlichen Informationen können aber auch zu Inkonsistenzen führen. Verzögerte t-Striche oder i-Punkte müssen identifiziert und den entsprechenden Segmenten zugeordnet werden.

Für die Verwendung von Off-Line Daten spricht, daß sie in dem Sinne allgemeiner sind, daß sich On-Line Daten einfach in die Off-Line Darstellung konvertieren lassen. Außerdem wird kein spezielles Schreibgerät benötigt. Einige Anwendungen erlauben nur die Verwendung von Off-Line Daten, z.B. in Briefsortieranlagen. Mit Hilfe von Konturverfolgungsalgorithmen kann die dynamische Information zurückgewonnen werden und somit wären die Methoden der On-Line Erkennung auf Off-Line Daten anwendbar [12].

2.3 Das Unipen-Datenformat

Das Ziel des UNIPEN Projektes [11] ist es, eine große Datenbank für On-Line Handschriftenerkennung aufzubauen. Für die On-Line Handschriftenerkennung existiert bisher kein offizieller Benchmark, so daß die Leistungen der einzelnen Systeme nur sehr schwer zu vergleichen sind. Dieses Problem wird mit dem noch zu veröffentlichenden Datensatz hoffentlich gelöst. Eine Entscheidung diesbezüglich wurde für Ende 1999 angekündigt.

Der hierarchische Aufbau des UNIPEN Formats ist flexibel gehalten. Prinzipiell läßt sich jede beliebige Hierarchie realisieren. Üblich ist eine Aufteilung in Textzeilen, Wörter und Buchstaben. Ein Element aus einer Hierarchieebene wird als Segment bezeichnet und besteht aus mehreren Komponenten oder Teilen von Komponenten. Eine Komponente ist ein vollständiger Strich (pen down-Bewegung), bzw. eine vollständige pen up-Bewegung.

```
.VERSION 1.0
.COORD X Y T P BUTTON THETA PHI
.HIERARCHY TEXT LINE WORD CHARACTER

.SEGMENT TEXT 0-1159 ? ""
.SEGMENT LINE 0-73 ? ""
.SEGMENT WORD 0-5:3 OK "The"
.SEGMENT CHARACTER 0-2 OK "T"
.PEN_DOWN
2076    12996   21099454        10896   1       337     52
[...]
2126    12724   21099493        145734  1       339     53
2126    12733   21099497        58566   1       339     53
2126    12758   21099502        7491    1       339     53
.PEN_UP
2126    12758   21099507        0       0       339     53
2112    12819   21099512        0       0       339     53
2080    12871   21099517        0       0       339     53

1885    13030   21099546        0       0       336     51
1873    13030   21099551        0       0       336     51
1862    13027   21099556        0       0       336     51
.PEN_DOWN
1862    13027   21099561        4086    1       336     51
```

Abbildung 2.3: Auszug aus einer UNIPEN Datei.

Eine ausführliche Beschreibung ist in [10] zu finden. Abb. 2.3 zeigt einen Auszug aus einer UNIPEN-Datei.

Die Speicherung der Daten im ASCII-Format erlaubt eine problemlose Portierung auf verschiedene Plattformen. Es existieren Programme zum Bearbeiten und Anzeigen der Handschriften (Uptools von G. Abbink, L. Schomaker und L. Vuurpijl von dem Nijmegen Institute for Cognition and Information, Nijmegen University).

2.4 Datenakquisition

Für die nachfolgend beschriebenen Untersuchungen sind On-Line-Handschriftproben gesammelt worden. Die ca. ein Dutzend Schreiber setzen sich aus Studenten und Mitarbeitern/-innen des Instituts zusammen. Es wurden zwei Datensätze erzeugt:

- *Einzelne Buchstaben und Ziffern*: Es sollte jeweils eine Zeile mit demselben Buchstaben beschrieben werden. Verlangt waren alle Buchstaben des Alphabets (ausgenommen der Umlaute), sowohl als Groß-, als auch Kleinbuchstaben und die Ziffern 0-9. Eine Zeile hatte je nach Schreiber und Zeichen Platz für 10-15 Zeichen. Da dieselben Buchstaben hintereinander geschrieben wurden, wurde meist die Art einen Buchstaben zu schreiben beibehalten. Dennoch weisen die Daten eine erkennbare Variabilität

auf. Dieser Datensatz ließ sich relativ einfach automatisch segmentieren, da zwischen den Zeichen etwas Platz gelassen wurde.

- *Englischsprachiger Text*: Die Probanden wurden gebeten einen englischen Text bestehend aus 197 Wörtern (1097 Zeichen) abzuschreiben[1]. Dieser Text mußte dann manuell erst in einzelne Wörter und dann in Buchstaben und Satzzeichen segmentiert werden. Dazu wurde das Programm upWorks aus der upTools Sammlung verwendet [11]. Während im ersten Datensatz alle Zeichen in annähernd gleicher Häufigkeit vorkommen, so sind hier einige Buchstaben gar nicht enthalten.

2.4.1 Technische Daten

Die Daten wurden mit einem WACOM UltraPad A3 Digitalisiertablett akquiriert. Die wichtigsten technischen Daten sind in der Tabelle 2.1 zusammengefaßt.

Auflösung	2540 lpi/100 lpmm[2]
Genauigkeit	$\pm 0,25$ mm
Druckstufen	256 Stufen
Maximale Lesehöhe	5 mm
Maximale Übertragungsrate	205 Punkte pro Sekunde
Aktive Fläche	457,2 x 304,8 mm

Tabelle 2.1: Technische Daten für das WACOM UltraPad A3

2.5 Methoden zur Handschriftenerkennung

Die Methoden zur On-Line Handschriftenerkennung sind vielfältig. Einige Methoden sind allgemeine Verfahren aus der Mustererkennung und sind daher problemunabhängig. Andere nutzen spezielle Eigenschaften des verwendeten Alphabets aus. Im folgenden werden nun einige Verfahren kurz erläutert.

Merkmalanalyse: Ein Buchstabe kann eindeutig durch eine Menge von Merkmalen beschrieben werden. Diese Merkmale können binär sein, zum Beispiel, ob der Buchstaben Oberlängen besitzt oder nicht, ob er einen Punkt hat oder nicht. Bei binären Merkmalen wird meistens mit Hilfe eines Entscheidungsbaumes die Klassenzugehörigkeit eines Testmusters bestimmt. Enthält ein Buchstabe Unterlängen, dann kann es sich dabei nur um ein *f, g, j, p, q* oder *y* handeln. Besitzt dieser Buchstabe ferner einen Punkt, so muß das Testmuster ein *j* sein.

Ein Nachteil von binären Entscheidungsbäumen ist, daß sie keine Alternative erzeugen. Das kann in einem Nachverarbeitungsschritt zum Beispiel dann von Nutzen sein, wenn das Programm an dieser Stelle unsicher ist, ob es sich um ein *e* oder ein *c* handelt. Wenn dann aber aus dem Kontext ersichtlich ist, daß das *e* wahrscheinlicher ist, dann könnte dies

[1] Der Text wurde entnommen aus: Life, the Universe and Everything von Douglas Adams, Kapitel 11
[2] lpi: lines per inch, lpmm: lines per mm

berücksichtigt werden.

Zeitliche Sequenzen von Zonen, Richtungen oder Extrema: Diese Methoden basieren hauptsächlich auf dynamischer Information. Ein das Zeichen umschließendes Rechteck wird in Zonen aufgeteilt. Die Sequenz von Zonen, die den Schriftzug enthalten, wird in der Reihenfolge des Durchkreuzens berechnet. Über diese Sequenz erhält man, etwa über ein Lexikon, die Klassenzugehörigkeit des Testmusters. Eine andere Methode beschreibt einen Buchstaben als eine Sequenz von lokalen Extrema (hoch, runter, links, rechts). Diese Sequenz wird *Kettenkodierung* (chain code) genannt.

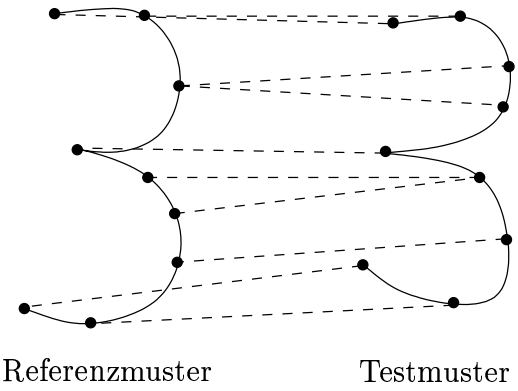


Abbildung 2.4: Beispiel für Dynamic Time Warping (DTW) [20]

Curve Matching: Die Kurve des Testmusters und die Kurve sämtlicher Referenzmuster werden angeglichen. Dasjenige Referenzmuster, dessen Kurve sich am besten anpassen läßt bestimmt dann die Klassenzugehörigkeit des Testmusters. Bei den betrachteten Kurven handelt es sich meist um Funktionen bzgl. der Zeit. Zu dieser Klasse gehört die in dieser Arbeit verwendete Methode des Dynamic Time Warping. C. Tappert hat bereits früh diese Methode auf die Handschriftenerkennung übertragen. Er bezeichnet dieses Verfahren als Elastic Matching [20].

Viele existierende Verfahren sind Kombinationen aus den genannten Methoden. Diesbezüglich sind weitere Methoden zu nennen, wie *Analyse durch Synthese, Kodierung der Striche (stroke codes)* und *Markov-Modelle.*

Kapitel 3

Vorverarbeitung und Merkmalsextraktion

3.1 Vorverarbeitung

Die Vorverarbeitung dient im allgemeinen dazu, unerwünschte Störungen, wie etwa ein durch das Aufnahmesystem induziertes Rauschen, zu entfernen. Da die dieser Arbeit zugrundeliegende Datenbasis von einem hochauflösenden Digitalisiertablett stammt, war in dieser Hinsicht keine besondere Behandlung notwendig.

3.1.1 Entfernen der Duplikate

Bei der Datenakquisition ist es häufig aufgetreten, daß am Anfang eines Schriftzuges die Stiftspitze kurze Zeit an einer Stelle auf dem Tablett ruht. Da die Position in zeitlich äquidistanten Abständen abgetastet wird, kann es dazu kommen, daß zeitlich benachbarte Punkte dieselben Werte haben. Die Wiederholung derselben Punkte erschwert die Berechnung einiger Merkmale, wie zum Beispiel der Richtung der ersten Ableitung (siehe 3.2.2). Dieses Verharren der Stiftspitze enthält keine wertvolle Information und wird daher entfernt.

3.1.2 Größennormierung

Diese Arbeit beschäftigt sich mit der Erkennung von Buchstaben. Die hier gewonnen Erkenntnisse zur Klassifikation von Buchstaben, sollen allerdings in späteren Arbeiten auch in die Worterkennung einfließen. Daher liegen einige Daten in Form von manuell in Buchstaben segmentierten Wörtern vor. Nicht alle Merkmale, die sich zur Klassifikation von Buchstaben eignen, lassen sich in der Worterkennung anwenden. So ist auch eine spezielle Größennormierung notwendig. Diese wird in diesem Abschnitt vorgestellt.

Ein für den Menschen entscheidendes Kriterium zur Unterscheidung verschiedener Buchstaben sind die *Oberlängen* (engl.: ascenders), bzw. *Unterlängen* (engl.: descenders) der Buchstaben. Unter Oberlängen versteht man die Teile eines Buchstaben, die sich oberhalb des Rumpfes eines Wortes befinden. Entsprechend bezeichnen Unterlängen diejenigen Teile, die sich unterhalb der Grundlinie befinden (siehe Abb. 3.1). Die *Kernlinie* ist parallel

zur Grundlinie und trennt die Oberlängen vom Rumpf des Wortes. Der vertikale Abstand zwischen Kern- und Grundlinie wird als *Kernhöhe* bezeichnet.

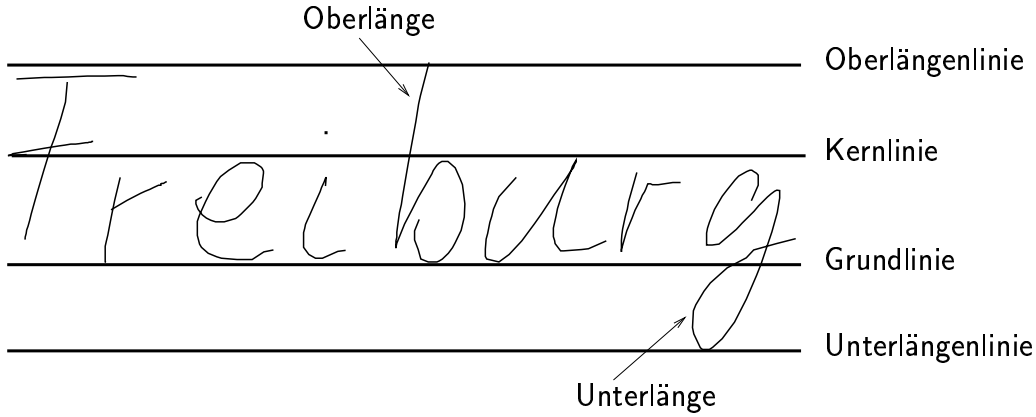


Abbildung 3.1: Die vier Referenzlinien am Beispiel des Wortes „Freiburg“.

Das Ziel einer Normierung sollte es sein, daß ein Buchstabe auf dieselbe Größe normiert wird, unabhängig davon, in welchem Wort er sich befindet. Folgendes Beispiel soll diese Problemstellung verdeutlichen. Würde ein Wort nur durch seine Ausdehnung in y-Richtung normiert werden, so würde das *a* aus *Haus* nach der Normierung nur halb so groß sein, wie das *a* aus *arm*.

Die gewünschte Eigenschaft kann dadurch erreicht werden, daß die Wörter nicht bezüglich der vertikalen Gesamtausdehnung, sondern auf eine bestimmte Kernhöhe h_{norm} normiert werden. Diese Methode hat den Vorteil, daß sie unabhängig von der Existenz von Unter- und Oberlängen ist.

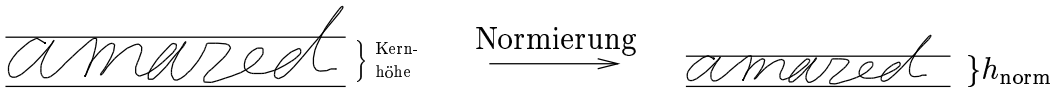


Abbildung 3.2: Größennormierung nach der Kernhöhe.

Angenommen, die Positionen der Referenzlinien seien bekannt und die Schreibrichtung ist parallel zur x-Achse, d.h. die Referenzlinien sind ebenfalls parallel zur x-Achse. Dann läßt sich die gewünschte Normierung wie folgt berechnen:

$$\begin{aligned} \hat{x}_n &= h \cdot x_n \\ \hat{y}_n &= h \cdot y_n \end{aligned} \tag{3.1}$$

wobei h der Normierungsfaktor ist

$$h = \frac{h_{\text{norm}}}{y_{\text{Grundlinie}} - y_{\text{Kernlinie}}} \tag{3.2}$$

Im nächsten Abschnitt wird ein Verfahren angegeben, das aus On-Line gewonnen Wörtern die vier Referenzlinien extrahiert.

3.1.3 Verfahren zur Bestimmung der Referenzlinien

Das hier beschriebene Verfahren zur Bestimmung der Referenzlinien orientiert sich an den Arbeiten von Bozinovic [1] und Bunke [2]. Das dort angegebene Verfahren setzt allerdings Off-Line Daten voraus. Im folgenden wird dieses Verfahren erläutert und an den entsprechenden Stellen für On-Line Daten modifiziert.

Es soll die Annahme gelten, daß die Referenzlinien parallel zur horizontalen x-Achse liegen. Dies ist eine realistische Einschränkung, denn es ist vorstellbar, daß durch die Vorgabe der Grundlinie der Benutzer gezwungen wird, die Zeichen in einer festgelegten Orientierung zu schreiben. Ist die Schreibrichtung unbekannt, muß vorher die Orientierung der Referenzlinien bestimmt werden [3].

Die Methode zur Bestimmung der Referenzlinien basiert auf der Analyse des *horizontalen Verteilungshistogramms* $H(y)$, auch *horizontales Projektionsprofil* (horizontal projection profile) genannt. Die erste Ableitung $\Delta H(y)$ von $H(y)$ wird berechnet und nach dem positiven und negativen Extremum durchsucht. Die y-Koordinate der maximalen Ableitung entspricht dann der Position der Grundlinie $y_{\text{Grundlinie}} = \arg\max \Delta H(y)$. Und entsprechend gilt für die Kernlinie: $y_{\text{Kernlinie}} = \arg\min \Delta H(y)$ (siehe Abb. 3.3).

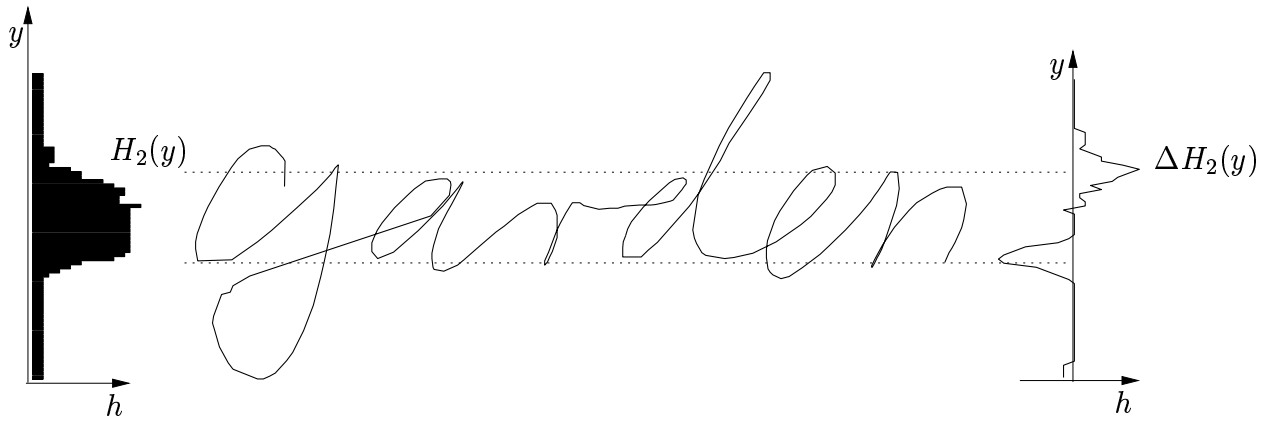


Abbildung 3.3: Links das horizontale Projektionsprofil $H(y)$ von „garden“ und rechts $\Delta H(y)$ (mit vorangegangener Mittelwertfilterung).

Im Off-Line Fall berechnet sich das horizontale Projektionsprofil aus der Anzahl der 'schwarzen' Werte pro Bildzeile. Leider läßt sich diese Methode in dieser Form nicht auf On-Line Daten anwenden. Der folgende Abschnitt behandelt zwei Verfahren zur Bestimmung des Verteilungshistogramms für On-Line Daten.

Berechnung des horizontalen Projektionsprofils für On-Line Daten

Beide Methoden unterteilen zunächst die Ebene zwischen dem maximalen und dem minimalen y-Wert in eine feste Anzahl horizontaler Streifen (Bins) gleicher Höhe. 60–80 Bins hat sich als ein guter Wert erwiesen.

Methode I. Die erste Methode zählt die Abtastpunkte, die in einem Bin enthalten sind. Für ein gegebenes Handschriftsegment $\mathbf{X} = ((x_1, y_1), \ldots, (x_N, y_N))$ ergibt sich dann:

$$H_1^X(n) = |\{i \in \{1, \ldots, N\} \mid (n-1) \cdot W + y_{min} \leq y_i < n \cdot W + y_{min}\}| \tag{3.3}$$

für $n \in \{1, \ldots, B\}$ und wobei

$$\begin{aligned} y_{min} &= \min_i y_i \\ y_{max} &= \max_i y_i \\ B &= \text{Anzahl der Histogrammbins} \\ W &= \tfrac{y_{max}-y_{min}}{B} \quad \text{Höhe der Bins} \end{aligned} \tag{3.4}$$

Diese Methode läßt sich effizient realisieren. Allerdings wirkt sich die zeitliche Abhängigkeit der Daten nachteilig auf die Qualität des Histogramms aus. Man ist eigentlich nur an der räumlichen Verteilung des Schriftzuges interessiert, aber dadurch, daß der Schriftzug zu äquidistanten Zeitpunkten abgetastet wurde, erscheinen in den Bins mehr Punkte, in denen langsamer geschrieben wurde, als in den Bins, in denen Schriftzüge liegen, die schnell geschrieben wurden und somit weniger häufig abgetastet wurden. Ist man an dem Projektionsprofil einer ganzen Textzeile interessiert, dann liefert diese Methode hinreichend gute Ergebnisse.

Methode II. Die zweite, etwas aufwendigere Methode zählt für jedes Histogrammbin, wie oft der Schriftzug es durchläuft.

$$\begin{aligned} H_2^X(n) = |\{i \in \{1, \ldots, N-1\} \mid \quad & y_i \leq (n-0.5) \cdot W + y_{min} < y_{i+1} \\ & \vee\, y_i \geq (n-0.5) \cdot W + y_{min} > y_{i+1}\}| \end{aligned} \tag{3.5}$$

Die Unabhängigkeit der zeitlichen Komponente erkauft man sich mit einer erhöhten Laufzeit. Die Berechnung des horizontalen Projektionsprofils reduziert sich auf das Linien-Schnitt-Problem.

Die erste Ableitung des horizontalen Verteilungshistogramms wird wie folgt approximiert:

$$\Delta H^X(n) = H^X(n+1) - H^X(n) \qquad n \in \{1, \ldots, B-1\} \tag{3.6}$$

In der Implementation wird vor der Berechnung der Ableitung auf das Projektionsprofil ein Mittelwertfilter angewendet. Dadurch wird eine Robustheit gegenüber Rauschen erreicht.

Zusammenfassend sei an dieser Stelle die Berechnung der Referenzlinien angeben:

$$\begin{aligned} y_{\text{Oberlängenlinie}} &= \max_i y_i \\ y_{\text{Grundlinie}} &= ((\arg\max_n \Delta H^X(n)) - 0.5) \cdot W + y_{min} \\ y_{\text{Kernlinie}} &= ((\arg\min_n \Delta H^X(n)) - 0.5) \cdot W + y_{min} \\ y_{\text{Unterlängenlinie}} &= \min_i y_i \end{aligned} \tag{3.7}$$

3.2 Merkmalsgewinnung

Ziel der Merkmalsgewinnung ist es, aus den gemessenen Daten (aufgenommenen Handschriften) einen Satz von Charakteriska zu finden, der die Daten hinreichend genau beschreibt, um sie unterscheiden zu können, aber bezüglich unbedeutender Variationen invariant ist. Zum Beispiel ist es unerheblich, an welcher Position ein Buchstaben geschrieben wurde.

Neben einer normierten Repräsentation der Schriftkurven werden im folgenden Merkmale beschrieben, welche eine Approximation von Eigenschaften der Differentialgeometrie darstellen und somit das idealisierte Verhalten in der Umgebung der Abtastwerte beschreiben. Das lokale Verhalten einer ebenen Kurve kann zum Beispiel durch die Tangente oder die Krümmung charakterisiert werden.

Die absoluten x/y-Koordinaten sind ungeeignet als Merkmale. Sie sind nicht *lage-* und *skalierungsinvariant*. Um Skalierungsinvarianz zu erreichen, wäre die einfachste Methode die einzelnen Zeichen auf dieselbe Höhe zu normieren. Doch dann läßt sich z.B. ein kursives 'l' schwer von einem kursiven 'e' unterscheiden. Sollen kursive Buchstaben erkannt werden, dann muß mindestens ein Merkmal der Tatsache Rechnung tragen, daß es Buchstaben mit Ober- und Unterlängen gibt.

3.2.1 Lage- und Skalierungsinvariante Ortskoordinaten

Eine Möglichkeit für die Gewinnung von lageinvarianten Ortskoordinaten ist, einen eindeutigen Referenzpunkt im Muster zu berechnen und dann das Muster so zu verschieben, daß der Referenzpunkt auf einem fest definierten Punkt zu liegen kommt (etwa dem Ursprung). Eine natürliche Wahl für den Referenzpunkt ist der Mittelwert aller x-Werte, bzw. aller y-Werte.

Durch eine anschließende Normierung durch die Varianz erhält man Skalierungsinvarianz [22]:

$$\tilde{x}_n = \frac{\mu_x - x_n}{\sigma} \tag{3.8}$$

$$\tilde{y}_n = \frac{\mu_y - y_n}{\sigma} \tag{3.9}$$

wobei

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{3.10}$$

und

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((\mu_x - x_i)^2 + (\mu_y - y_i)^2)} \tag{3.11}$$

Guyon et al [9] verwenden anstelle des Mittelwertes als Zentrum eines Buchstaben, die Mitte der Ausdehnung in x-, bzw. y-Richtung:

$$x_0 = \frac{x_{\max} + x_{\min}}{2}, \quad y_0 = \frac{y_{\max} + y_{\min}}{2} \tag{3.12}$$

Der Vorteil der ersten Methode ist die größere Robustheit gegenüber einzelnen Ausreißern am Rand.

3.2.2 Richtung der ersten Ableitung

Ein Schriftzug läßt sich als eine parametrisierte Kurve $\mathbf{x} : I \rightarrow R^2$ in der Ebene beschreiben, wobei $\mathbf{x}(t) = (x(t), y(t))$ die Position der Stiftspitze zum Zeitpunkt t darstellt. Im Folgenden beschreibt $\mathbf{x}(s)$ die Kurve $\mathbf{x}$ parametrisiert nach der Bogenlänge.

Für jedes $s \in S$ mit $\mathbf{x}'(s) \neq 0$ gibt es eine wohldefinierte Gerade, die durch den Punkt $\mathbf{x}(s)$ geht und in Richtung des Vektors $\mathbf{x}'(s)$ zeigt. Diese Gerade heißt *Tangente* an $\mathbf{x}$ bei s [4]. Man beachte, da der Parameter s die Bogenlänge ist, daß der Tangentenvektor die Länge eins hat: $|\mathbf{x}'(s)| = 1$, wobei $|\mathbf{x}'(s)| = \sqrt{(x'(s))^2 + (y'(s))^2}$.

Ein häufig verwendetes Merkmal ist daher die Richtung des Tangentenvektors [20, 9, 16]. Dieses Merkmal ist sowohl lage-, als auch skalierungsinvariant. In [9] wird die Richtung der Ableitung als zweidimensionaler Richtungsvektor $(\cos\theta_n, \sin\theta_n)$ repräsentiert, wobei der Winkel zwischen der Tangente und der x-Achse sich wie folgt berechnet:

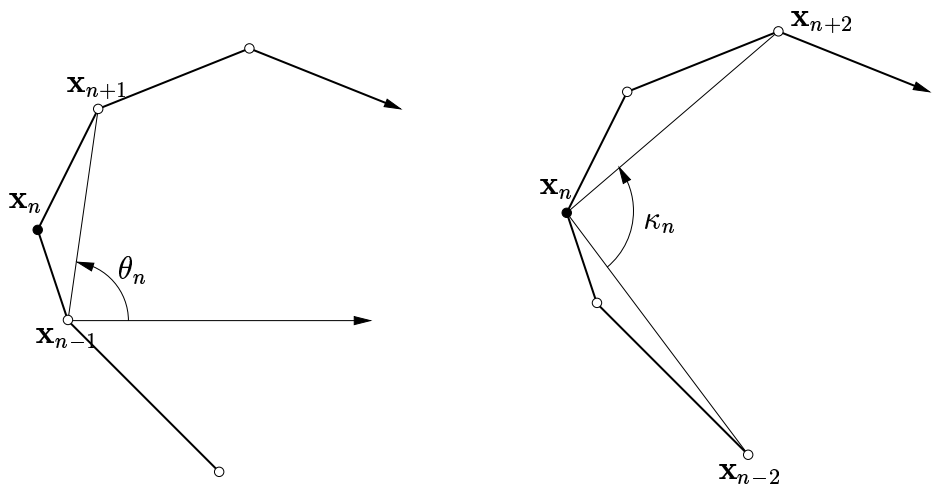
$$\theta_n = \arctan \frac{\Delta y_n}{\Delta x_n} \tag{3.13}$$

wobei für $2 \leq n \leq N-1$ gilt

$$\begin{aligned} \Delta x_n &= x_{n+1} - x_{n-1} \\ \Delta y_n &= y_{n+1} - y_{n-1} \end{aligned} \tag{3.14}$$

und sonst

$$\begin{aligned} \Delta x_1 &= x_2 - x_1 & \qquad \Delta x_N &= x_N - x_{N-1} \\ \Delta y_1 &= y_2 - y_1 & \qquad \Delta y_N &= y_N - y_{N-1} \end{aligned} \tag{3.15}$$



(a) Tangentenwinkel (b) Krümmung

Abbildung 3.4: Approximation des Tangentenwinkels an der Stelle $\mathbf{x}_n$ durch θ_n und das Maß κ_n für die Krümmung an der Stelle $\mathbf{x}_n$.

Alternativ dazu kann der Tangentenwinkel θ_n direkt als Merkmal verwendet werden. Die vektororientierte Methode hat den Vorteil, daß der Wertebereich nicht periodisch ist

und somit keiner gesonderten Betrachtung bei der Berechnung von Distanzen bedarf. Gegen diese Kodierung spricht, daß die Verteilungsdichte von $(\sin\theta, \cos\theta)$ nicht durch eine multivariate Gaußverteilung modelliert werden kann (der Wert von $\cos\theta_n$ bestimmt bis auf das Vorzeichen den Wert von $\sin\theta_n$). Jedoch wird später bei der Konstruktion des Klassifikators die Gaußverteilung der Merkmale gefordert. Auf die Besonderheiten bei der Berechnung des Mittelwertes für einen Winkel wird in Abschnitt 5.2.1 näher eingegangen. Im Folgenden wird die in [20, 16] verwendete zweite Methode benutzt.

Im Vergleich zur normierten Ortskoordinate hängt dieses Merkmal ausschließlich von der Lage des direkten Vorgängers und Nachfolgers im Schriftzug ab. Der Wert von θ_n hängt demnach von der Frequenz ab, mit der in seiner Umgebung abgetastet wurde. Oder da im allgemeinen mit konstanter Frequenz abgetastet wird, ändert sich θ_n abhängig von der Schreibgeschwindigkeit. Abhilfe würde eine Neuabtastung des Schriftzuges mit konstanten Abständen bezüglich der Bogenlänge schaffen.

3.2.3 Krümmung

Die Norm der zweiten Ableitung $|\mathbf{x}''(s)|$ mißt die Änderungsrate zwischen benachbarten Tangenten und der Tangente bei s [4]. $|\mathbf{x}''(s)|$ ist also ein Maß dafür, wie schnell sich die Kurve in der Umgebung von s von der Tangente an s wegdreht. Die Zahl $|\mathbf{x}''(s)| =: \kappa(s)$ heißt die *Krümmung* von $\mathbf{x}$ bei s.

Ein Merkmal, das die Krümmung einer Kurve beschreibt, wäre wünschenswert. Leider ist die Krümmung nicht skalierungsinvariant. Nachteilig ist auch, daß ihr Wertebereich unbeschränkt ist. In [9] wird das Krümmungverhalten einer Kurve durch den Winkel zwischen zwei benachbarten Segmenten beschrieben (siehe Abb. 3.4(b)):

$$\kappa_n = \pi - \theta_{n-1} + \theta_{n+1} \tag{3.16}$$

wobei θ_{n-1} der Tangentenwinkel zum Zeitpunkt $n-1$ und θ_{n+1} der Tangentenwinkel zum Zeitpunkt $n+1$ ist.

3.2.4 Kernhöhennormierter Abstand zur Grundlinie

Dieses Merkmal mißt den kernhöhennormierten vertikalen Abstand des Abtastpunktes zur Grundlinie:

$$\bar{y}_n = h \cdot (y_n - y_{Grundlinie}) \tag{3.17}$$

wobei

$$h = \frac{h_{\text{norm}}}{y_{\text{Grundlinie}} - y_{\text{Kernlinie}}} \tag{3.18}$$

Die relative y-Position wird eigentlich schon mit dem zu Beginn des Abschnitts erläuterten Merkmal $\tilde{y}_n$ beschrieben. Dieses Merkmal eignet sich jedoch nicht für die Worterkennung. Bei der Worterkennung bestehen die Referenzdaten zwar weiterhin aus den einzelnen Buchstaben des Alphabets, aber die Testmuster sind nicht segmentierte Zeichenketten. Da

die Grenzen zwischen den Buchstaben unbekannt sind, müssen die verwendeten Merkmale invariant bzgl. der Position im Wort sein. D.h. ein bestimmter Punkt im Buchstaben a muß dasselbe Merkmal haben, unabhängig davon, ob sich der Buchstabe am Anfang oder am Ende des Wortes befindet. Diese Eigenschaft erfüllen die letzten drei Merkmale θ_n, κ_n und $\bar{y}_n$.

Kapitel 4

Dynamic Time Warping

4.1 Einleitung

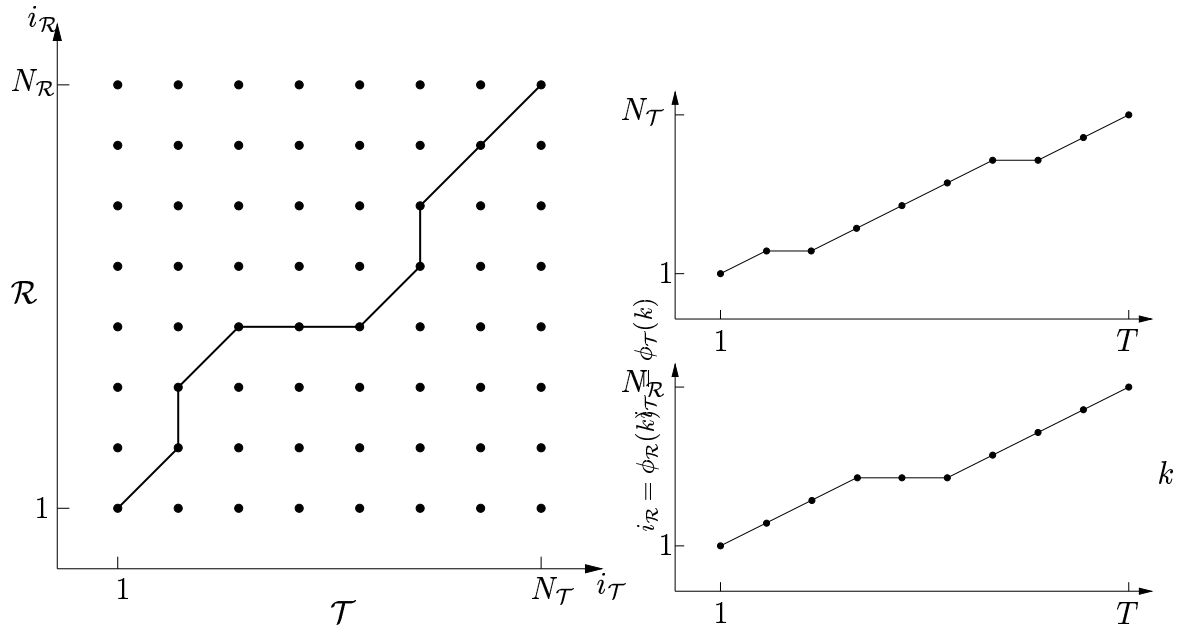
Eine in der Mustererkennung häufig verwendete Methode ist das *Template Matching* [21]. Man sucht zu einem unbekannten Muster (*Testmuster*) dasjenige Muster aus einer gegebenen Menge von Referenzmustern, das dazu am besten „paßt“. Dabei muß zunächst ein Maß gefunden werden, welches ausdrückt, wie ähnlich zwei Muster sind. Da die einzelnen Zeichen als Vektoren gegeben sind, könnte man versucht sein, bekannte Distanzmaße zu verwenden, wie etwa die euklidische Distanz. Da aber die Vektoren unterschiedlich lang sein können, ist dieses Vorgehen nicht möglich. Ferner müssen die zeitlichen Variationen berücksichtigt werden, d.h. die Schreibgeschwindigkeit kann stark unterschiedlich sein.

Es müssen also zunächst korrespondierte Punkte in den zu vergleichenden Mustern gefunden werden, die dann mit einem Ähnlichkeitsmaß verglichen werden können. *Dynamic Time Warping* (DTW), auch bekannt als *Time Alignment* oder *Elastic Matching*, löst dieses Korrespondenzproblem und wird in der Spracherkennung seit langem erfolgreich eingesetzt [15]. DTW-Techniken wurden auch schon in der Handschriftenerkennung angewendet [20, 16].

4.2 Dynamic Time Warping

Gegeben seien zwei Muster $\mathcal{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_{N_\mathcal{T}})$ und $\mathcal{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_{N_\mathcal{R}})$, wobei die Länge $N_\mathcal{T}$ des Musters $\mathcal{T}$ und die Länge $N_\mathcal{R}$ von $\mathcal{R}$ durchaus unterschiedlich sein können. Die Größen $\mathbf{t}_i$ und $\mathbf{r}_j$ sind n-dimensionale Merkmalsvektoren. Auf die Wahl der Merkmale wurde im vorangegangenen Kapitel näher eingegangen.

Die Distanz zwischen $\mathcal{T}$ und $\mathcal{R}$ wird über einer gegebenen Distanzfunktion $d(\mathbf{t}_i, \mathbf{r}_j)$ über dem Merkmalsraum definiert. Für d kann zunächst die euklidische Distanz genommen werden: $d(\mathbf{t}_i, \mathbf{r}_j) = \|\mathbf{t}_i - \mathbf{r}_j\|$. Die zeitlichen Variationen zwischen $\mathcal{T}$ und $\mathcal{R}$ werden mit dem Paar von *Warping-Funktionen* $\phi(m) = (\phi_\mathcal{T}(m), \phi_\mathcal{R}(m))$ modelliert, indem die korrespondierenden Musterindizes $i_\mathcal{T}$ und $i_\mathcal{R}$ anhand einer gemeinsamen Zeitskala k identifiziert werden.



(a) Warping-Pfad

(b) Warping-Funktionen $\phi_\mathcal{T}(k)$ und $\phi_\mathcal{R}(k)$

Abbildung 4.1: Beispiel für eine Zeitnormalisierung zweier Muster $\mathcal{T}$ und $\mathcal{R}$ entlang des Warpingpfades ϕ.

$$\left.\begin{array}{rcl} i_\mathcal{T} & = & \phi_\mathcal{T}(k) \\ i_\mathcal{R} & = & \phi_\mathcal{R}(k) \end{array}\right\} \quad k = 1, 2, \ldots, T \tag{4.1}$$

wobei das Tupel $(i_\mathcal{T}, i_\mathcal{R})$ angibt welche Punkte miteinander verglichen werden sollen. Der Einfachheit halber wird im folgenden ϕ als *Warping-Funktion*, bzw. gemäß der graphischen Interpretation als *Warping-Pfad* bezeichnet (siehe Abbildung 4.1(a)).

Darauf aufbauend läßt sich nun ein globales Distanzmaß $D_\phi(\mathcal{T}, \mathcal{R})$ in Abhängigkeit von der Warping-Funktion ϕ definieren. $D_\phi(\mathcal{T}, \mathcal{R})$ berechnet sich aus der Summe aller lokalen Distanzen entlang des Warping-Pfades ϕ:

$$D_\phi(\mathcal{T}, \mathcal{R}) = \sum_{k=1}^{T} d(\phi_\mathcal{T}(k), \phi_\mathcal{R}(k))\, m(k) \tag{4.2}$$

wobei $m(k)$ ein Pfadgewichtungskoeffizient ist.

Der Abstand $D(\mathcal{T}, \mathcal{R})$ zwischen einem Testmuster $\mathcal{T}$ und einem Referenzmuster $\mathcal{R}$ läßt sich nun als das Minimum von $D_\phi(\mathcal{T}, \mathcal{R})$ über alle Warping-Pfade ϕ definieren:

$$D(\mathcal{T}, \mathcal{R}) := \min_\phi D_\phi(\mathcal{T}, \mathcal{R}) \tag{4.3}$$

Bevor eine effiziente Methode zur Berechnung des in Gleichung 4.3 gestellten Optimierungsproblemes dargestellt wird, soll im nächsten Abschnitt auf notwendige Einschränkungen der erlaubten Warping-Funktionen eingegangen werden.

4.2.1 Einschränkungen der Warping-Funktionen

Einschränkung der Anfangs- und Endpunkte (endpoint constraints): Es wird gefordert, daß der erste Merkmalsvektor $\mathbf{t}_1$ von $\mathcal{T}$ mit dem ersten Merkmalsvektor $\mathbf{r}_1$ von $\mathcal{R}$ verglichen wird. Entsprechendes soll für die Vektoren am Ende der Muster gelten:

$$\begin{array}{lll} \text{Anfangspunkt} & \phi_{\mathcal{T}}(1) = 1, & \phi_{\mathcal{R}}(1) = 1 \\ \text{Endpunkt} & \phi_{\mathcal{T}}(T) = N_{\mathcal{T}}, & \phi_{\mathcal{R}}(T) = N_{\mathcal{R}} \end{array} \tag{4.4}$$

Monotonie (monotonicity conditions): Die zeitliche Reihenfolge soll erhalten bleiben:

$$\phi_{\mathcal{T}}(k+1) \geq \phi_{\mathcal{T}}(k), \quad \phi_{\mathcal{R}}(k+1) \geq \phi_{\mathcal{R}}(k) \tag{4.5}$$

Diese Forderung ist nicht nur aus semantischen Überlegungen notwendig, sondern sie ist ein wichtiger Teil der Voraussetzungen, die eine effiziente Berechnung des DTW-Abstandes ermöglichen, wie im nächsten Abschnitt noch gezeigt wird.

Stetigkeit (local continuity constraints) : Normalerweise trägt jeder Teil eines Schriftzuges zur Erkennung des Zeichens bei. Daher sollte das Weglassen von Teilen des Schriftzuges vermieden werden. Um dem Rechnung zu tragen, wird eine Menge von lokalen Stetigkeitsbedingungen definiert. Eine mögliche Bedingung ist:

$$\phi_x(k+1) - \phi_x(k) \leq 1, \quad \phi_y(k+1) - \phi_y(k) \leq 1 \tag{4.6}$$

Solche Formulierungen der Nebenbedingungen sind meist recht kompliziert und nicht sehr anschaulich. Es ist deshalb angebracht, sie durch eine Menge relativer Pfade $\{\mathcal{P}_1, ..., \mathcal{P}_n\}$ zu beschreiben. Wobei ein Pfad $\mathcal{P}$ durch eine Konkatenation von Bewegungen spezifiziert wird, die jeweils durch ein Paar von Erhöhungen der Koordinatenindizes repräsentiert werden:

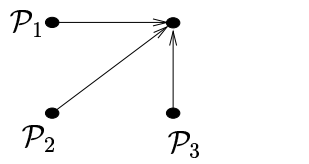
$$\mathcal{P} \rightarrow (p_1, q_1)(p_2, q_2) \ldots (p_T, q_T). \tag{4.7}$$

Abb. 4.2 zeigt zwei einfache Beispiele. Gleichung 4.6 würde in dieser Schreibweise wie folgt aussehen:

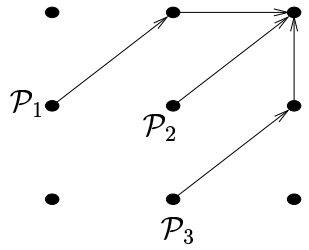
$$\{\mathcal{P}_1 \rightarrow (1,0), \quad \mathcal{P}_2 \rightarrow (1,1), \quad \mathcal{P}_3 \rightarrow (0,1)\} \tag{4.8}$$

Diese lokalen Stetigkeitsbedingungen wurden von Sakoe und Chiba vorgeschlagen [15]. In der Implementierung wurden ausschließlich diese Pfadübergänge verwendet.

Weitere Einschränkungen : Desweiteren können gewisse globale Eigenschaften von den Pfaden verlangt werden oder eine Gewichtung der lokalen Pfade eingeführt werden. Eine ausführliche Beschreibung dieser Einschränkungen wird in [15] gegeben.



$\mathcal{P}_1 \rightarrow (1,0)$
$\mathcal{P}_2 \rightarrow (1,1)$
$\mathcal{P}_2 \rightarrow (0,1)$



$\mathcal{P}_1 \rightarrow (1,1)(1,0)$
$\mathcal{P}_2 \rightarrow (1,1)$
$\mathcal{P}_2 \rightarrow (1,1)(0,1)$

Abbildung 4.2: Zwei einfache Beispiele für lokale Pfadübergänge [15].

4.2.2 Effiziente Lösung der zeitlichen Angleichung

Ein naiver Algorithmus zur Berechnung von Gleichung (4.3) würde entlang jedes erlaubten Warpingpfades die akkumulierten Distanzen berechnen und anschließend das Minimum bestimmen. Da es aber je nach Wahl der lokalen Stetigkeitsbedingungen exponentiell viele Pfade geben kann, ist dies keine praktikable Lösung.

Zur Lösung von Optimierungsproblemen wird häufig die *Dynamische Programmierung* angewendet. Dafür muß sich eine optimale Lösung des zu lösenden Problemes aus mehreren optimal gelösten Teilproblemen zusammensetzen. Diese Eigenschaft wird als *Optimalitätsprinzip* bezeichnet. Ein Algorithmus, der auf dem Prinzip der Dynamischen Programmierung basiert, berechnet die Lösung für jedes Teilproblem nur einmal und speichert diese in einer Tabelle. Damit diese Technik zu einer Effizienzsteigerung gegenüber der „naiven" Auswertung der Rekursionsformel führt, sollten Teilprobleme wiederum Teilprobleme gemeinsam haben [5].

Das Problem der zeitlichen Angleichung kann mit Hilfe der dynamischen Programmierung gelöst werden. Doch dazu muß zunächst Gleichung (4.3) rekursiv formuliert werden.

Die DTW-Distanz zwischen dem Teilmuster $(\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{i_\mathcal{T}})$ von $\mathcal{T}$ und dem Teilmuster $(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{i_\mathcal{R}})$ von $\mathcal{R}$ wird definiert als

$$D_M(i_\mathcal{T}, i_\mathcal{R}) := \min_{\phi} \sum_{k=1}^{T'} d(\phi_\mathcal{T}(k), \phi_\mathcal{R}(k))\, m(k) \tag{4.9}$$

wobei $\phi(T') = (i_\mathcal{T}, i_\mathcal{R})$. Der Pfadgewichtungsfaktor $m(k)$ ermöglicht bestimmte lokale Entscheidungen stärker zu gewichten. Für die folgenden Betrachtungen genügt es, $m(k) = 1\, \forall k = 1, \ldots, T'$ anzunehmen. Der Faktor $m(k)$ kann daher weggelassen werden.

Aufgrund der Einschränkungen bzgl. der Anfangs- und Endpunkte (der Warping-Pfad endet in $(N_\mathcal{T}, N_\mathcal{R})$) gilt:

$$\begin{aligned} D_M(N_\mathcal{T}, N_\mathcal{R}) &= \min_{\phi} \sum_{k=1}^{T} d(\phi_\mathcal{T}(k), \phi_\mathcal{R}(k)) \\ &= D(\mathcal{T}, \mathcal{R}) \end{aligned} \tag{4.10}$$

D_M kann als $N_\mathcal{T} \times N_\mathcal{R}$ Matrix interpretiert werden, wobei der Eintrag $D_M(N_\mathcal{T}, N_\mathcal{R})$ an der Stelle $(N_\mathcal{T}, N_\mathcal{R})$ der gesuchten DTW-Distanz entspricht.

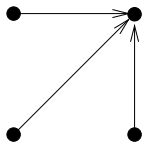
$D(i_\mathcal{T}, i_\mathcal{R})$ läßt sich rekursiv berechnen, indem über alle lokalen Pfade, die im Punkt $(i_\mathcal{T}, i_\mathcal{R})$ enden, minimiert wird:

$$D(i_\mathcal{T}, i_\mathcal{R}) = \min_{(i'_\mathcal{T}, i'_\mathcal{R})} D(i'_\mathcal{T}, i'_\mathcal{R}) + \delta((i'_\mathcal{T}, i'_\mathcal{R}), (i_\mathcal{T}, i_\mathcal{R})) \tag{4.11}$$

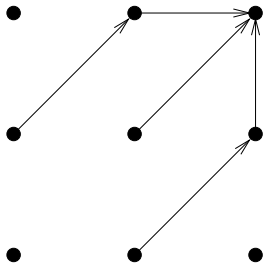
wobei $\delta((i'_\mathcal{T}, i'_\mathcal{R}), (i_\mathcal{T}, i_\mathcal{R}))$ die akkumulierte gewichtete lokale Distanz von $(i'_\mathcal{T}, i'_\mathcal{R})$ nach $(i_\mathcal{T}, i_\mathcal{R})$ bezüglich der lokalen Stetigkeitsbedingungen ist:

$$\delta((i'_\mathcal{T}, i'_\mathcal{R}), (i_\mathcal{T}, i_\mathcal{R})) = \sum_{l=0}^{L_s} d(\phi_\mathcal{T}(T' - l), \phi_\mathcal{R}(T' - l)) \tag{4.12}$$

mit L_s = Länge des lokalen Pfades von $(i'_\mathcal{T}, i'_\mathcal{R})$ nach $(i_\mathcal{T}, i_\mathcal{R})$. Abbildung 4.3 verdeutlicht dies anhand der in Abb. 4.2 eingeführten Stetigkeitsbedingungen.



$$\min \left\{ \begin{array}{l} D_M(i_\mathcal{T} - 1, i_\mathcal{R}) + d(i_\mathcal{T}, i_\mathcal{R}), \\ D_M(i_\mathcal{T} - 1, i_\mathcal{R} - 1) + d(i_\mathcal{T}, i_\mathcal{R}), \\ D_M(i_\mathcal{T}, i_\mathcal{R} - 1) + d(i_\mathcal{T}, i_\mathcal{R}) \end{array} \right\}$$



$$\min \left\{ \begin{array}{l} D_M(i_\mathcal{T} - 2, i_\mathcal{R}) + d(i_\mathcal{T} - 1, i_\mathcal{R}) + d(i_\mathcal{T}, i_\mathcal{R}), \\ D_M(i_\mathcal{T} - 1, i_\mathcal{R} - 1) + d(i_\mathcal{T}, i_\mathcal{R}), \\ D_M(i_\mathcal{T}, i_\mathcal{R} - 2) + d(i_\mathcal{T}, i_\mathcal{R} - 1) + d(i_\mathcal{T}, i_\mathcal{R}) \end{array} \right\}$$

Abbildung 4.3: Zwei Beispiele für lokale Stetigkeitsbedingungen mit Gewichtung der Pfadübergänge und den daraus resultierenden Rekursionsformeln.

Implementation

Die in Gleichung 4.11 angebene Rekursionsformel läßt sich direkt als Algorithmus angeben. Die Kernidee dabei ist, daß sich der Algorithmus die bereits berechneten akkumulierten Distanzen der Teilmuster $(\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{i_\mathcal{T}})$ und $(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{i_\mathcal{R}})$ in $D_M(i_\mathcal{R}, i_T)$ merkt und dann in der Rekursion darauf zurückgreifen kann. Daraus ergibt sich dann eine asymptotische Laufzeit von $O(N_\mathcal{T} \cdot N_\mathcal{R} \cdot L)$, mit L = Anzahl der lokalen Pfade.

An dieser Stelle sei zusammenfassend der Algorithmus zur Berechnung der DTW-Distanz angegeben. Der Einfachheit halber werden die in (4.6) beschriebenen lokalen Stetigkeitsbedingungen verwendet.

function *DTW-Algorithmus*
Initialisierung:
$D_M(0,0) = 0$
$D_M(0,i) = \infty \quad \forall i = 1,2,\ldots,N_\mathcal{R}$
$D_M(i,0) = \infty \quad \forall i = 1,2,\ldots,N_\mathcal{T}$
Rekursion:
for $i_\mathcal{T} = 1,\ldots,N_\mathcal{T}$
for $i_\mathcal{R} = 1,\ldots,N_\mathcal{R}$

$$D_M(i_\mathcal{T}, i_\mathcal{R}) = \min \left\{ \begin{array}{l} D_M(i_\mathcal{T}-1, i_\mathcal{R}) + d(i_\mathcal{T}, i_\mathcal{R}), \\ D_M(i_\mathcal{T}-1, i_\mathcal{R}-1) + d(i_\mathcal{T}, i_\mathcal{R}), \\ D_M(i_\mathcal{T}, i_\mathcal{R}-1) + d(i_\mathcal{T}, i_\mathcal{R}) \end{array} \right\}$$

end
end
return $D_M(N_\mathcal{T}, N_\mathcal{R})$
end

Da die Berechnung der lokalen Distanz $d(i_\mathcal{T}, i_\mathcal{R})$ mitunter sehr aufwendig sein kann und gerade bei komplexeren Modellierungen der Pfadübergänge häufig benötigt wird, ist es vorteilhaft, wenn die lokalen Distanzen $d(i_\mathcal{T}, i_\mathcal{R})$ für alle Tupel $(i_\mathcal{T}, i_\mathcal{R})$ im voraus berechnet und in einer Matrix gespeichert werden.

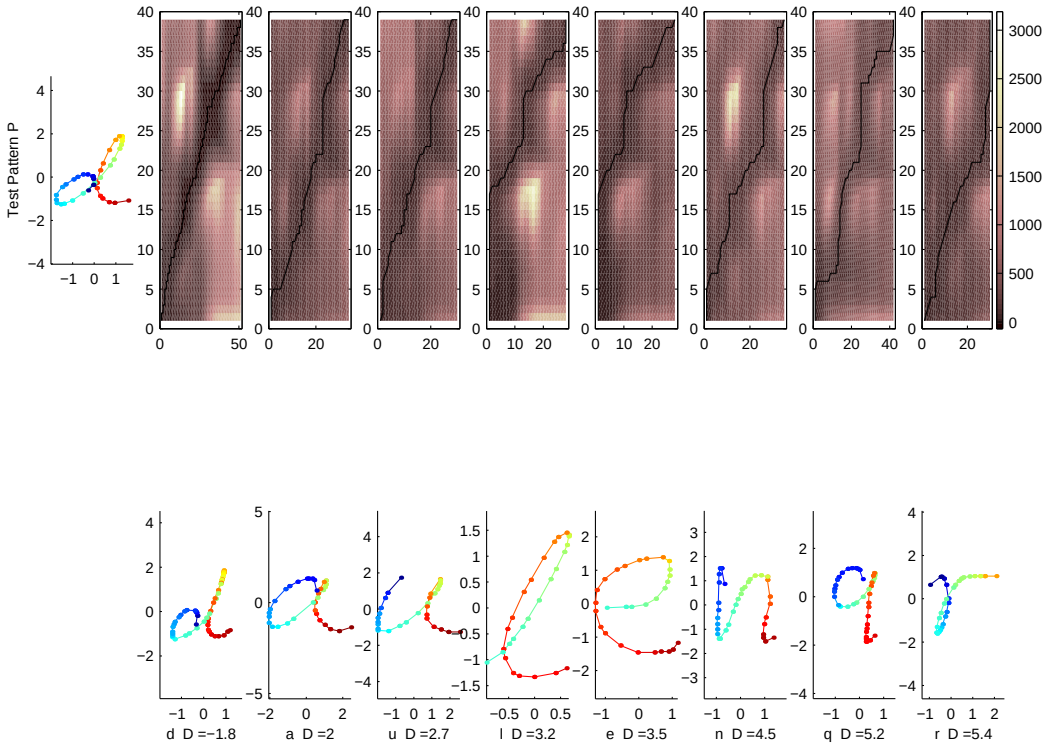


Abbildung 4.4: Links oben ist das Testmuster aufgetragen. Die Schaubilder in der untersten Zeile zeigen die in aufsteigendem DTW-Abstand sortierten Referenzmuster. Die Referenzmuster sind mit der Zeichenklasse und dem jeweiligen DTW-Abstand zum Testmuster beschriftet. Oberhalb der Referenzmuster sind die entsprechenden DTW-Matrizen D_M dargestellt. Der optimale Warping-Pfad ist in Schwarz eingezeichnet.

4.3 Verbessertes lokales Distanzmaß

Im vorangegangenen Abschnitt wurde der Dynamic Time Warping Algorithmus als eine Methode zur Berechnung des Abstandes zwischen zeitlich verzerrten Mustern vorgestellt. Dabei wurde ein lokales Distanzmaß d vorausgesetzt. Wählt man für d etwa die euklidische Metrik, so werden Informationen über die statistischen Verteilungen ignoriert, die zur Verbesserungen der Klassifikation beitragen könnten.

In diesem Abschnitt soll nun ausgehend von einem allgemeinen Klassifikationsansatz ein lokales Distanzmaß entwickelt werden, das die Eigenschaften der Verteilungen der Merkmalsvektoren abhängig von der Position im Buchstaben berücksichtigt. Die Idee dabei ist, daß zum Beispiel die Merkmale am Anfang eines Buchstabens stark variieren und daher eine Abweichung vom Referenzmuster weniger gravierend ist, als an einer Stelle, an der sie nur sehr wenig vom Referenzmuster abweichen.

4.3.1 Herleitung der lokalen Metrik

MAP Ansatz

Man kann Klassifikationsprobleme als Optimierungsprobleme auffassen: der Klassifikator soll sich bzgl. einer gegebenen Gütefunktion optimal entscheiden. Soll etwa die Wahrscheinlichkeit einer Fehlentscheidung bei vorliegendem Testmuster $\mathcal{T}$ minimiert werden, so entspricht dies der Maximierung der *a posteriori Wahrscheinlichkeit*, also der Wahrscheinlichkeit, daß es sich bei einem gegebenen Muster $\mathcal{T}$ um einen Vertreter der Klasse l handelt. Diese Klasse von Klassifikatoren nennt man *Maximum a posteriori Klassifikatoren* oder *Bayes-Klassifikatoren.*

Im folgenden soll vorausgesetzt werden, daß die klassenspezifische Verteilungsdichte $P(\mathcal{T} \mid l)$ und die a priori Wahrscheinlichkeit $P(l)$ bekannt sind. Folgt man dem MAP-Ansatz, ist demnach die Klasse $\hat{l}$ gesucht, die die a posteriori Wahrscheinlichkeit maximiert:

$$\hat{l} = \arg\max_{l}\{P(l \mid \mathcal{T})\} \tag{4.13}$$

Durch zweifache Anwendung der Bayes'schen Regel ($P(A,B) = P(A|B) \cdot P(B)$) ergibt sich:

$$\begin{aligned} \arg\max_{l}\{P(l \mid \mathcal{T})\} &= \arg\max_{l}\left\{\tfrac{P(l,\mathcal{T})}{P(\mathcal{T})}\right\} \\ &= \arg\max_{l}\left\{\tfrac{P(l)P(\mathcal{T} \mid l)}{P(\mathcal{T})}\right\} \end{aligned} \tag{4.14}$$

Da die Wahrscheinlichkeit $P(\mathcal{T})$ unabhängig von der Klasse l ist, genügt es, das Produkt der a priori Wahrscheinlichkeit $P(l)$ und der klassenspezifischen Verteilungsdichte $P(\mathcal{T} \mid l)$ zu maximieren:

$$\hat{l} = \arg\max_{l}\{P(l)P(\mathcal{T} \mid l)\} \tag{4.15}$$

Es soll nun angenommen werden, daß sich die zeitlichen Variationen zwischen einem Muster $\mathcal{T}$ und dem entsprechenden Referenzmuster mit der gegeben Menge von Warpingfunktionen Φ modellieren lassen. Demnach ist die Vereinigung der voneinander unabhängigen Warpingfunktionen $\bigcup_{\phi\in\Phi}\phi$ ein sicheres Ereignis und für die klassenspezifische Verteilungsdichte gilt unter Verwendung des Satzes der totalen Wahrscheinlichkeiten $P(\mathcal{T} \mid l) = \sum\limits_{\phi\in\Phi} P(\mathcal{T}, \phi \mid l)$. Es läßt sich nun Gleichung 4.15 umformen zu:

$$\hat{l} = \arg\max_{l} \left\{ P(l) \sum_{\phi\in\Phi} P(\mathcal{T}, \phi \mid l) \right\} \tag{4.16}$$

Optimaler Warping-Pfad

Die im zu maximierenden Term enthaltene Wahrscheinlichkeit $\sum\limits_{\phi\in\Phi} P(\mathcal{T}, \phi \mid l)$ unterscheidet sich von einer anderen Größe, der sogenannten Viterbi-Bewertung $\max\limits_{\phi} P(\mathcal{T}, \phi \mid l)$ genau dann, wenn mehr als ein Pfad ϕ existiert mit $P(\mathcal{T}, \phi \mid l) \neq 0$. Schukat-Talamazzini [17] führt an, daß diese Größen sehr stark korreliert sind. In den meisten Worterkennnern in der Spracherkennung wird $\max\limits_{\phi} P(\mathcal{T}, \phi \mid l)$ daher als Ersatzgröße für $P(\mathcal{T} \mid l)$ verwendet. Merhav und Ephraim [13] weisen für die Differenz zwischen $P(\mathcal{T} \mid l)$ und der Viterbi-Bewertung $P(\mathcal{T} \mid l) - \max\limits_{\phi} P(\mathcal{T}, \phi \mid l)$ eine obere Schranke nach und demonstrieren dies anhand eines Beispiels aus der Spracherkennung.

Es gelte also die Annahme, daß $\max_{\phi}\{P(\mathcal{T}, \phi \mid l)\}$ die Summe über alle Warpingpfade so dominiert, daß $\hat{l}$ als Argument des Maximums erhalten bleibt. Dann gilt:

$$\hat{l} = \arg\max_{l} \left\{ P(l) \max_{\phi}\{P(\mathcal{T}, \phi \mid l)\} \right\} \tag{4.17}$$

$\max\limits_{\phi}\{P(\mathcal{T}, \phi \mid l)\}$ entspricht der Wahrscheinlichkeit für den wahrscheinlichsten Warping-Pfad[1]. D.h. es muß nun der wahrscheinlichste Warping-Pfad, bzw. dessen Wahrscheinlichkeit berechnet werden. Dies geschieht analog zur Herleitung des Viterbi-Algorithmus [8]. Zunächst wird Gleichung 4.17 unter Verwendung der Bayes'schen Regel umgeformt zu:

$$\hat{l} = \arg\max_{l} \left\{ P(l) \max_{\phi} \{P(\phi \mid l) \cdot P(\mathcal{T} \mid \phi, l)\} \right\} \tag{4.18}$$

Produkt unabhängiger Normalverteilungen

Für die Wahrscheinlichkeit $P(\mathcal{T}|\phi, l)$ wird angenommen, daß sie sich als Produkt von einander unabhängigen Einzelverteilungen schreiben läßt:

$$P(\mathcal{T} \mid \phi, l) = \prod_{m=1}^{T} P\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)} \mid \phi_{\mathcal{R}}(m), l\right) \tag{4.19}$$

[1] Mit $\hat{\phi} = \arg\max_{\phi}\{P(\mathcal{T}, \phi \mid l)\}$ erhält man den wahrscheinlichsten Warping-Pfad, da $\max\limits_{\phi} P(\mathcal{T}, \phi \mid l) = \max\limits_{\phi}\{P(\phi \mid \mathcal{T}, l) \cdot P(\mathcal{T} \mid l)\} = \max\limits_{\phi} P(\phi \mid \mathcal{T}, l)$.

Für (4.18) ergibt sich daraus folgende Gleichung:

$$\begin{aligned}\hat{l} &= \arg\max_{l}\left\{P(l)\max_{\phi}\left\{P(\phi \mid l)\cdot\prod_{m=1}^{T}P\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)} \mid l,\phi_{\mathcal{R}}(m)\right)\right\}\right\}\\ &\overset{-\ln}{=} \arg\min_{l}\left\{-\ln P(l)+\min_{\phi}\left\{-\ln P(\phi \mid l)+\sum_{m=1}^{T}-\ln P\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)} \mid l,\phi_{\mathcal{R}}(m)\right)\right\}\right\}\end{aligned} \tag{4.20}$$

Aufgrund der Monotonie des Logarithmus ändert sich das Argument des Maximums nicht. Die anschließende Multiplikation mit -1 macht aus dem Maximierungsproblem eine Suche nach dem Minimum über alle möglichen Pfade und Klassen.

Nimmt man nun für die Verteilung der Merkmale $P\left(\mathbf{t}_{\phi_{\mathcal{T}}}(m) \mid l,\phi_{\mathcal{R}}(m)\right)$ eine Gaußsche Normalverteilung an, dann ergibt sich:

$$\begin{aligned}&-\ln P\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)} \mid l,\phi_{\mathcal{R}}(m)\right)\\ =\;& -\ln\left(\frac{1}{(2\pi)^{n/2}\left|\mathbf{K}^{l}_{\phi_{\mathcal{R}}(m)}\right|^{1/2}}\exp\left(-\tfrac{1}{2}\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)}-\boldsymbol{\mu}^{l}_{\phi_{\mathcal{R}}(m)}\right)^{T}\left(\mathbf{K}^{l}_{\phi_{\mathcal{R}}(m)}\right)^{-1}\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)}-\boldsymbol{\mu}^{l}_{\phi_{\mathcal{R}}(m)}\right)\right)\right)\\ =\;& \tfrac{n}{2}\ln(2\pi)+\tfrac{1}{2}\ln\left|\mathbf{K}^{l}_{\phi_{\mathcal{R}}(m)}\right|+\underbrace{\frac{1}{2}\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)}-\boldsymbol{\mu}^{l}_{\phi_{\mathcal{R}}(m)}\right)^{T}\left(\mathbf{K}^{l}_{\phi_{\mathcal{R}}(m)}\right)^{-1}\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)}-\boldsymbol{\mu}^{l}_{\phi_{\mathcal{R}}(m)}\right)}_{=:\;\delta\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)},\boldsymbol{\mu}^{l}_{\phi_{\mathcal{R}}(m)},\mathbf{K}^{l}_{\phi_{\mathcal{R}}(m)}\right)}\\ =:\;& d^{w}_{\mathbf{K}^{l}_{\phi_{\mathcal{R}}(m)}}\left(\mathbf{t}_{\phi_{\mathcal{T}}(m)},\boldsymbol{\mu}^{l}_{\phi_{\mathcal{R}}(m)}\right)\end{aligned} \tag{4.21}$$

Darauf aufbauend wird ein lokales Distanzmaß $\tilde{d}_{\mathcal{T},\mathcal{R}^l}(i,j)$ zwischen dem i-ten Element des Testmusters $\mathcal{T}$ und dem j-ten Element des Referenzmusters $\mathcal{R}$ wie folgt definiert:

$$\tilde{d}_{\mathcal{T},\mathcal{R}^l}(i,j) := \frac{n}{2}\ln(2\pi)+\frac{1}{2}\ln|\mathbf{K}^{l}_{j}|+\delta(\mathbf{t}_i,\boldsymbol{\mu}^{l}_{j},\mathbf{K}^{l}_{j}) \tag{4.22}$$

Lösung mit Hilfe des DTW

Jetzt läßt sich Gleichung 4.20 schreiben als:

$$\hat{l} = \arg\min_{l}\left\{-\ln P(l)+\min_{\phi}\left\{-\ln P(\phi \mid l)+\sum_{m=1}^{T}\tilde{d}_{\mathcal{T},\mathcal{R}^l}(\phi_{\mathcal{T}}(m),\phi_{\mathcal{R}}(m))\right\}\right\} \tag{4.23}$$

Nun fließt eine weitere Annahme ein: $P(\phi \mid l) = \text{konstant}$, d.h. für eine gegebene Klasse sind alle Pfade gleich wahrscheinlich. Somit vereinfacht sich die Gleichung zu:

$$\hat{l} = \arg\min_{l}\left\{-\ln P(l)+\min_{\phi}\left\{\sum_{m=1}^{T}\tilde{d}_{\mathcal{T},\mathcal{R}^l}(\phi_{\mathcal{T}}(m),\phi_{\mathcal{R}}(m))\right\}\right\} \tag{4.24}$$

In

$$\begin{aligned}&\min_{\phi}\left\{\sum_{m=1}^{T}\tilde{d}_{\mathcal{T},\mathcal{R}^l}(\phi_{\mathcal{T}}(m),\phi_{\mathcal{R}}(m))\right\}\\ =:\;& \min_{\phi} D_{\phi}(\mathcal{T},\mathcal{R}^l)\end{aligned} \tag{4.25}$$

erkennt man sofort die DTW-Distanz analog zur Gleichung 4.2. Nun läßt sich das hergeleitete Distanzmaß in Gleichung 4.25 effizient mit dem DTW-Algorithmus bestimmen.

Ist die a priori Wahrscheinlichkeit $P(l)$ für alle Klassen l konstant, so erhält man einen Minimum-Abstandsklassifikator basierend auf der DTW-Distanz:

$$\hat{l} = \arg\min_{l} \left\{ \min_{\phi} D_{\phi}(\mathcal{T}, \mathcal{R}^l) \right\} \tag{4.26}$$

4.3.2 Zusammenfassung der Modellannahmen

Schritt für Schritt sind in den vorangegangenen Überlegungen eine Menge von Modellannahmen getroffen worden. Somit ist das hergeleitete Abstandsmaß (4.22) nur dann optimal im Bayes'schen Sinne, wenn diese Annahmen erfüllt sind, bzw. im praktischen Fall annähernd erfüllt sind. Aus diesem Grund sind sie nachfolgend noch einmal explizit zusammengefaßt:

1. Mit der Menge aller erlaubten Warping-Pfade Φ lassen sich alle zeitlichen Variationen modellieren. Die in Gleichung 4.6 formulierten lokalen Stetigkeitsbedingungen erlauben beliebige zeitliche Variationen, verbieten allerdings Auslassungen.

2. $\arg\max_{l}\{P(l)\sum_{\phi} P(\mathcal{T}, \phi \mid l)\} = \arg\max_{l}\{P(l)\max_{\phi}\{P(\mathcal{T}, \phi \mid l)\}\}$. Man beachte, daß nicht die Gleichheit der Summe der Pfad-Wahrscheinlichkeiten über alle Pfade $\sum_{\phi} P(\mathcal{T}, \phi \mid l)$ und der Wahrscheinlichkeit $\max_{\phi}\{P(\mathcal{T}, \phi \mid l)$ für den wahrscheinlichsten Pfad gefordert wird. Es wird lediglich gefordert, daß die Klasse mit dem wahrscheinlichsten Pfad auch die a posteriori Wahrscheinlichkeit $P(\mathcal{T} \mid l)$ maximiert.

3. Für $P(\phi \mid l)$ wurde angenommen, daß alle Warping-Pfade einer Klasse gleich wahrscheinlich sind.

   Eine andere Möglichkeit, diese Wahrscheinlichkeit zu modellieren, wäre für ϕ gegeben l die Markoveigenschaft $P(\phi(m) \mid \phi(m-1), \ldots, \phi(1), l) = P(\phi(m) \mid \phi(m-1), l)$ vorauszusetzen. Dann berechnet sich $P(\phi \mid l)$ durch die Multiplikation der Wahrscheinlichkeiten der lokalen Pfadübergänge:

$$\begin{aligned} P(\phi \mid l) &= P(\phi(T) \mid \phi(1), \ldots, \phi(T-1), l)) \cdot P(\phi(1), \ldots, \phi(T-1) \mid l) \\ &= P(\phi(T) \mid \phi(T-1)) \cdot P(\phi(1), \ldots, \phi(T-1) \mid l) \\ &\vdots \\ &= \prod_{m=1}^{T} P(\phi(m) \mid \phi(m-1), l) \end{aligned} \tag{4.27}$$

   wobei $P(\phi(m) \mid \phi(m-1), l)$ die Wahrscheinlichkeit für den Übergang von $\phi(m-1)$ nach $\phi(m)$ ist (mit $m > 1$). Für $m = 1$ wird $P(\phi(1) \mid \phi(0), l) := 1$ definiert. Diese Annahmen eingesetzt in Gleichung 4.23 ergibt anstelle von Gleichung 4.25 folgende Formulierung, die sich mit Hilfe des DTW-Algorithmus lösen läßt:

$$
\begin{aligned}
& \min_{\phi} \left\{ -\ln P(\phi \mid l) + \sum_{m=1}^{T} \tilde{d}_{\mathcal{T},\mathcal{R}^l}(\phi_{\mathcal{T}}(m), \phi_{\mathcal{R}}(m)) \right\} \\
= & \min_{\phi} \left\{ -\ln \left(\prod_{m=1}^{T} P(\phi(m) \mid \phi(m-1), l) \right) + \sum_{m=1}^{T} \tilde{d}_{\mathcal{T},\mathcal{R}^l}(\phi_{\mathcal{T}}(m), \phi_{\mathcal{R}}(m)) \right\} \\
= & \min_{\phi} \left\{ \sum_{m=1}^{T} -\ln P(\phi(m) \mid \phi(m-1), l) + \sum_{m=1}^{T} \tilde{d}_{\mathcal{T},\mathcal{R}^l}(\phi_{\mathcal{T}}(m), \phi_{\mathcal{R}}(m)) \right\} \\
= & \min_{\phi} \underbrace{\left\{ \sum_{m=1}^{T} -\ln P(\phi(m) \mid \phi(m-1), l) + \tilde{d}_{\mathcal{T},\mathcal{R}^l}(\phi_{\mathcal{T}}(m), \phi_{\mathcal{R}}(m)) \right\}}_{D'_{\phi}(\mathcal{T},\mathcal{R}^l)}
\end{aligned}
\tag{4.28}
$$

Die Pfadübergangswahrscheinlichkeiten $P(\phi(m) \mid \phi(m-1), l)$ müssen wie die übrigen Parameter bei der Modellberechnung geschätzt werden.

4. Die klassenspezifische Verteilung $P(\mathcal{T} \mid l)$ läßt sich als Sequenz voneinander unabhängiger Verteilungen darstellen.

5. Die Verteilung der Merkmale entspricht einer Gaußverteilung. Die tatsächlichen Verteilungen sind unbekannt. In Kapitel 6 wird untersucht, ob diese Annahme für die dieser Arbeit vorliegenden Daten gerechtfertigt ist.

4.3.3 Interessante Sonderfälle

Abschließend soll auf zwei Sonderfälle aufmerksam gemacht werden. Es wird angenommen, daß die Kovarianzmatrizen zu jedem Zeitpunkt $i_{\mathcal{R}^l}$ gleich sind.

1. Spezialfall $K_j^l = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$

Unter der Annahme, daß die Merkmale unkorreliert im Raum verteilt sind, läßt sich das lokale Abstandsmaß vereinfachen zu:

$$
\begin{aligned}
\tilde{d}_{\mathcal{T},\mathcal{R}^l}(i,j) &= \frac{n}{2}\ln(2\pi) + \frac{1}{2}\ln|\mathbf{K}_j^l| + \delta(\mathbf{t}_i, \boldsymbol{\mu}_j^l, \mathbf{K}_j^l) && (4.29) \\
&= \underbrace{\frac{n}{2}\ln(2\pi) + \sum_{c=1}^{n} \ln \sigma_c}_{\text{konstant}} + \frac{1}{2}\sum_{c=1}^{n} \frac{1}{\sigma_c^2}(t_{ic} - \mu_{j_c}^l)^2 && (4.30)
\end{aligned}
$$

wobei σ_c die Standardabweichung des c-ten Merkmals ist.

Wenn die erlaubten Warpingpfade eine konstante Länge haben (etwa bzgl. der Länge des Testmusters), dann läßt sich der konstante Summand vor die Summe in Gleichung 4.25 ziehen. Er kann dann auch ganz aus dem Minimierungsausdruck entfernt werden. Somit ergibt sich folgende lokale Distanz:

$$
d^{\sigma}_{\mathcal{T},\mathcal{R}^l}(i,j) := \sum_{c=1}^{n} \frac{1}{\sigma_c^2}(t_{ic} - \mu_{j_c}^l)^2 \tag{4.31}
$$

Dies entspricht dem quadratischen euklidischen Abstand, wobei vorher jede Dimension des Differenzenvektors bezüglich seiner Standardabweichung normiert wurde.

2. Spezialfall $K_j^l = \alpha \cdot I$

Sind die Varianzen für jedes Merkmal außerdem noch gleich, dann erhält man als lokales Abstandsmaß:

$$\begin{aligned} \tilde{d}_{\mathcal{T},\mathcal{R}^l}(i,j) &= \frac{n}{2}\ln(2\pi) + \frac{1}{2}\ln|\mathbf{K}_j^l| + \delta(\mathbf{t}_i, \boldsymbol{\mu}_j^l, \mathbf{K}_j^l) && (4.32) \\ &= \frac{n}{2}\ln(2\pi) + \frac{n\ln\alpha}{2} + \frac{1}{2\alpha}\|\mathbf{t}_i - \boldsymbol{\mu}_j^l\|^2 && (4.33) \end{aligned}$$

Eine äquivalente Formulierung ist:

$$2\alpha \cdot \tilde{d}_{\mathcal{T},\mathcal{R}^l}(i,j) = n\alpha(\ln 2\pi + \ln\alpha) + \|\mathbf{t}_i - \boldsymbol{\mu}_j^l\|^2 \tag{4.34}$$

Wenn wie eben wieder angenommen werden kann, daß die erlaubten Warpingpfade eine konstante Länge haben, kann der konstante Faktor und Summand aus dem Optimierungsausdruck herausgenommen werden. Man erhält dann das häufig verwendete Quadrat des *euklidischen Abstandes*:

$$d^e_{\mathcal{T},\mathcal{R}^l}(i,j) := \|\mathbf{t}_i - \boldsymbol{\mu}_j^l\|^2 \tag{4.35}$$

4.4 Zusammenfassung der Distanzfunktion

In der folgenden Definition werden die benötigten Parameter für die lokale Distanzfunktion zusammengefaßt.

Definition: Ein *Modell der Klasse l* wird definiert als Tripel $\mathcal{M}^l := (\mathcal{R}^l, \mathcal{A}^l, \mathcal{W}^l)$ bestehend aus dem Referenzmuster $\mathcal{R}^l$, der Sequenz additiver Gewichte $\mathcal{A}^l$ und multiplikativer Gewichte $\mathcal{W}^l$:

$$\begin{array}{lcll} \mathcal{R}^l & := & (\mathbf{r}_1^l, \mathbf{r}_2^l, \ldots, \mathbf{r}_{N_{\mathcal{R}^l}}^l) & \text{mit } \mathbf{r}_i^l \in R^n \\ \mathcal{A}^l & := & (\mathbf{a}_1^l, \mathbf{a}_2^l, \ldots, \mathbf{a}_{N_{\mathcal{R}^l}}^l) & \text{mit } \mathbf{a}_i^l \in R \\ \mathcal{W}^l & := & (\mathbf{W}_1^l, \mathbf{W}_2^l, \ldots, \mathbf{W}_{N_{\mathcal{R}^l}}^l) & \text{mit } \mathbf{W}_i^l \in R^{n\times n} \end{array} \tag{4.36}$$

wobei entsprechend zu den vorangegangenen Untersuchungen für die Parameter gelten soll:

$$\begin{aligned} \mathbf{r}_i^l &= \boldsymbol{\mu}_i^l \\ \mathbf{a}_i^l &= \tfrac{n}{2}\ln(2\pi) + \tfrac{1}{2}|\mathbf{K}_i^l| \\ \mathbf{W}_i^l &= \tfrac{1}{2}\left(\mathbf{K}_i^l\right)^{-1} \end{aligned} \tag{4.37}$$

Definition: Die Distanz zwischen einem Testmuster $\mathcal{T}$ und dem Modell $\mathcal{M}^l$ ist wie folgt definiert:

$$\begin{aligned} D(\mathcal{T},\mathcal{M}^l) &:= \min_{\phi} \sum_{m=1}^{T} d_{\mathcal{T},\mathcal{M}^l}(\phi_{\mathcal{T}}(m), \phi_{\mathcal{R}}(m)) \\ &= \min_{\phi} D_{\phi}(\mathcal{T},\mathcal{M}^l) \end{aligned} \tag{4.38}$$

mit

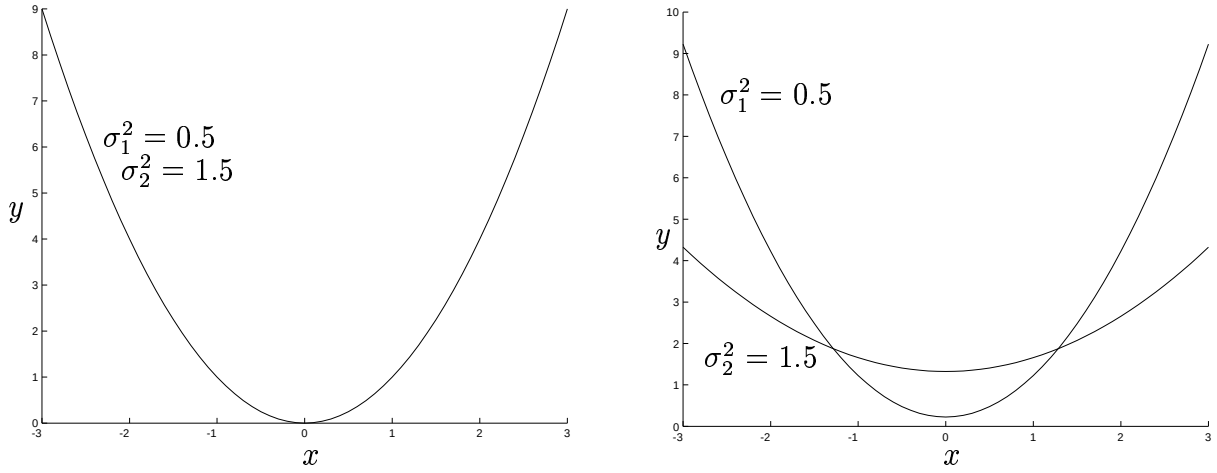
$$d_{\mathcal{T},\mathcal{M}^l}(i,j) := \mathbf{a}_i^l + (\mathbf{t}_i - \mathbf{r}_j^l)^T \cdot \mathbf{W}_j^l \cdot (\mathbf{t}_i - \mathbf{r}_j^l) \tag{4.39}$$

4.4.1 Anschauung

In diesem Abschnitt soll die im vorangegangenen Abschnitt hergeleitete lokale Distanz anhand eines einfachen Beispiel veranschaulicht werden. In Abbildung 4.5 wird das Quadrat des euklidischen Abstands $\|x-\mu\|^2$ dem gewichteten Abstandsmaß $d^w_{\mathbf{K}}(x,\mu) = \frac{n}{2}\ln(2\pi) + \ln|\mathbf{K}| + \delta(x,\mu,\mathbf{K})$ aus Gleichung 4.21 gegenübergestellt. Betrachtet wird der eindimensionale Fall ($n = 1$). Untersucht wird das Verhalten der beiden Abstandsmaße für zwei Verteilungen, wobei in beiden Fällen für den Mittelwert $\mu = 0$ gelte und für die Varianz jeweils $\sigma_1^2 = 0.5$ bzw. $\sigma_2^2 = 1.5$ gilt.

Das Quadrat des euklidischen Abstandes ist unabhängig von der Varianz und verhält sich somit für beide Verteilungen gleich. Sind x und μ identisch, so ist der Fehler Null. Er wächst quadratisch mit dem Abstand zum Mittelwert μ.

Das quadratische Ansteigen des Fehlers läßt sich auch bei $d^w_{\mathbf{K}}(x,\mu)$ beobachten. Allerdings wächst der Fehler für die kleinere Varianz σ_1^2 schneller, als für die größere Varianz σ_2^2. Dieses Verhalten entspricht der Vorstellung, daß niedrige Varianzen signifikante Merkmale beschreiben und somit ein starkes Abweichen stärker „bestraft“ wird. In Abbildung 4.5(b) ist die Auswirkung des additiven Terms $\frac{n}{2}\ln(2\pi) + \frac{1}{2}\ln|\mathbf{K}|$ zu erkennen. Je größer die Varianz ist, desto weiter entfernt sich die Fehlerkurve von der x-Achse. Ohne diesen Offset würden beim DTW-Algorithmus signifikante Regionen (Punkte im Referenzmuster mit kleiner Varianz) gemieden werden, was keineswegs erwünscht wäre. Diese Verhalten konnte auch in [16] bei einem anderen lokalen Distanzmaß festgestellt werden.



(a) $\|x-\mu\|^2$ (b) $d^w_{\sigma^2}(x,\sigma^2)$

Abbildung 4.5: Vergleich zwischen dem Quadrat des euklidischen Abstandes und dem gewichteten lokalen Abstandmaß. Untersucht wird das Verhalten der beiden Abstandsmaße für zwei Verteilungen, wobei in beiden Fällen für den Mittelwert $\mu = 0$ gelte und für die Varianz jeweils $\sigma_1^2 = 0.5$ bzw. $\sigma_2^2 = 1.5$ gilt.

Kapitel 5

Modellbildung

Dieses Kapitel widmet sich der Berechnung der Buchstabenmodelle. Da die Modellparameter von unbekannten statistischen Größen abhängen, müssen diese in geeigneter Weise aus der gegeben Trainingsmenge geschätzt werden. Für jedes Zeichen $z \in \Sigma$ des Alphabets Σ ist eine Menge $\mathcal{B}^z = \{\mathcal{T}_1^z, \ldots, \mathcal{T}_{N_z}^z\}$ von Trainingsbeispielen für das Zeichen z gegeben. Aus diesen wird für jedes Zeichen mindestens ein Modell berechnet. Manchmal reicht ein Modell jedoch nicht aus, um eine Zeichenklasse vollständig zu beschreiben. Dies ist dann der Fall, falls für dasselbe Zeichen unterschiedliche Schreibweisen existieren. Im ersten Schritt muß die Trainingsmenge in diese unterschiedlichen Ausprägungen unterteilt werden. Anschließend werden aus diesen Teilmengen die Modellparameter bestimmt. Abb. 5.1 illustriert das schematische Vorgehen zur Bestimmung der Modelle für eine Buchstabenklasse.

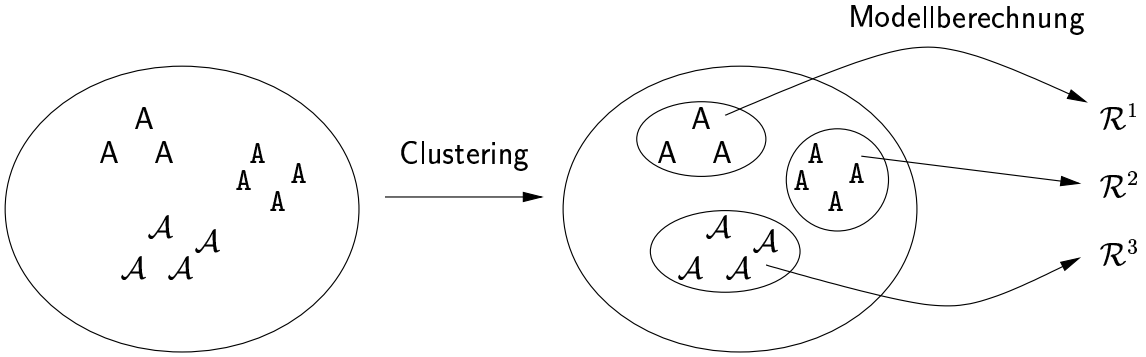


Abbildung 5.1: Ähnliche Ausprägungen derselben Buchstaben werden zu einem Cluster zusammengefaßt. Aus den Beispielen in einem Cluster werden die Referenzmuster und Gewichte extrahiert.

Der nächste Abschnitt beschäftigt sich mit dem ersten Schritt, dem Identifizieren der ähnlichen Ausprägungen (*Clustering*). Im darauffolgenden Abschnitt wird ein Verfahren zur Schätzung der Modellparameter angegeben.

5.1 Clustering

Bevor ein Verfahren zur Bestimmung der Cluster angegeben wird, werden, der Notation von Theodoridis folgend, einige Begriffe und Distanzmaße eingeführt [21].

5.1.1 Definitionen

Clustering

Unter *Clustering* (cluster: Anhäufung) versteht man die Partitionierung einer Menge von Elementen $X = \{x_1, x_2, \ldots, x_N\}$ in eine Menge $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ nichtleerer, disjunkter Mengen, so daß gilt:

$$\begin{aligned} &C_i \neq \emptyset \quad \text{für} \quad i = 1, \ldots, m \\ &\cup_{i=1}^{m} C_i = X \\ &C_i \cap C_j = \emptyset \quad \text{für} \quad i \neq j \quad \text{und} \quad i, j = 1, \ldots, m \end{aligned} \tag{5.1}$$

Distanzfunktionen zwischen Clustern

Die Elemente sollen später so gruppiert werden, daß möglichst ähnliche Elemente zu einem Cluster zusammengefaßt werden. Diese Ähnlichkeit wird durch ein gegebenes Distanzmaß $d(\cdot, \cdot)$ bestimmt. Ferner ist man an einem Distanzmaß $\mathcal{D}$ interessiert, welches die Ähnlichkeit zwischen zwei Clustern C_i und C_j beschreibt. Aufbauend auf d und den Elementen von C_i und C_j sind folgende Definitionen von Distanzmaßen zwischen Mengen möglich [21]:

- *max proximity function:*

$$\mathcal{D}_{max}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \tag{5.2}$$

  $\mathcal{D}_{max}$ ist vollständig definiert durch das Paar $(x, y) \in (C_i, C_j)$, das (bzgl. d) am unähnlichsten ist.

- *min proximity function:*

$$\mathcal{D}_{min}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \tag{5.3}$$

  Entsprechend gilt für $\mathcal{D}_{min}$, daß die Distanz nur von dem ähnlichsten Paar $(x, y) \in (C_i, C_j)$ bestimmt wird.

- *average proximity function:*

$$\mathcal{D}_{avg}(C_i, C_j) = \frac{1}{N_i N_j} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y) \tag{5.4}$$

  wobei N_i (N_j) die Anzahl der Elemente im Cluster C_i (C_j) ist. Im Gegensatz zu den beiden ersten Funktionen tragen hier alle Elemente von C_i und C_j zur Distanz bei.

Punktrepräsentanten

Wie später noch gezeigt wird, wird zur Bestimmung der Modellparameter ein charakteristischer Vertreter aus einem Cluster benötigt. Häufig verwendete Punktrepräsentanten sind [21]:

- Das *Mittelwertzentrum* (*mean center*) $m_{\text{mean}} \in C$ ist definiert als:

$$\sum_{y \in C} d(m_{\text{mean}}, y) \leq \sum_{y \in C} d(z, y), \qquad \forall z \in C \tag{5.5}$$

  m_{mean} minimiert die akkumulierten Distanzen zu allen anderen Elementen in C.

- Das *Medianzentrum* (*median center*) $m_{\text{med}} \in C$ ist definiert als:

$$\text{med}(d(\{m_{\text{med}}, y) | y \in C\}) \leq \text{med}(\{d(z, y) | y \in C\}), \quad \forall z \in C, \tag{5.6}$$

  wobei $\text{med}(S)$ den Median der Menge S liefert. Dieser Punktrepräsentant wird häufig verwendet, wenn es sich bei d nicht um eine Metrik handelt [21].

5.1.2 Hierarchisches Clustering

Es existiert eine Vielfalt an verschiedenen Verfahren zur Berechnung von Clustern. Hierarchische Clustering-Verfahren gehören aufgrund ihrer konzeptionellen Einfachheit zu den bekanntesten Methoden [7].

Hierarchische Clustering Algorithmen erzeugen eine Sequenz von Partitionierungen über den N Elementen der Menge X. Die erste Partitionierung der Sequenz teilt die Menge in N Cluster auf, d.h. jedes Cluster enthält genau ein Element. Die nächste Partitionierung besteht aus $N - 1$ Cluster, die dritte aus $N - 2$, usw. Die N-te und somit letzte Partitionierung in der Sequenz besteht aus genau einem Cluster, das alle Elemente enthält. Ferner hat die Sequenz die Eigenschaft, daß für den Fall, daß zwei Elemente x und x' in demselben Cluster enthalten sind, sie sich auch in den darauffolgenden Partitionierungen in demselben Cluster befinden.

Jedes Hierarchische Clustering kann in einem binären Baum, dem *Dendrogramm*, dargestellt werden. Jedem der N Blätter wird genau ein Element zugeordnet. Ein innerer Knoten p repräsentiert ein Cluster. Dieses Cluster besteht aus den Blättern des Teilbaumes mit der Wurzel p. Die Höhe eines Knotens p wird bestimmt durch die Distanz zwischen den Clustern, die durch die beiden Söhne von p repräsentiert werden. Je höher ein Knoten ist, desto verschiedener sind die in diesem Cluster zusammengefaßten Elemente.

Schneidet man nun das Dendrogramm in einer bestimmten Höhe c ab, so erhält man ein Clustering $\mathcal{C}$ mit der Eigenschaft, daß $\mathcal{D}(C_i, C_j) < c$ für alle Cluster $C_i, C_j \in \mathcal{C}$.

Algorithmen

Die Verfahren zur Berechnung von Hierarchischen Clustern können in zwei Klassen unterteilt werden: *anhäufende* (engl.: agglomerative) und *teilende* (engl.: divisive) Verfahren. Erstere sollen im folgenden erläutert werden.

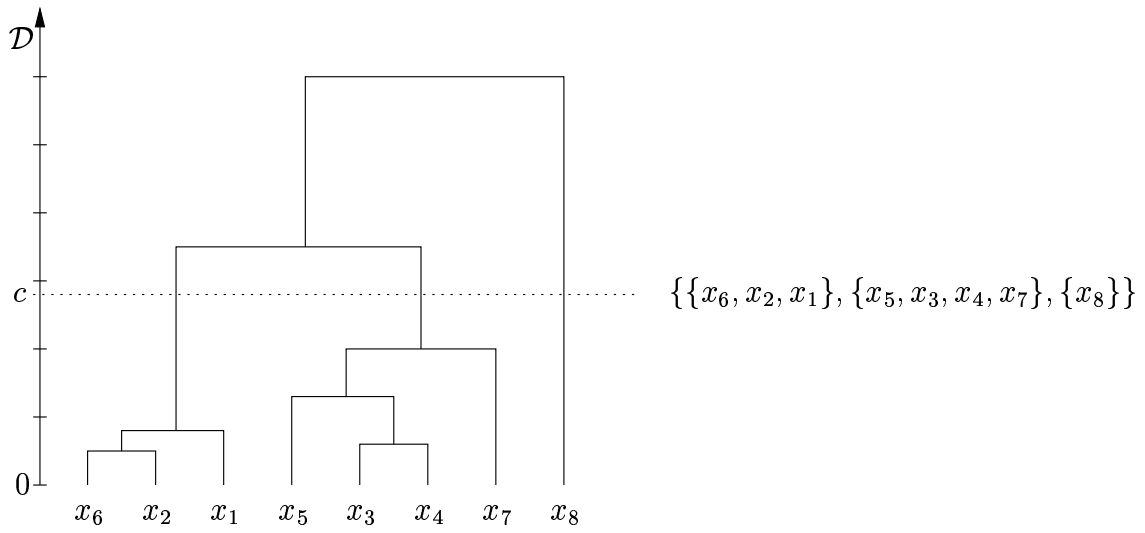


Abbildung 5.2: Das Dendrogramm für die Menge $\{x_1, x_2, \ldots, x_8\}$. Auf der rechten Seite wird die aus dem Schnitt in der Höhe c resultierende Partitionierung gezeigt.

Anhäufende Verfahren fangen mit N einelementigen Clustern an und verschmelzen Schritt für Schritt Paare von Clustern. Teilende Verfahren gehen in umgekehrter Reihenfolge vor. Zu Beginn befinden sich alle Elemente in einem Cluster. In jedem Schritt wird ein Cluster in zwei neue Cluster aufgeteilt.

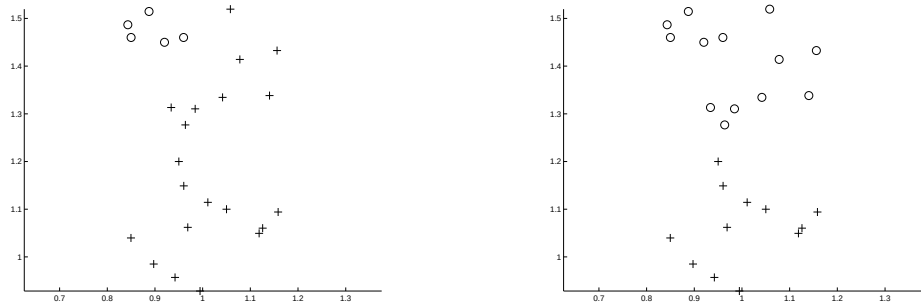
Sei ein Distanzmaß $\mathcal{D}$ zwischen zwei Mengen und ein Schwellwert c gegeben. Der Kern eines anhäufenden Verfahrens besteht dann aus der folgenden Schleife:

while $c > \min\limits_{i,j,i\neq j} \mathcal{D}(C_i, C_j)$
Verschmelze $C_{i_{\min}}$ *und* $C_{j_{\min}}$, *wobei* $(i_{\min}, j_{\min}) = \arg\min\limits_{(i,j),i\neq j} \mathcal{D}(C_i, C_j)$
end

In jedem Schritt wird das bzgl. $\mathcal{D}$ ähnlichste Cluster-Paar verschmolzen. Dies macht deutlich, daß die Wahl von $\mathcal{D}$ großen Einfluß auf die Gestalt der Cluster hat, während durch die Wahl von c maßgeblich Einfluß auf die Anzahl der resultierenden Cluster genommen werden kann.

Ein anhäufendes Verfahren, das $\mathcal{D}_{min}$ als Distanzfunktion zwischen zwei Mengen verwendet, wird *Single-Linkage* Algorithmus genannt oder auch *Nächster-Nachbar* Algorithmus [7]. Der Name leitet sich aus dem Verhalten des Algorithmus ab, gemäß $\mathcal{D}_{min}$ das Paar von Clustern (C_i, C_j) zu verschmelzen, das die am nächsten benachbarten Elemente enthält.

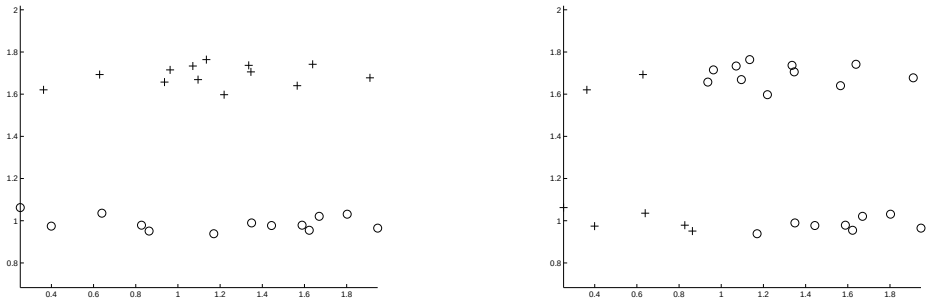
Abb. 5.3(a) und 5.4(a) illustrieren das Verhalten dieses Algorithmus. Aufgetragen sind Vektoren in der euklidischen Ebene, als Abstandsmaß d wurde die euklidische Metrik verwendet. In Abb. 5.3(a) sind zwei Anhäufungen von Punkten zu erkennen, mit einem die Anhäufungen verbindenden Ausreißer dazwischen. Dieser Ausreißer führt dazu, daß ein großes, langgezogenes Cluster und ein kleines, kompaktes Cluster entstehen. Diese Eigenschaft, langgezogene Gruppierungen zu bevorzugen, ist typisch für Single-Linkage Algorithmen. Für Abb. 5.3(a) hätte man eher eine Partitionierung in zwei kompakte Cluster



(a) $\mathcal{D}_{min}$ (b) $\mathcal{D}_{max}$

Abbildung 5.3: Zu erkennen sind zwei kompakte Cluster, die durch eine Kette verbunden sind. (a) Der Single-Linkage Algorithmus findet ein großes und ein kleines Cluster. (b) Der Complete-Linkage Algorithmus identifiziert die beiden in etwa gleich großen Cluster.

erwartet. Das Beispiel aus Abb. 5.4(a) hingegen entspricht dem intuitiv richtigen Clustering.



(a) $\mathcal{D}_{min}$ (b) $\mathcal{D}_{max}$

Abbildung 5.4: Aufgetragen sind zwei langgezogene Cluster. (a) Die intuitiv richtige Partitionierung wird vom Single-Linkage Algorithmus gefunden. (b) Der Complete-Linkage Algorithmus versucht zwei kompakte Cluster zu finden.

Wird hingegen $\mathcal{D}_{max}$ als Distanzfunktion verwendet, spricht man vom *Complete-Linkage* Algorithmus [7]. Die Distanz zwischen zwei Clustern wird durch die größte Entfernung zwischen zwei Elementen aus diesen Clustern bestimmt. Dieser Algorithmus erzeugt kompakte Cluster. Das ist von Vorteil, wenn wie in Abb. 5.3(b), für die vorliegende Datenverteilung eine kompakte Form und in etwa die gleiche Größe der Gruppierungen angenommen werden kann. Jedoch werden die langgezogenen Punktwolken in Abb. 5.4(b) nicht korrekt erkannt.

$\mathcal{D}_{avg}$ ist ein Kompromiß aus $\mathcal{D}_{min}$ und $\mathcal{D}_{max}$ [7] und wird im folgenden als *Average-Linkage* Algorithmus bezeichnet. Neben diesen drei Distanzfunktionen sind viele weitere denkbar [21]. Welche dieser Distanzmaße sich am ehesten für die Identifizierung der verschiedenen Modellklassen eignet, wird in Abschnitt 6.3.3 näher untersucht.

Im folgenden werden noch einige Aspekte besprochen, welche bei der praktischen Umsetzung des Clusterings von Bedeutung sind.

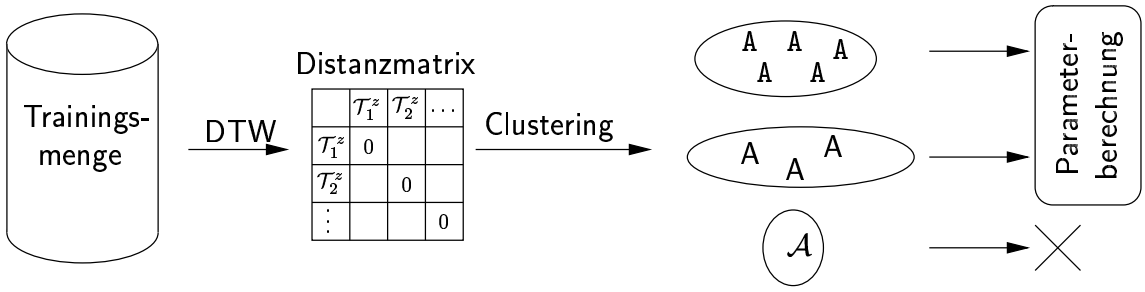


Abbildung 5.5: Für jedes Beispiel einer Zeichenklasse wird der DTW-Abstand zu jedem Beispiel derselben Zeichenklasse berechnet und in der Distanzmatrix abgespeichert. Das Clustering-Verfahren identifiziert die ähnlichen Schreibweisen. Cluster, die zuwenig Beispiele enthalten, werden verworfen. Für die restlichen Cluster werden die Modellparameter berechnet.

Die Berechnung des DTW-Abstandes ist relativ aufwendig und da er häufig benötigt wird, ist es angebracht, alle Abstände zwischen jedem Beispiel im voraus zu berechnen und in einer Matrix, der Distanzmatrix, abzuspeichern.

Im allgemeinen ist der DTW-Abstand nicht symmetrisch und daher ist die resultierende Distanzmatrix ebenfalls nicht symmetrisch. Die hier vorgestellten Methoden sind aber nur auf einem symmetrischen Abstandsmaß wohldefiniert. Beschränkt man sich bei den erlaubten lokalen Übergängen auf symmetrische lokale Stetigkeitsbedingungen, wie etwa Gl. 4.6, und verwendet das ungewichtete lokale Distanzmaß ($\mathbf{a}_i^l = 0$ und $\mathbf{W}_i^l = I$, wobei I die Einheitsmatrix ist), dann erhält man Symmetrie.

Eine offene Frage ist die Bestimmung der optimalen Anzahl der Cluster, bzw. des optimalen Schwellwertes c für jede Zeichenklasse. In der Implementation muß der Schwellwert manuell gesetzt werden und gilt für alle zu trainierenden Zeichenklassen.

Nicht jedes Cluster ist geeignet, um daraus ein Modell zu berechnen. Befinden sich zuwenig Beispiele in einem Cluster, so lassen sich die statistischen Parameter aus Abschnitt 4.4 nur ungenau bestimmen. Daher werden alle Cluster, deren Mächtigkeit unterhalb eines bestimmen Schwellwertes liegt, aus der Modellberechnung entfernt. Wird dieser Schwellwert klein genug gewählt, so ist davon auszugehen, daß es sich dabei um ein „schlechtes“ Beispiel handelt und deshalb nicht als Modell geeignet wäre.

5.2 Berechnung der Modellparameter

Das im vorangegangenen Abschnitt beschriebene Clustering-Verfahren partitioniert die Trainingsmenge in die verschiedenen Schreibweisen der Zeichen. Für jedes so erhaltene Cluster C^l müssen nun die Modellparameter geschätzt werden.

Im folgenden wird ein iteratives Verfahren zur Berechnung der Modellparameter angeben, welches in abgeänderter Form als *Viterbi-Training* oder *segmentweise K-means* bekannt ist [17]. In einem ersten Schritt werden die Modellparameter initialisiert. Als Referenzmuster wird ein Clusterrepräsentant von C^l gewählt. Dies kann entweder das Mittelwertzentrum oder Medianzentrum sein (vgl. Abbschnitt 5.1.1). Die Gewichtsmatrizen $\mathbf{W}_j^l$

werden auf die Einheitsmatrix gesetzt und entsprechend werden die additiven Gewichte mit 0 intialisiert. Dieses initiale Modell wird mit $\mathcal{M}_0^l$ bezeichnet.

Nun beginnt der iterative Teil des Verfahrens. Für jedes Trainingsmuster $\mathcal{X}_i \in C^l$ wird der optimale Warpingpfad $\hat{\phi}_i$ bzgl. des Modells $\mathcal{M}_r^l$ berechnet (in der ersten Iteration ist $r = 0$). Anhand der Warpingpfade lassen sich die Merkmalsvektoren bestimmen, die mit dem Punkt $\mathbf{r}_j^l$ im Referenzmuster des Modells $\mathcal{M}_r^l$ korrespondieren. Für jedes $\mathbf{r}_j^l$ wird so die Menge korrespondierender Merkmalsvektoren bestimmt, aus der dann die Momente erster und zweiter Ordnung bestimmt werden. Aus diesen lassen sich unter Verwendung der Formeln aus Abschnitt 4.4 die Modellparameter für $\mathcal{M}_{r+1}^l$ berechnen.

Die Menge $P_C^{\mathcal{M}^l}(j)$ faßt die mit dem j-ten Referenzpunkt $\mathbf{r}_j^l$ korrespondierenden Merkmalsvektoren zusammen:

$$P_C^{\mathcal{M}^l}(j) := \{\mathbf{x}^i_{\hat{\phi}_\mathcal{T}(k)} \mid \begin{array}{l} \mathcal{X}_i \in C, \\ \hat{\phi} = (\hat{\phi}_\mathcal{T}, \hat{\phi}_{\mathcal{R}^l}), \\ \hat{\phi} = \arg\min_\phi D_\phi(\mathcal{X}_i, \mathcal{M}^l), \\ \hat{\phi}_\mathcal{R}(k) = j\} \end{array} \tag{5.7}$$

wobei $\mathcal{X}_i = (\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}^i_{N_{\mathcal{X}_i}})$.

Die Iteration wird nach einer vorgegebenen Anzahl von Iterationsschritten abgebrochen.

function *Iterative Modellberechnung*
begin
Initialisiere Referenzmodell $\mathcal{M}_0^l$:
$\mathcal{R}^l = \mathcal{X}_{med} \in C^l$
$\mathbf{W}_j^l = I$ für $j = 1, \ldots, N_{\mathcal{R}^l}$ und wobei I die Einheitsmatrix ist
$\mathbf{a}_j^l = 0$ für $j = 1, \ldots, N_{\mathcal{R}^l}$

Iteriere die Berechnung des Referenzmuster und der Gewichte:
for r = 0,1,2, ...
Berechne Gewichte und Referenzmuster entlang der optimalen Warpingpfade bzgl $\mathcal{M}_r^l$:
for $j = 1, 2, \ldots, N_{\mathcal{R}^l}$
$\mathbf{r}_j^l = \text{mean}\left(P_{C^l}^{\mathcal{M}^l}(j)\right)$
$\mathbf{W}_j^l = \frac{1}{2}\text{cov}\left(P_{C^l}^{\mathcal{M}^l}(j)\right)^{-1}$
$\mathbf{a}_j^l = \frac{n}{2}\ln(2\pi) + \frac{1}{2}\text{det}\left(\text{cov}\left(P_{C^l}^{\mathcal{M}^l}(j)\right)\right)$
end
$\mathcal{M}_{r+1}^l = (\mathcal{R}^l, \mathcal{A}^l, \mathcal{W}^l)$
end
end

5.2.1 Berechnung der Momente erster und zweiter Ordnung

Mittelwert

Für die Berechnung des Modells ist es notwendig, den Erwartungswert μ (Moment erster Ordnung) der Merkmalsvektoren zu schätzen. Dieser wird üblicherweise durch das arith-

metische Mittel über alle Vektoren der Trainingsmenge approximiert [18]:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} \alpha_i \tag{5.8}$$

Aufgrund der periodischen Eigenschaften der Winkelwerte läßt sich die oben angebene Vorschrift nicht für die Schätzung des Erwartungswertes für Winkel anwenden. Abb. 5.6 macht dies deutlich. Abhängig davon, in welchem Wertebereich die Wertemenge repräsentiert wird, ergibt sich ein anderer Mittelwert. $\hat{\mu}_1$ entspricht dem intuitiv richtigen mittleren Winkel, wohingegen $\hat{\mu}_2$ eine schlechte Approximation für den zu erwartenden Winkel darstellt.

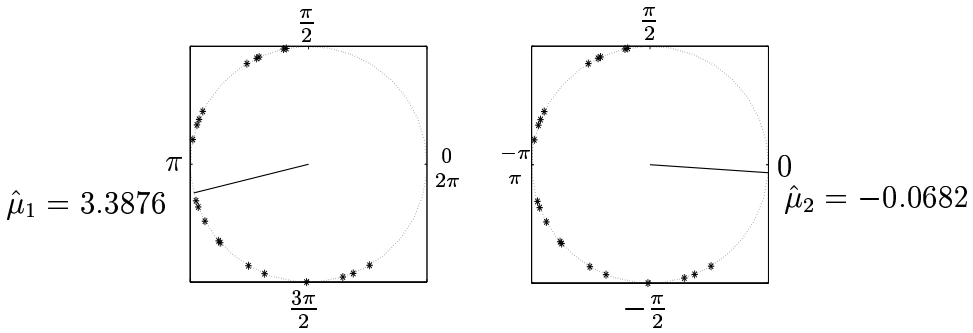


Abbildung 5.6: In Abb. (a) und (b) sind dieselben Werte aufgetragen. In Abb. (a) werden sie im Wertebereich $[0, 2\pi)$ und in Abb. (b) im Wertebereich $[-\pi, \pi)$ dargestellt.

Ein erster Ansatz umgeht dieses Problem und interpretiert die Winkelwerte als Richtungen von normierten Vektoren in der Ebene. Dabei werden die Werte entsprechend in die komplexe Ebene konvertiert. Der Winkel des arithmetischen Mittels wird dann als Mittelwert für die Winkel interpretiert:

$$\hat{\alpha} = \text{angle} \left(\frac{1}{N} \sum_{k=1}^{N} e^{i\alpha_k} \right) \tag{5.9}$$

$\hat{\alpha}$ entspricht der mittleren Richtung, was aber nicht dasselbe wie der mittlere Winkel ist.

Das von Döker et al [6] vorgestellte Verfahren zur Berechnung des mittleren Winkels nutzt die Eigenschaft des Erwartungswertes μ aus, daß er die erwartete Abweichung zu der Zufallsvariable X minimiert: $\mu = \arg\min_{\mu'} E[|X - \mu'|^2]$.

Angenommen die Winkel werden im halboffenen Intervall $[0, 2\pi)$ dargestellt, dann wird für jede mögliche Verschiebung $c \in (0, 2\pi]$ des Intervalls nach $[0 + c, 2\pi + c]$ der Mittelwert und die Standardabweichung berechnet. Derjenige Mittelwert, der die minimale Standardabweichung impliziert, ist der gesuchte. Seien die Winkel $\alpha_1, \ldots, \alpha_N$ aufsteigend sortiert. Dann ergibt sich für alle Verschiebungen c mit $\alpha_i < c \leq \alpha_{i+1}$ die gleiche Standardabweichung und derselbe Mittelwert. Somit genügt es die Sprungstelle nur N mal zu verschieben. An dieser Stelle ist die Funktion im Pseudo-Code zusammengefaßt:

function $\bar{\mu} = meanAngle(\ \{\alpha_1, \alpha_2, \ldots, \alpha_N\}\ )$
begin
  for $i = 1, \ldots, N$
    $c_i := \alpha_i$
    $\alpha'_j := \alpha_j - c_i \bmod 2\pi \quad j = 1, \ldots, N$
    $\bar{\mu}_i := \frac{1}{N} \sum \alpha'_j$
    $s_i^2 := \frac{\sum(\alpha'_j - \bar{\mu}_i)^2}{N}$
  end
  $i_{min} := \arg\min_i s_i$
  $\bar{\mu} := (\bar{\alpha}_{i_{min}} + \alpha_{i_{min}}) \bmod 2\pi$
end

Der mit dieser Methode errechnete Mittelwert ist nicht immer eindeutig. So ist der mittlere Winkel von $0, \frac{\pi}{2}, \pi$ und $\frac{3\pi}{2}$ einer der folgenden vier: $\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}$ oder $\frac{7\pi}{4}$. Interpretiert man diese Werte wiederum als Winkel komplexer Zahlen auf dem Einheitskreis, erhält man 0 als Mittelwert.

Der Nachteil dieses Verfahrens ist die hohe Laufzeitkomplexität ($O(N^2)$ im Vergleich zur linearen Laufzeit der ersten Methode). Versuche, den Wertebereich zunächst so zu verschieben, daß sich die, mit der ersten Methode berechnete, mittlere Richtung in der Mitte des Intervalls befindet und darauf wie gewohnt das arithmetische Mittel zu berechnen, haben in der Regel zu guten Approximationen geführt. Der Fehler lag meist in der Größenordnung von ca. 10^{-10}. Allerdings konnte der Fehler bis zu 5% betragen, wenn die Werte stark gestreut waren.

Kovarianzmatrix

Die Kovarianzmatrix wird anhand der Formel

$$K = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \tag{5.10}$$

geschätzt. Für nichtperiodische Größen ist dies wiederum problemlos durchführbar. Für den periodischen Wertebereich von Winkeln gilt folgendes:

Ist der Mittelwert der Winkel bekannt, so läßt sich die Kovarianzmatrix einfach berechnen. Da das zweite Moment die mittelwertfreien Eigenschaften einer statischen Verteilung beschreibt, können die Werte der Winkel so verschoben werden, daß der Mittelwert auf π liegt, ohne dabei den Wert der Kovarianzmatrix zu ändern. Und da die Winkel im Wertebereich $[0, 2\pi)$ liegen und der Mittelwert μ nun genau auf π liegt, gilt $|\alpha_i - \mu| < \pi$, d.h. die Periodizität der Werte muß nicht gesondert beachtet werden und somit muß das Verfahren zur Berechnung der Kovarianzmatrix außer der Translation um $(\pi - \hat{\mu})$ nicht an die periodischen Verhältnisse angepaßt werden.

5.2.2 Berechnung der Modellparameter W und a

Im vorangegangen Abschnitt wurde erläutert, wie sich die statistischen Parameter aus der Trainingsmenge schätzen lassen. Im folgenden soll nun beschrieben werden, wie die

Berechnungsvorschrift der Modellparameter **W** und **a** umgesetzt wurde.

Sei K die zu invertierende Kovarianzmatrix und $\lambda_1, \ldots, \lambda_n$ die absteigend sortierten Eigenwerte von K. Dann läßt sich K schreiben als

$$K = U \cdot \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \cdot U^T \tag{5.11}$$

wobei der i-te Spaltenvektor u_i von U der zu λ_i gehörende Eigenvektor von K ist. Aufgrund der Orthogonalität von U läßt sich die Inverse von K schreiben als:

$$K^{-1} = U \cdot \begin{pmatrix} 1/\lambda_1 & & \\ & \ddots & \\ & & 1/\lambda_n \end{pmatrix} \cdot U^T \tag{5.12}$$

Im Hauptachsenraum läßt sich die Funktionsweise der Gewichtsmatrix am besten beschreiben. Je größer der Eigenwert, desto stärker ist die Streuung entlang der entsprechenden Hauptachse und um so schwächer wird ein Abweichen vom Mittelwert „bestraft“. Ist der Eigenwert dagegen klein, wird der Abstand zum Mittelwert vergrößert.

Liegen die Beispiele sehr nah beieinander, so sind die Eigenwerte der Kovarianzmatrix sehr klein. Eine geringe Varianz konnte häufig auf eine zu kleine Menge von Trainingsbeispielen zurückgeführt werden. Um eine zu starke Gewichtung zu verhindern, die sonst dazu führen könnte, daß ein Punkt im Merkmalsvektor einen zu großen Anteil an der Gesamtdistanz erhält, wird eine modifizierte Berechnung der Gewichtsmatrix angewendet, die eine zu große Gewichtung verhindert:

$$\mathbf{W} = \frac{1}{2} \cdot U \cdot \begin{pmatrix} 1/\tilde{\lambda}_1 & & \\ & \ddots & \\ & & 1/\tilde{\lambda}_n \end{pmatrix} \cdot U^T \tag{5.13}$$

wobei $\tilde{\lambda}_i = \max\{\lambda_i, \lambda_{\min}\}$.

Das additive Gewicht wird daraufhin im Hauptachsen-System wie folgt berechnet:

$$\mathbf{a} = \frac{n}{2} \ln(2\pi) + \frac{1}{2} \prod_{i=1}^{n} \tilde{\lambda}_i \tag{5.14}$$

Kapitel 6

Ergebnisse

In diesem Kapitel werden die Ergebnisse dieser Arbeit unter folgenden Aspekten zusammengefaßt. Zunächst werden die statistischen Eigenschaften der verwendeten Merkmale untersucht. Anschließend soll festgestellt werden, inwieweit die in Abschnitt 4.3.1 hergeleitete lokale Distanz zu Verbesserungen der Klassifikationsrate führt. Darauf folgt eine empirische Analyse der im vorangegangenen Kapitel vorgestellten Algorithmen zur Modellberechnung. Abschließend werden die Klassifikationsergebnisse präsentiert, die mit verschiedenen Merkmalskombination erreicht wurden.

6.1 Statistische Eigenschaften der Merkmale

Bei der Herleitung des lokalen Distanzmaßes wurden einige Modellannahmen getroffen. So wurde in Gleichung 4.21 für die Merkmalsvektoren eine Gaußsche Normalverteilung angenommen. In diesem Abschnitt soll nun untersucht werden, ob diese Annahme gerechtfertigt ist. D.h. ob die Verteilung der Merkmale durch ein Gaußsche Normalverteilung approximiert werden kann. Dabei wurde wie folgt vorgegangen.

Das in Abschnitt 5.2 vorgestellte Verfahren zur Berechnung der Modellparameter bestimmt die optimalen Warping-Pfade aller Muster in einem Cluster bzgl. eines Initialmodells. Anhand dieser Warping-Pfade lassen sich für einen Zeitpunkt j im Referenzmuster korrespondierende Merkmalsvektoren in der Menge $P_C^{\mathcal{M}}(j)$ zusammenfassen (siehe Gleichung 5.7). Diese Mengen wurden auf ihre statistischen Eigenschaften hin untersucht.

Normierte Ortskoordinaten $(\tilde{x}, \tilde{y})$

Abbildung 6.1 zeigt die Verteilung der Merkmale $(\tilde{x}, \tilde{y})$. Aufgetragen sind korrespondierende Merkmalsvektoren zu 16 Zeitpunkten im Referenzmuster. Unten links befindet sich das Schaubild zum Zeitpunkt $j = 1$, oben rechts entsprechend zum Zeitpunkt $j = N_{\mathcal{M}}$. Wobei die Schaubilder zeilenweise von rechts nach links aufgetragen wurden. Man beachte, daß keine einheitliche Skalierung gewählt wurde.

Um eine realistische Aussage über die tatsächliche Verteilung machen zu können sind sicherlich viel mehr Trainingsbeispiele notwendig. Dennoch läßt sich erkennen, daß die hier dargestellten Punktwolken gut durch eine Gaußsche Normalverteilung approximiert werden können. Auch in anderen Modellklassen konnte nicht festgestellt werden, daß etwa

zwei Punktwolken existieren, die zum Beispiel durch eine Gaußsche Mischverteilung hätten modelliert werden müssen.

Die unterschiedlichen Ausmaße und Orientierungen der Punktwolken rechtfertigen, für jeden Zeitpunkt j eine eigene Kovarianzmatrix anzunehmen. Somit ist eine Verbesserung der Klassifikationergebnisse gegenüber der Verwendung der euklidischen Distanz als lokale Distanzfunktion zu erwarten (vergleiche Abschnitt 4.3.3).

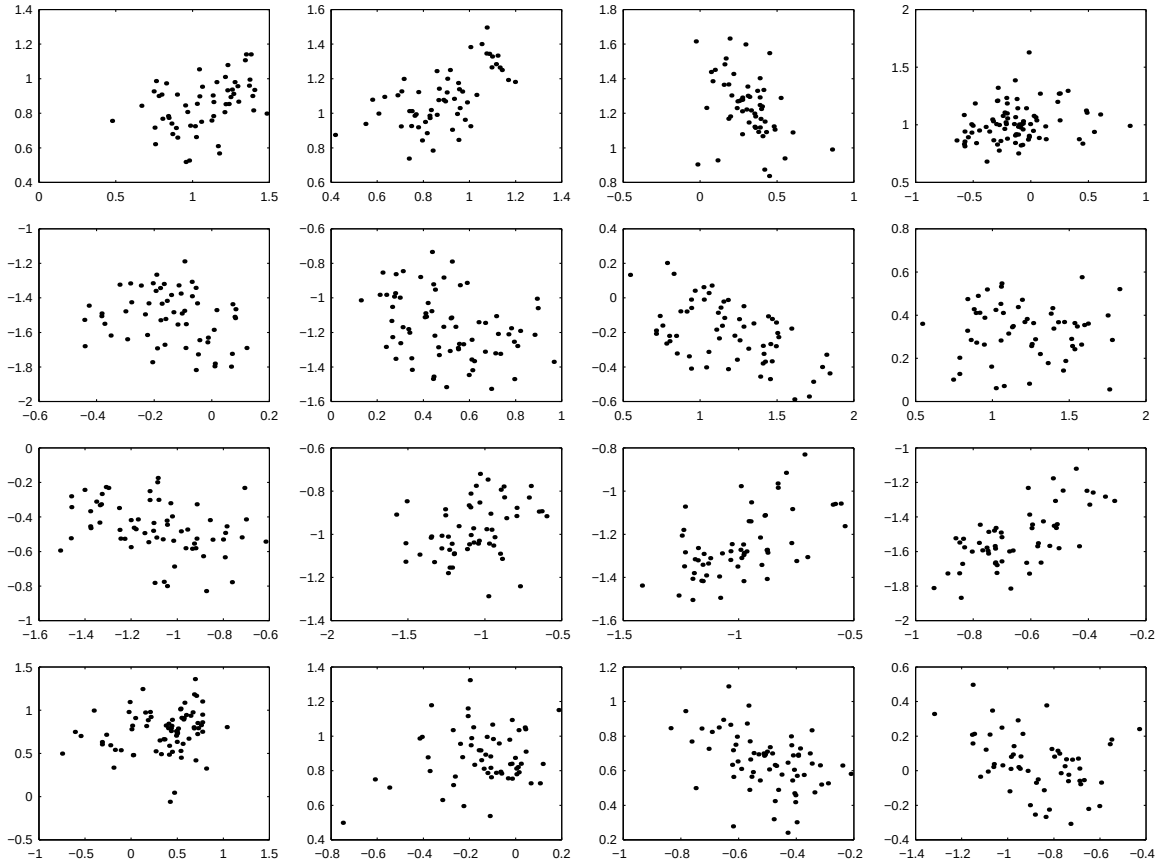


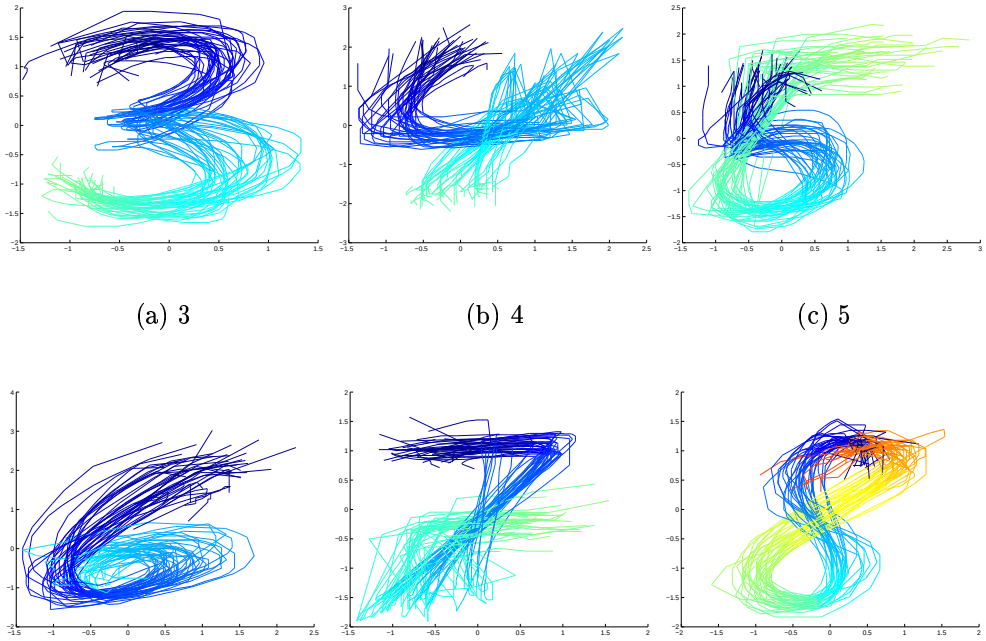
Abbildung 6.1: Je ein Schaubild repräsentiert einen Zeitpunkt im Referenzmuster „0“. Das Schaubild links unten zeigt die Verteilung der Merkmale zum Zeitpunkt $j = 1$. Entlang der x-Achse ist das Merkmal $\tilde{x}$ und entlang der y-Achse ist $\tilde{y}$ aufgetragen. Als Clusterrepräsentant für das Initialmodell wurde das Medianzentrum gewählt.

In Abbildung 6.2 wird für ausgewählte Klassen die Verteilung der Buchstaben bzw. Ziffern im Musterraum dargestellt. In einem Schaubild sind jeweils alle Muster aus einem Cluster gemäß ihrer Merkmale aufgetragen: $\tilde{x}$ auf der x-Achse und $\tilde{y}$ auf der y-Achse. Alle Muster wurden an das initiale Modell angeglichen. Als Clusterrepräsentant wurde das Medianzentrum gewählt. Die bezüglich des Initialmodells korrespondierenden Punkte der Muster sind mit der gleichen Farbe kodiert. Die Farbkodierung wurde so gewählt, daß die Punkte zum Anfang des Schriftzuges blau dargestellt werden. Da die Pen-Up-Bewegungen ignoriert werden, sind in den Schaubildern jeweils das Ende eines Striches und der Anfang des darauffolgenden Striches verbunden.

Bei einigen Ziffern sind deutlich Regionen mit unterschiedlich hohen Varianzen zu erkennen. Betrachtet man etwa den unteren „Bauch“ der Ziffer 3, erkennt man eine große

Varianz in x-Richtung. Wohingegen am unteren Scheitelpunkt eine geringere Varianz in y-Richtung vorhanden ist. Besonders deutlich läßt sich dies auch bei der Ziffer 7 beobachten. Während der erste horizontale Strich eine kleine Streuung aufweist, variiert der vertikale Strich sehr stark in x-Richtung. Das liegt hauptsächlich daran, daß hier zwei verschiedene Schreibweisen der Ziffer 7 zusammengefaßt wurden: eine kursive Schreibweise und eine, bei der der vertikale Strich tatsächlich parallel zur y-Achse ist.

Diese zeitlich variierenden Verteilungen verdeutlichen nochmals die Notwendigkeit einer Modellierung der Merkmalsverteilung in Abhängigkeit des Zeitpunktes im Referenzmuster.



(a) 3 (b) 4 (c) 5

(d) 6 (e) 7 (f) 8

Abbildung 6.2: Verteilung der Ziffern 3–8 im Musterraum. Das Merkmal $\tilde{x}$ ist auf der x-Achse aufgetragen und entsprechend $\tilde{y}$ auf der y-Achse. Die bezüglich des Initialmodells korrespondierenden Punkte der Muster sind mit der gleichen Farbe kodiert.

Entfernung zur Grundlinie und Tangentenwinkel $(\bar{y}, \theta)$

In Abbildung 6.3 ist analog zu Abb. 6.1 die normierte, vertikale Entfernung zur Grundlinie $\bar{y}$ auf der x-Achse und der Tangentenwinkel θ auf der y-Achse aufgetragen. Da der Wertebereich für θ zyklisch ist, wurden jeweils die Werte der Winkel so verschoben, daß der Mittelwert auf π liegt. Da an dieser Stelle nur das zweite Moment (Varianz) untersucht werden soll, entsteht dadurch kein wesentlicher Informationsverlust.

Die Punktwolken sind im Vergleich zu $(\tilde{x}, \tilde{y})$ teilweise etwas „unförmig“. Negativ fallen die Verteilungen im Schaubild in der dritten Spalte und dritten Zeile bzw. in der vierten Spalte und dritten Zeile auf. Einige Punkte liegen exakt auf der θ-Achse, d.h. für diese Punkte gilt $\bar{y} = 0$. Dies läßt sich dadurch erklären, daß die Referenzlinien manuell gesetzt wurden. Die diesem Schaubild zugrundeliegenden Daten bestehen aus einzeln segmentierten

Ziffern. Auf diesen kann der in Abschnitt 3.1.3 vorgestellte Algorithmus zur Bestimmung der Referenzlinien nicht angewendet werden. Daher wurde die Grundlinie für einen Buchstaben bzw. eine Ziffer auf den minimalen y-Wert desselben gesetzt. Die Kernlinie wurde auf halbe Höhe gesetzt: $y_{\text{Kernlinie}} = y_{\min} + (y_{\max} - y_{\min})/2$.

Für viele der Verteilungen kann dennoch festgehalten werden, daß diese Verteilungen sich durch eine einfache Gaußverteilung ausreichend genau approximieren lassen.

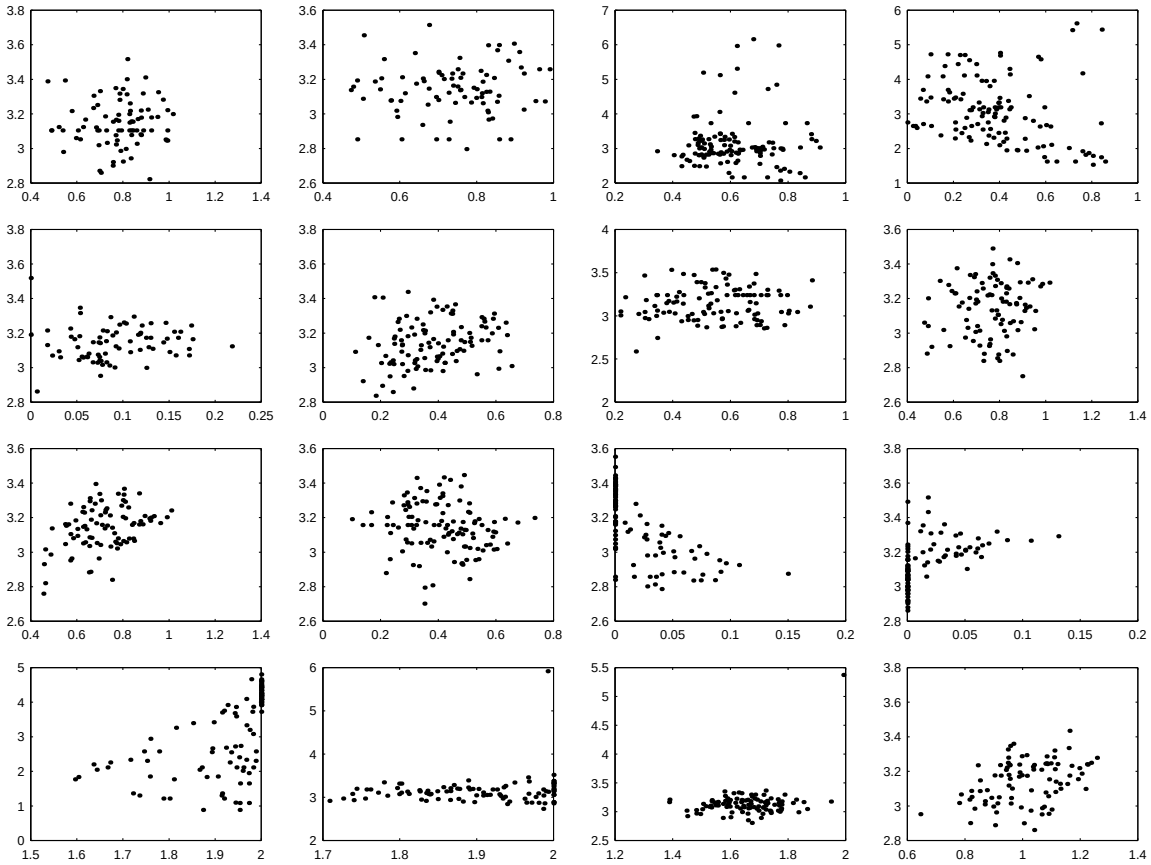
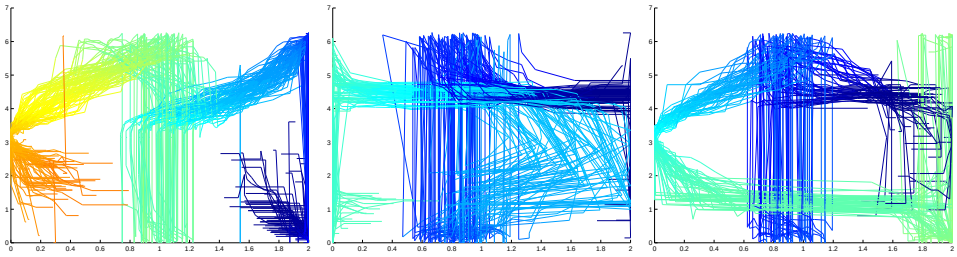


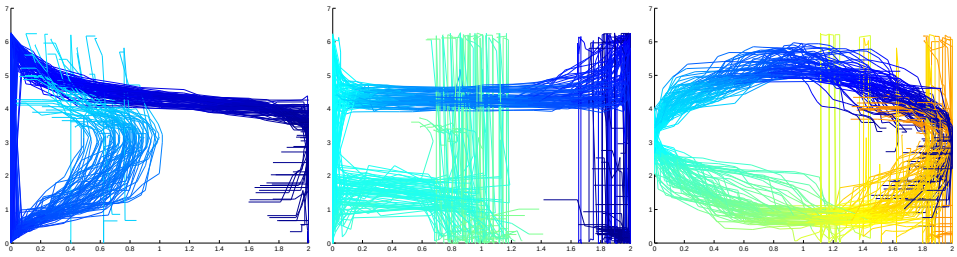
Abbildung 6.3: Je ein Schaubild repräsentiert einen Zeitpunkt im Referenzmuster „6“. Das Schaubild links unten zeigt die Verteilung der Merkmale zum Zeitpunkt $j = 1$. Entlang der x-Achse ist der normierte Abstand zur Grundlinie $\bar{y}$ und entlang der y-Achse ist der Tangentenwinkel θ aufgetragen. Als Clusterrepräsentant für das Initialmodell wurde das Medianzentrum gewählt.

In Abbildung 6.4 wird analog zu Abb. 6.2 die Verteilung der Ziffern im Musterraum dargestellt. Auf der x-Achse ist der normierte Abstand zur Grundlinie $\bar{y}$ und auf der y-Achse der Tangentenwinkel θ aufgetragen. Die vertikalen Linien zwischen 0 und 2π sind auf den zyklischen Wertebereich von θ zurückzuführen.

Für diese Kombination von Merkmalen lassen sich ebenfalls Regionen unterschiedlicher Varianz in den einzelnen Musterklassen identifizieren. Dies kann besonders im Fall der Ziffer 6 beobachtet werden. Am Anfang des Schriftzuges weisen die Merkmale eine geringe Streuung auf. Wohingegen die Merkmale im letzten Drittel stark variieren.



(a) „3“ (b) „4“ (c) „5“



(d) „6“ (e) „7“ (f) „8“

Abbildung 6.4: Verteilung der Ziffern 3–8 im Musterraum. Das Merkmal $\bar{y}$ ist auf der x-Achse aufgetragen und entsprechend θ auf der y-Achse. Die bezüglich des Initialmodells korrespondierenden Punkte der Muster sind mit der gleichen Farbe kodiert. Die Farbe „blau“ kennzeichnet den Anfang des Schriftzuges, „rot“ das Ende.

6.2 Benchmarking

Benchmarking bezeichnet den Leistungsvergleich zwischen konkurrierenden Systemen durch eine sorgfältige Auswahl, Messung und Vergleich von kritischen Aspekten des zu lösenden Problemes [14]. Leider existieren zur Zeit keine allgemein zugänglichen Benchmarks, so daß ein Leistungsvergleich mit existierenden Buchstabenerkennern nicht möglich ist. Im folgenden sollen daher durch Experimente die einzelnen Varianten der in den vorangegangenen Kapiteln vorgestellten Algorithmen untersucht werden.

6.2.1 Bewertung eines Buchstabenerkenners

Wahl der Test- und Trainingsmenge

Die dieser Arbeit zugrundeliegenden Daten bestehen zum einen aus einzeln geschriebenen Buchstaben und zum anderen aus von Wörtern segmentierten Buchstaben (vergleiche Abschnitt 2.4). Die ersten vier Benchmarks in Tabelle 6.1 enthalten einzeln geschriebene Buchstaben (spaced) bzw. Ziffern. Die Buchstaben aus M_01 wurden manuell aus Wörtern segmentiert. Ein Benchmark besteht aus zwei disjunkten Mengen. Aus der Trainingsmenge werden die Modellparameter geschätzt. Das resultierende Modell wird anschließend anhand der Testmenge evaluiert.

Name	Schreib-stil	Alphabet	Anz. der Schreiber	Anz. der Beispiele	
				Training	Test
N_01	spaced	0–9	10	800	940
B_01	spaced	A–Z	10	2080	2206
C_01	spaced	a–z	10	2080	2353
X_01	spaced	0–9,a–z,A–Z	10	-	10459
M_01	mixed	$\subset$ a–z	7	1008	4332

Tabelle 6.1: In dieser Arbeit verwendete Benchmarks

Bei der Zusammenstellung der Trainings- und Testmengen wurde wie folgt vorgegangen. Von jedem Schreiber wurden jeweils acht Beispiele eines jeden Zeichens in die Trainingsmenge aufgenommen. Die restlichen Zeichen wurden der Testmenge zugeordnet. Der Benchmark X_01 bildet eine Ausnahme. Dieser besteht nur aus der Testmenge. Der Modellsatz für diesen Benchmark wird aus den auf N_01, B_01 und C_01 trainierten Modellsätzen zusammengefügt.

Für M_01 mußte das Alphabet auf eine Teilmenge der Kleinbuchstaben a–z beschränkt werden, da in dem zugrundeliegenden Text einige Buchstaben gar nicht oder nur selten aufgetreten sind. Insgesamt enthält dieser Benchmark 18 verschiedene Buchstaben.

Bewertungsmethoden

Von einem Buchstabenerkenner wird erwartet, daß er möglichst alle Eingaben korrekt klassifiziert. Daher ist der relative Anteil der richtig erkannten Eingaben ein natürliches Maß für die Güte des Klassifikationssystems. Die *Klassifikationsrate* berechnet sich aus

dem Quotienten der Anzahl der korrekt erkannten Muster und der Anzahl der Testmuster insgesamt, und wird in Prozent angegeben:

$$C_{\text{cor}} = \frac{\#\text{Testmuster} - \#\text{Fehler}}{\#\text{Testmuster}} \times 100 \tag{6.1}$$

In einigen Anwendungen wird vom Klassifikationssystem eine Alternative zu dem wahrscheinlichsten Lösungskandidaten erwartet. So kann meist unter der Verwendung von Kontextwissen die Klassifikationsrate verbessert werden. Aus diesem Grund wird untersucht, ob das richtige Zeichen unter den k besten Treffern liegt.

$$B^k = \frac{\#k\text{-Besten}}{\#\text{Testmuster}} \times 100 \tag{6.2}$$

wobei $\#k$-Besten die Anzahl der Testmuster ist, bei denen sich die korrekte Klasse unter den ersten k Kandidaten befindet.

In einigen Systemen ist die Möglichkeit einer Zurückweisung vorgesehen. Wenn eine eindeutige Klassifikation nicht möglich ist, wird das Testmuster zurückgewiesen und muß gegebenenfalls manuell klassifiziert werden. Eine häufige Vorgegehensweise dabei ist, ein Testmuster zurückzuweisen, wenn die Distanz des besten Kandidaten einen gewissen Zurückweisungsschwellwert β überschritten hat.

Der Prozentsatz der Muster, die zurückgewiesen werden, wird als *Zurückweisungsrate* (engl.: reject rate) R bezeichnet. Der Prozentsatz der Muster, die nicht zurückgewiesen, aber falsch klassifiziert werden, heißt *Fehlerrate* E [14].

$$E = \frac{\#\text{Fehler}}{\#\text{Testmuster} - \#\text{Zurückweisungen}} \times 100 \tag{6.3}$$

Dies entspricht der bisherigen Definition der Fehlerrate, wenn man bedenkt, daß keine Muster zurückgewiesen werden.

Chow schlägt vor, zur Charakterisierung der Leistungsfähigkeit eines Klassifikationssystems die Fehlerrate gegen die Zurückweisungsrate aufzutragen. Diese Kurve heißt *Fehler/Zurückweisungs-Kurve* (engl.: error-reject curve) [14].

6.3 Modellberechnung: Experimente

Die ursprüngliche Formulierung des DTW-Abstandes in Kapitel 4 basiert auf dem Quadrat der euklidischen Distanz als lokales Distanzmaß. In Abschnitt 4.3.1 wurde eine lokale Distanz hergeleitet, die die statischen Verteilungen der Muster berücksichtigt. In diesem Abschnitt soll nun überprüft werden, inwieweit sich diese gewichtete lokale Distanz auf die Klassifikationsergebnisse gegenüber der ungewichteten Distanz (Quadrat der euklidischen Metrik) auswirkt.

6.3.1 Gewichtete lokale Distanz

Bei der Schätzung der Modellparameter aus einem Cluster wird im Initialisierungschritt ein Repräsentant aus dem Cluster als Referenzmuster gewählt. Neben dem Einfluß der

Gewichtung soll nun untersucht werden, inwieweit die Wahl des Clusterrepräsentanten die Klassifikationsrate beeinflußt.

Untersucht wurden das Mittelwertzentrum und das Medianzentrum. Jedem Benchmark wurde dieselbe Partitionierung zugrunde gelegt. Anschließend wurde der jeweilige Repräsentant bestimmt und anhand dessen in einem Iterationsschritt die Modellparameter geschätzt. Für den ungewichteten Fall wurden die Clusterrepräsentanten als Referenzmuster verwendet. Die Tabelle 6.2 zeigt die Klassifikationsraten, die mit diesen Modellsätzen erzeugt wurden.

		Mittelwert-zentrum	Median-zentrum
N_01	ungewichtet	85.764	97.571
	gewichtet	99.485	99.066
B_01	ungewichtet	47.724	88.101
	gewichtet	93.684	95.570
C_01	ungewichtet	32.604	84.619
	gewichtet	84.412	92.893

Tabelle 6.2: Klassifikationsrate C_{cor} für Ziffern (N_01), Großbuchstaben (B_01) und Kleinbuchstaben (C_01). Clustermindestgröße: 4, Clustering-Methode: Average Linkage, Merkmale: $(\tilde{x}, \tilde{y})$. Die Berechnung der Gewichte wurde nach einem Iterationsschritt abgebrochen.

Tabelle 6.2 zeigt, daß im Fall des Mittelwertzentrums die Fehlerrate ($100\% - C_{\text{cor}}$) durch die Einführung der Gewichte um den Faktor 4–9 verbessert werden konnte. Beim Medianzentrum konnte immerhin noch eine Verbesserung die Fehlerrate um den Faktor 2–2,5 erreicht werden. Aus der Tabelle 6.2 können drei Schlüsse gezogen werden. Erstens ist das Medianzentrum besser als Clusterrepräsentant geeignet als das Mittelwertzentrum. Dies entspricht auch der Empfehlung von Theodoris *et al* [21], das Medianzentrum dann zu verwenden, wenn es sich bei dem zugrundeliegenden Abstandsmaß um keine Metrik handelt, wie dies bei dem DTW-Abstand ja der Fall ist.

Zweitens, der Grund dafür, daß die relative Verbesserung der Fehlerraten beim Mittelwertzentrum größer ausfallen, als beim Medianzentrum, liegt darin, daß beim Schätzen der Modellparameter auch ein mittleres Referenzmuster berechnet wird. Im ungewichteten Fall stellt der Clusterrepräsentant tatsächlich auch das Referenzmuster dar. Im gewichteten Fall hingegen, wird der Clusterrepräsentant dazu benötigt, die Korrespondenzen herzustellen, aus denen dann das Referenzmuster berechnet wird. Dennoch wird ersichtlich, daß ein gutes Initialmodell die Qualität der Modelle erheblich verbessern kann.

Die wichtigste Schlußfolgerung ist sicherlich, daß das Ausnutzen der statistischen Gegebenheiten der verschiedenen Modellklassen zu einer Verbesserung der Klassifikationsrate geführt hat. Diese soll nun im nächsten Abschnitt weiter untersucht werden.

6.3.2 Anzahl der Modelle

In diesem Abschnitt soll untersucht werden, inwieweit sich die Anzahl der Referenzmodelle auf die Klassifikationsrate auswirkt. Dazu wurde der Schwellwert c, ab dem das Cluste-

ring abgebrochen wird, solange verringert, bis keine geeignete Partitionierung mehr erzeugt wurde. Auf jeder dieser Partitionierungen wurden anschließend die Modelle berechnet und anhand der Testmenge evaluiert. Gleichzeitig wurde für jede dieser Konfigurationen auch bzgl. dem Quadrat der euklidischen Distanz klassifiziert, wobei wieder die Clusterrepräsentanten als Referenzmuster dienten. Die ermittelten Klassifkationsraten für die Benchmarks N_01, B_01 und C_01 werden in den Tabellen 6.3, 6.4 und 6.5 gegenübergestellt.

Auffallend ist, daß die Anzahl der Modelle die Klassifikationsrate im gewichteten Fall kaum beeinflußt. Zwar ist tendenziell eine Verbesserung der Klassifikationsrate feststellbar, wenn man die Anzahl der Modelle erhöht, aber eine größere Anzahl an Modellen impliziert nicht unbedingt eine Verbesserung der Klassifikationsrate. Dies gilt nicht für das ungewichtete Verfahren. Im ungewichteten Fall steigt die Klassifikationsrate mit der Anzahl der Referenzmuster.

Schwellwert c	0.1	0.15	0.2	0.25	0.3	0.4
Anz. Modelle	58	44	31	21	16	14
C_{cor} (ungewichtet)	99.064	99.378	97.571	97.582	95.493	93.843
C_{cor} (gewichtet)	98.908	98.957	99.066	99.039	98.411	98.510

Tabelle 6.3: Benchmark: N_01, Clustermindestgröße: 4, Anzahl der Iterationen: 1, Clustering-Methode: Average-Linkage, Clusterrepräsentant: Medianzentrum.

Zwar kann mit der ungewichteten Methode jeweils dieselbe Klassifikationsrate erreicht werden, wie mit dem gewichteten Verfahren. Allerdings werden dann zwei bis drei mal soviele Modelle benötigt. Was einer Beschleunigung des gewichteten Verfahren gegenüber dem ungewichteten um den gleichen Faktor bedeutet, wenn man die etwas aufwendigere Berechnung der lokalen Distanz außer acht läßt.

Mit dem gewichteten Verfahren konnten die besten Ergebnisse teilweise mit relativ kleinen Modellsätzen erreicht werden. Dies läßt auf gute Generalisierungsfähigkeiten der gewichteten Modelle schließen.

Schwellwert c	0.15	0.2	0.25	0.4	0.5	0.6	0.7
Anz. Modelle	126	99	74	52	45	42	40
C_{cor} (ungewichtet)	97.142	95.815	93.608	89.454	88.101	87.420	86.553
C_{cor} (gewichtet)	96.688	97.116	95.481	95.882	95.570	95.225	94.905

Tabelle 6.4: Benchmark: B_01, Clustermindestgröße: 4, Anzahl der Iterationen: 1, Clustering-Methode: Average-Linkage, Clusterrepräsentant: Medianzentrum.

Je größer die Modellsätze gewählt werden, umso kleiner werden die entsprechenden Cluster. Dadurch werden die statistischen Größen weniger genau geschätzt. Dies wirkt sich auf die Klassifkationsergebnisse aus. Für die ungewichtete Methode gilt dies nicht, denn es wird lediglich ein Repräsentant für jedes Cluster benötigt. Wenn nur sehr wenig Daten vorliegen, ist das sicherlich von Vorteil.

Schwellwert c	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Anz. Modelle	121	86	64	57	51	46	42	40
C_{cor} (ungewichtet)	90.734	88.851	85.416	86.494	84.738	84.619	83.259	80.603
C_{cor} (gewichtet)	91.299	92.678	91.486	92.493	92.786	92.893	92.032	90.615

Tabelle 6.5: Benchmark: C_01, Clustermindestgröße: 4, Anzahl der Iterationen: 1, Clustering-Methode: Average-Linkage, Clusterrepräsentant: Medianzentrum.

6.3.3 Untersuchung der Clustering-Methoden

In Abschnitt 5.1.2 wurde anhand von zwei Beispielen illustriert, wie sich die Wahl der Clustering-Methoden auf die resultierende Partitionierung auswirkt. Im folgenden sollen nun die drei Clustering-Methoden Single-, Complete- und Average-Linkage dahingehend untersucht werden, welcher der drei Algorithmen sich für die Identifizierung der verschiedenen Modellklassen am besten eignet

Bewertung nach Klassifikationsresultat

Die Güte einer Partitionierung mißt sich an dem Klassifikationsergebnis, welches sich aus dem daraus berechneten Modellsatz ergibt. Die Idee ist also, für jede Clustering-Methode eine Partitionierung zu berechnen und daraus den Modellsatz zu bestimmen. Dieser wird anhand der Trainingsmenge evaluiert.

Da die Parameter beim Clustering manuell eingestellt werden müssen, ist keine objektive Beurteilung möglich. Daher wird wie folgt vorgegangen: der Schwellwert, ab dem das Clustering-Verfahren abbricht, wird intuitiv so eingestellt, daß er zu einer guten Partitionierung führt. Dabei wurde darauf geachtet, daß nicht zuviele (max. 15 pro Zeichen) Elemente aus der Modellberechnung entfernt wurden. Ferner durften nicht zuviele Modelle entstehen. In die Modellberechnung sind jeweils nur Cluster mit mindestens vier Elementen eingeflossen. Als Clusterrepräsentant wurde das Medianzentrum gewählt.

Zunnächst wurde versucht, für jede Clustering-Methode den Schwellwert so einzustellen, daß in etwa die gleiche Anzahl an Referenzmodellen berechnet wurde. Tabelle 6.6 zeigt für den Benchmark B_01 die resultierenden Klassifkationsraten. Die Ergebnisse sind praktisch identisch. Die unterschiedlichen Partitionierung der drei Clustering-Verfahren haben sich demnach nicht meßbar auf das Gesamtergebnis der Klassifikation ausgewirkt.

	Schwellwert	Anz. Modelle	C_{cor}
Single-Linkage	0.15	51	95.72
Complete-Linkage	1.0	51	95.35
Average-Linkage	0.5	45	95.28

Tabelle 6.6: Es wurde versucht, für jede Clustering-Methode etwa gleich große Modellsätze zu erzeugen. Die Klassifikationsraten für die Modellsätze werden präsentiert. Benchmark: B_01.

Beim zweiten Experiment wurde jeweils der Schwellwert solange runtergesetzt, bis nach subjektiven Bemessen noch eine gute Partitionierung erhalten wurde. Tabelle 6.7 zeigt das

Ergebnis für dieses Experiment. Mit dem Single-Linkage Algorithmus wurde die geringste Anzahl an Modellen bestimmt. Die Modellsätze, die mit dem Average-Algorithmus erzeugt wurden, sind wesentlich kleiner, erreichen aber die gleichen Klassifikationsraten, wie die Modellsätze, die mit dem Complete-Linkage Verfahren berechnet wurden.

	Schwellwert	Anz. Modelle	C_{cor}
Single-Linkage	0.3	36	91.20
Complete-Linkage	1.4	59	93.01
Average-Linkage	0.8	42	93.02

Tabelle 6.7: Für jede Clustering-Methode, wurde der Schwellwert intuitiv so eingestellt, daß er zu einer „guten“ Partitionierung führt. Die Klassifikationsraten für die resultierenden Modellsätze werden präsentiert. Benchmark: C_01.

Bewertung nach Einstellbarkeit des Parameters

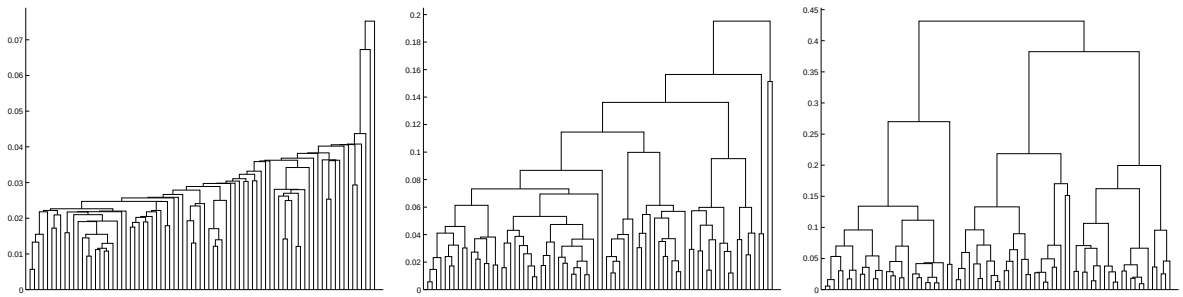
Jedes Clustering-Verfahren gruppiert die Testmuster unterschiedlich. Die hier präsentierten Ergebnisse lassen allerdings nicht darauf schließen, daß eines der Verfahren in dem Sinne besser partitioniert, daß sich bessere Modelle aus den berechneten Clustern schätzen lassen. Zwar ist die Klassifikationsrate für das Single-Linkage Modell am niedrigsten (siehe Tabelle 6.7), doch läßt sich das eher auf die geringere Größe des Modellsatzes zurückführen als auf die Qualität der Partitionierung.

Eine weitere Verkleinerung des Schwellwertes führte beim Single-Linkage Algorithmus nicht zu einer Erhöhung der Anzahl an Modellen, da die neuen Cluster kleiner waren als die vorgeschriebene Mindestgröße, und somit nicht in die Modellberechnung eingeflossen sind. Bei den beiden anderen Clustering-Methoden hingegen konnte der Modellsatz noch weiter verfeinert werden, ohne daß zuviele Cluster verworfen wurden. Dieser Sachverhalt ist deutlich erkennbar, vergleicht man die Dendrogramme der drei Verfahren. Abbildung 6.5 zeigt für jedes Verfahren das resultierende Dendrogramm für die Ziffer 3. Verkleinert man den Schwellwert sukzessive etwa beim Single-Linkage Verfahren, so erhält man ein großes Cluster, von dem sehr kleine Cluster abgetrennt werden. Wohingegen mit den beiden Verfahren mehrere größere Cluster erzeugen lassen. Während der Complete-Linkage Algorithmus kompakte Cluster erzwingt, ist mit dem Average-Linkage Verfahren möglich Ausreißer zu identifizieren und somit aus der Modellberechnung zu entfernen.

Identifikation unterschiedlicher Schreibweisen

Ein Resultat aus Abschnitt 5.1.2 ist, daß der Single-Linkage Algorithmus dazu neigt getrennte Anhäufungen von Punkte zu verschmelzen, wenn diese durch eine Kette verbunden sind. Dieses Verhalten konnte am Beispiel der Ziffer „0“ beobachtet werden.

Die Ziffer 0 existiert in zwei Schreibweisen im N_01 Benchmark (vergleiche linkes und rechtes Schaubild in Abbildung 6.6). Mit Complete-Linkage und Average-Linkage konnten diese Schreibweisen eindeutig identifiziert werden. Versuche, diese beiden Variationen mit dem Single-Linkage Algorithmus zu trennen, führten zu einer Überpartitionierung.



(a) Single-Linkage (b) Average-Linkage (c) Complete-Linkage

Abbildung 6.5: Dendrogramm für die Ziffer „3“. Als Merkmale wurden die normierten Ortskoordinaten $(\tilde{x}, \tilde{y})$ verwendet. Bei einer Verringerung des Schwellwerts in (a) ensteht ein großes Cluster mit vielen kleinen Clustern, während in (b) etwa gleich große Cluster entstehen.

Dies läßt sich dadurch erklären, daß in der Trainingsmenge eine Sequenz von Testmustern existiert, die diese beiden Schreibweisen ineinander überführt (siehe Abbildung 6.6).

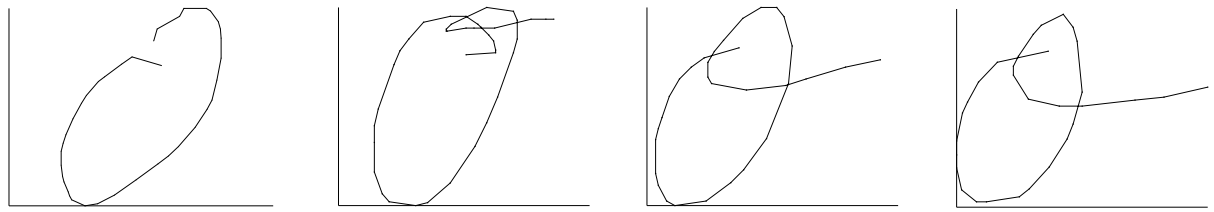


Abbildung 6.6: Variationen der Ziffer 0 aus der Trainingsmenge von N_01. Sequenzen, die unterschiedliche Schreibweisen in einander überführen, lassen sich mit Single-Linkage Algorithmus nur schwer trennen.

Es kann festgehalten werden, daß der Single-Linkage Algorithmus für diese Aufgabenstellung ungeeignet ist. Bewährt haben sich der Complete- und Average-Linkage Algorithmus. Wobei ersterer nur dann verwendet werden sollte, wenn sichergestellt ist, daß alle Trainingsbeispiele in die Modellberechnung aufgenommen werden können, etwa durch eine manuelle Überprüfung. Da davon im folgenden nicht ausgegangen werden kann, werden daher alle Modellberechnungen mit Average-Linkage durchgeführt.

6.3.4 Verbesserung durch Iteration

Bei der Formulierung der iterativen Berechnung der Modellparameter (vgl. Abschnitt 5.2) wurde offengelassen, wann die Iteration abgebrochen werden soll. Zudem ist nicht klar, ob das Verfahren überhaupt konvergiert.

Bei Experimenten wurde festgestellt, daß eine Erhöhung der Anzahl der Iterationen nicht zu einer Verbesserung der Klassifikationsrate geführt hat. Tatsächlich wurde sie sogar geringfügig schlechter. Wurden die Modellparameter in nur einem Iterationsschritt geschätzt, wurde eine Klassifikationsrate von $92,03\%$ erreicht. Nach zwei Iterationen lag

sie hingegen bei $91,56\%$. In den darauffolgenden Iterationsschritten konnte eine leichte, stetige Verbesserung der Klassifikationsrate festgestellt werden. Dennoch lag sie nach 30 Iterationen weiterhin bei $91,79\%$.

Da durch eine Erhöhung der Anzahl der Iterationen keine Verbesserung der Klassifikationsrate erreicht werden konnte, soll nun untersucht werden, ob sich das Verhalten des Klassifikators bzgl. einer Zurückweisung verbessert. Dazu wurde basierend auf dem C_01 Benchmark eine Partitionierung mittels des Average-Linkage Algorithmus berechnet. Als initiales Referenzmuster wurde jeweils das Medianzentrum gewählt. Die Clustermindestgröße betrug 4. Ausgehend von dieser Konfiguration wurden die Modelle berechnet. Zunächst wurden die Modellparameter in verschiedenen Anzahlen an Iterationschritten geschätzt.

Die Fehler- und die Zurückweisungsrate wurden für die Zurückweisungsschwellwerte $\beta = -1, -0.9, \ldots, 0.9, 1, 2, 3, 4$ ausgewertet. Das Ergebnis ist in Abbilung 6.7 aufgetragen. Man stellt fest, daß in dem für die Anwendung interessanten Bereich von $R < 10\%$, die Fehlerrate bei einmaliger Iteration geringfügig besser ist. Nach 30 Iterationen ist die Fehlerrate bei einer Zurückweisungsrate von fast 70%, ca. 1.5-mal niedriger als nach einer Iteration. Dieses Verhalten deckt sich mit der Beobachtung, daß Testmuster, die bereits nach einer Iteration gut erkannt wurden, nach mehreren Iterationen in dem Sinne besser erkannt wurden, als daß die Distanz zum korrekten Modell noch kleiner wurde, während die Distanz zum nächstbesten Modell größer wurde.

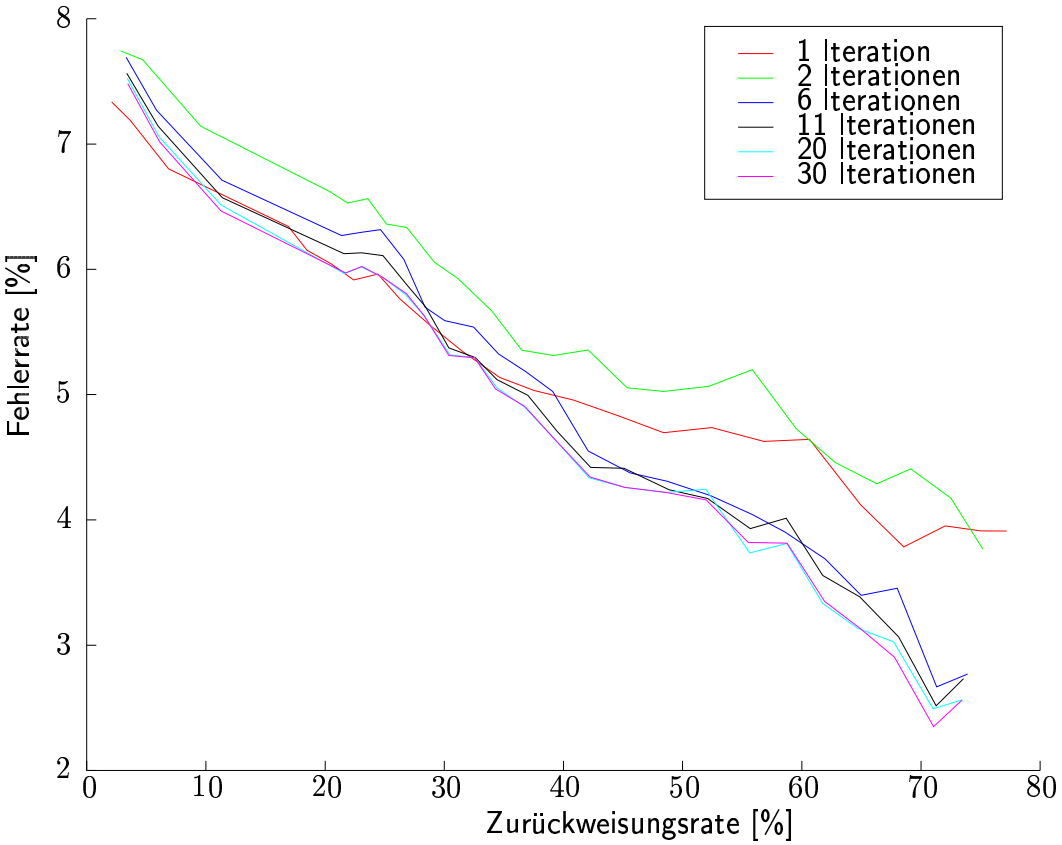


Abbildung 6.7: Fehler/Zurückweisungs-Kurve

Allerdings bleibt festzuhalten, daß keine Verbesserung der Klassifikationsrate festge-

stellt werden konnte und daher wird im folgenden die Berechnung der Modellparameter nach einem Iterationsschritt abgebrochen.

6.4 Anmerkungen zur Implementation und Laufzeit

Die Arbeit wurde größtenteils in der Matlab-Programmierumgebung implementiert. Die Interpretersprache Matlab hat den Vorteil, daß aufgrund der umfangreichen Bibliothek von Funktionen sich die entwickelten Algorithmen sehr schnell implementieren lassen. Allerdings sind die so erhaltenen Programme weniger effizient, als in maschinennahen Sprache wie C. Vor allem Schleifen sind in Matlab diesbezüglich ein Problem. Daher wurde der DTW-Alogrithmus in C implementiert und in die Matlab Umgebung eingebunden. Weiterhin wurde der DTW-Algorithmus flexibel programmiert, um z.B. unterschiedliche lokale Stetigkeitsbedingungen modellieren zu können und ist somit nicht für kurze Laufzeiten optimiert.

Es folgt nun eine kurze Untersuchung der Laufzeiten. Die Messungen wurden auf einer SGI 02 mit einem R1000 Prozessor durchgeführt. Die durchschnittliche Laufzeit für einen Aufruf des DTW-Algorithmus betrug $\sim 0,1$ Sekunden. Dh. bei einer Modellsatzgröße von 40 muß der Anwender etwa vier Sekunden auf das Klassifikationsergebnis warten. Dies ist für den praktischen Einsatz zu lange. Messungen der CPU-Zeit, in der sich das Programm im tatsächlichen DTW-Algorithmus befindet, haben ergeben, daß für den Vergleich zweier Muster etwa 1–2 hundertstel Sekunden benötigt werden. Dies läßt auf einen großen Overhead in der Kommunikation zwischen Matlab und der C Funktion schließen. Ein reine Implementierung in C würde daher sicherlich eine erhebliche Beschleunigung einbringen. Weiterhin ist der Algorithmus noch sehr allgemein gehalten, so daß genug Raum für Optimierungen vorhanden ist.

6.5 Klassifikationsergebnisse

In diesem Abschnitt werden die Klassifkationsergebnisse für vier verschiedene Kombinationen von Merkmalen verglichen. Es werden hier ausschließlich die Gesamtklassifikationsraten gegenübergestellt. Eine detailliertere Interpretation der Ergebnisse befindet sich im Anhang.

Es wurden folgende Merkmalsvektoren evaluiert:

$(\tilde{x}, \tilde{y})$	:	Normierte Ortskoordinaten
$(\tilde{x}, \tilde{y}, \theta)$	:	Normierte Ortskoordinaten und Tangentenwinkel
$(\bar{y}, \theta)$	:	Abstand zur Grundlinie und Tangentenwinkel[1]
$(\bar{y}, \theta, \kappa)$	:	Abstand zur Grundlinie, Tangentenwinkel und Krümmung

In den folgenden Tabellen 6.8 bis 6.12 sind jeweils die Klassifkationsresultate für einen Benchmark zusammengefaßt. Man beachte, daß die Größe der Modellsätze für verschiedene Merkmalsvektoren stark variieren kann. Dies muß beim Vergleich der Klassifikationsraten berücksichtigt werden.

[1] Die Merkmale $(\bar{y}, \theta)$ entsprechen den von Tappert in [20] verwendeten.

Merkmale	C_{cor}	B^2	B^3	#Mod.
$(\tilde{x}, \tilde{y})$	99.07	99.79	99.90	31
$(\tilde{x}, \tilde{y}, \theta)$	98.52	99.77	99.90	24
$(\bar{y}, \theta)$	99.44	99.90	100.00	24
$(\bar{y}, \theta, \kappa)$	98.94	99.69	99.79	26

Tabelle 6.8: N_01

Merkmale	C_{cor}	B^2	B^3	#Mod.
$(\tilde{x}, \tilde{y})$	97.12	99.35	99.68	99
$(\tilde{x}, \tilde{y}, \theta)$	96.93	99.15	99.46	59
$(\bar{y}, \theta)$	96.67	99.08	99.54	55
$(\bar{y}, \theta, \kappa)$	95.19	98.61	99.10	46

Tabelle 6.9: B_01

Merkmale	C_{cor}	B^2	B^3	#Mod.
$(\tilde{x}, \tilde{y})$	92.89	97.77	98.69	46
$(\tilde{x}, \tilde{y}, \theta)$	94.99	98.06	98.75	59
$(\bar{y}, \theta)$	95.22	98.42	99.02	74
$(\bar{y}, \theta, \kappa)$	89.09	96.75	97.93	52

Tabelle 6.10: C_01

Merkmale	C_{cor}	B^2	B^3	#Mod.
$(\tilde{x}, \tilde{y})$	84.39	95.72	97.78	176
$(\tilde{x}, \tilde{y}, \theta)$	86.59	96.72	98.31	142
$(\bar{y}, \theta)$	94.38	98.50	99.36	153
$(\bar{y}, \theta, \kappa)$	88.79	97.13	98.19	124

Tabelle 6.11: X_01

Der Benchmark N_01, bestehend aus den zehn arabischen Ziffer, ist der einfachste verwendete Benchmark. Entsprechend wurden hier die höchsten Klassifikationsraten erreicht, wobei mit $(\bar{y}, \theta)$ sowohl die beste Klassifikationsrate erzielt wurde, als auch der kleinste Modellsatz zugrunde lag.

Die höchste Klassifikationsrate konnte im Benchmark B_01 mit der Merkmalskombination $(\tilde{x}, \tilde{y})$ erreicht werden. Allerdings geht dies auf Kosten der Größe des Modellsatzes, mit 99 Modellen. Bei annähernd gleichbleibender Klassifikationsrate konnte durch Hinzunahme des Merkmals θ eine Reduzierung der Modellsatzgröße um ca. 40% erzielt werden. Der Merkmalsvektor $(\bar{y}, \theta, \kappa)$ bereitete bei der Berechnung der Modellsätze Probleme. So mußten große Cluster der Modellberechnung zugrunde gelegt werden, da sonst für einige Merkmale sehr geringe Varianzen aufgetreten sind. Weiterhin wurde der Schwellwert für den minimalen Eigenwert der Kovarianzmatrix auf $\lambda_{\min} = 0.075$ gesetzt (vgl. Abschnitt 5.2.2).

Für den Merkmalsvektor $(\tilde{x}, \tilde{y})$ konnte im Benchmark C_01 maximal eine Klassifikationsrate $92,89\%$ erreicht werden (vgl. auch Tabelle 6.5). Durch die Hinzunahme von θ verbesserte sich die Klassifikationsrate auf $94,99\%$. Eine vergleichbare Klassifikationsrate $95,22\%$ konnte mit den Merkmalen $\bar{x}$ und θ erzielt werden, wenn man den etwas größeren Modellsätz berücksichtigt.

Für jeden Merkmalsvektor wurden die im vorangegangenen trainierten Modellsätze vereinigt und als Modellsatz für den Benchmark X_01 verwendet. Auffällig in Tabelle 6.11 ist das verhältnismäßig schlechte Ergebnis der Merkmale $(\tilde{x}, \tilde{y})$. Die Ursache liegt darin, daß diese Merkmale skalierungsinvariant sind, und somit Buchstaben, deren Groß- und Kleinschreibung bis auf eine Skalierung identisch sind, nicht unterscheiden können. Dies spiegelt

Merkmale	C_{cor}	B^2	B^3	#Modelle
$(\tilde{x}, \tilde{y})$	83.23	92.34	95.56	35
$(\tilde{x}, \tilde{y}, \theta)$	83.19	92.29	95.28	37
$(\bar{y}, \theta)$	83.78	94.73	96.95	41
$(\bar{y}, \theta, \kappa)$	84.55	92.96	95.58	31

Tabelle 6.12: M_01

sich in Beobachtung wieder, daß ca. 40% der Klassifikationsfehler auf eine Verwechselung von Groß- und Kleinbuchstaben zurückzuführen sind. Die höchste Klassifikationsrate von $94,38\%$ konnte mit dem Merkmalsvektor $(\bar{y}, \theta)$ erreicht werden. An dieser Stelle sei nochmal darauf hingewiesen, daß der in Abschnitt 3.1.3 vorgestellte Algorithmus zur Bestimmung der Refenzlinien nicht auf einzeln geschriebenen Zeichen anwenden läßt. Daher wurden für diese Zeichen die Referenzlinien manuell gesetzt.

Der Benchmark M_01 besteht aus Kleinbuchstaben, die manuell aus Wörtern segmentiert wurden. Das zugrundeliegende Alphabet besteht aus 18 verschiedenen Buchstaben. Die erzielten Klassifikationsraten liegen deutlich unter den Ergebnissen, die für C_01 erreicht wurden. Der Grund liegt zum einen in der hohen Variabilität der Zeichen. Zum anderen lassen sich selbst für den Menschen die Buchstaben außerhalb des Kontextes des jeweiligen Wörtes nicht eindeutig zuordnen (siehe Abbildung A.4).

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Arbeit wurde ein bekanntes Verfahren zur Angleichung zeitlich variierender Sequenzen erweitert. Die Idee war es, signifikante Regionen bzw. Abtastpunkte in einem Zeichen stärker und weniger wichtige Regionen schwächer zu gewichten. Dies wurde umgesetzt, indem der DTW-Algorithmus um eine Modellierung der statistischen Eigenschaften erweitert wurde. Dazu wurde aus der MAP-Annahme eine statistische Formulierung für die lokale Distanz, die die einzelnen Abtastpunkte mit einander vergleicht, hergeleitet.

Ferner mußte ein Verfahren zur Schätzung dieser statistischen Eigenschaften der Referenzmodelle entwickelt werden. Dieses Problem wurde in zwei Schritten gelöst. Zunächst wurden die unterschiedlichen Schreibweisen einzelner Zeichenklassen in der Trainingsmenge durch ein Clustering-Verfahren identifiziert. Anschließend wurden anhand dieser Partitionierung für jedes Cluster die Modellparameter geschätzt.

Der in dieser Arbeit verwendete Ansatz verlangt gewisse statistische Eigenschaften der Merkmale. Diese mußten bei der Wahl der Merkmale berücksichtigt werden. In Kapitel 6 wurden zusätzlich die Verteilungen der Merkmale untersucht.

Gegenüber der ungewichteten Methode konnte für die hier vorgestellte Methode bei gleicher Größe des Modellsatzes eine Verbesserung der Fehlerrate um den Faktor drei erreicht werden. Die Generalisierungsfähigkeiten des Verfahrens konnten erheblich verbessert werden. Zur Evaluation wurden einzeln geschriebene Buchstaben und Ziffern von 10 verschiedenen Schreiber aufgenommen. Die Ziffern konnten zu 99,4% richtig klassifiziert werden. Für die Großbuchstaben wurde eine Klassifikationsrate von 97,12% erzielt. Die Kleinbuchstaben wurden zu 95,22% richtig erkannt.

7.2 Ausblick

In dieser Arbeit wurde für die Wahrscheinlichkeit der lokalen Pfadübergänge eine Gleichverteilung angenommen. In Abschnitt 4.3.2 wurde bereits eine komplexere Modellierung der Wahrscheinlichkeiten der Pfadübergänge beschrieben. Diese Möglichkeit der Verfeinerung des Klassifikationsmodells ist sicherlich eine weitergehende Untersuchung wert.

Die verwendeten lokalen Pfadübergänge erlauben es, jede monotone zeitliche Variation zu modellieren. Aus diesem Grund wurde auf ein Resampling (nach konstanter Bogenlänge) des Schriftzuges verzichtet. Es konnten jedoch Fehlklassifikationen festgestellt werden, die durch eine gleichmäßigere Abtastung womöglich hätten verhindert werden können. Ein Resampling hätte auch zur Folge, daß sich die Menge der möglichen Warpingfunktionen einschränken ließe und somit eine Beschleunigung der Berechnung der DTW-Distanz einhergehen würde.

Bei der Herleitung der lokalen Distanz wurde für die Verteilung der Merkmalsvektoren eine multivariate Gaußverteilung vorausgesetzt. Eine Anpassung des Ansatzes an diskrete Werte würde es erlauben, diskrete Merkmale, wie etwa Kontextbitmaps, zu verwenden. Wobei eine Vergrößerung des Merkmalsvektors gleichzeitig bedeutet, daß die Trainingsmenge vergrößert werden muß, etwa für die Schätzung der Kovarianzmatrix.

Die in dieser Arbeit verwendeten Merkmale beschreiben das lokale Verhalten eines Schriftzuges. Merkmale, die das globale Verhalten eines Schriftzuges beschreiben, wie etwa die Gewelltheit oder die Linearität, könnten wesentliche Informationen zur besseren Klassifikation beitragen.

Bisher werden die Modelle für jede Zeichenklasse einzeln geschätzt. Dabei bleibt die relative Lage der Referenzmodelle im Musterraum zu Modellen anderer Zeichenklassen unberücksichtigt. Eine mögliche Verbesserung könnte daher eine Verfeinerung der Modellsätze, d.h. eine Erhöhung der Anzahl an Modellen, für Zeichenklassen einbringen, die häufig falsch klassifiziert werden.

Anhang A

Klassifikationsergebnisse

Im ersten Teil dieses Abschnitts wird examplarisch für die normierten Ortskoordinaten $(\tilde{x}, \tilde{y})$ das Klassifikationsergebnis beschrieben. Diese Merkmale eignen sich deshalb für eine gründliche Untersuchung, da sie sich in der graphischen Darstellung am anschaulichsten interpretieren lassen. Im zweiten Teil befinden sich die Klassifkationsergebnisse der verwendeten Benchmarks für verschiedene Kombinationen von Merkmalen.

A.1 Genaue Untersuchung der normierten Ortskoordinaten $(\tilde{x}, \tilde{y})$

Der Benchmark N_01 besteht aus den arabischen Ziffern 0–9 und ist mit nur zehn Bedeutungsklassen der einfachste Benchmark. Die Klassifkationsrate liegt bei 99.07%, wobei der Modellsatz aus 31 Modellen besteht. Insgesamt wurden also neun Ziffern falsch erkannt. Die Ziffern 0, 2 und 3 wurden alle fehlerfrei klassifiziert. Die ausführlichen Klassifkationsergebnisse sind in Tabelle A.1 aufgeführt.

	Klassifikationsergebnisse				Anz. der
Zeichen	C_{cor} [%]	B^2 [%]	B^3 [%]	Falsch [%]	Modelle
0	100.00	100.00	100.00	0.00	5
1	98.96	98.96	98.96	1.04	3
2	100.00	100.00	100.00	0.00	2
3	100.00	100.00	100.00	0.00	1
4	98.92	100.00	100.00	0.00	6
5	97.92	100.00	100.00	0.00	3
6	99.03	100.00	100.00	0.00	2
7	98.97	100.00	100.00	0.00	3
8	98.92	100.00	100.00	0.00	3
9	97.94	98.97	100.00	0.00	3
$\sum$	99.07	99.79	99.90	0.10	31

Tabelle A.1: Benchmark: N_01, Schwellwert c: 0.2, Merkmale: $(\tilde{x}, \tilde{y})$

Abbildung A.1 zeigt acht der neun falsch klassifizierten Ziffern. Abgesehen von der Ziffer 5, lassen sich die Fehlklassifikationen auf die Deformationen der Testmuster zurückführen.

Beide 5er werden fälschlicherweise als 3 erkannt. Jedoch entspricht jeweils das zweitbeste Modell der richtigen Zeichenklasse. Die Distanzen zum erst- und zweitbesten Modell sind ähnlich niedrig. Die beiden Testmuster wurden wir folgt geschrieben: zuerst der waagerechte Strich von links nach rechts und dann der Rest der Ziffer. Obwohl insgesamt drei verschiedene Modelle für die Ziffer 5 trainiert wurden, ist diese Schreibweise der 5 nicht im Modellsatz enthalten. Im ersten Modell wird die 5 in einem Strich von rechts oben nach links unten geschrieben. Im zweiten, bzw. dritten Modell wird zunächst der Rumpf von oben nach unten gezeichnet und das Muster wird mit dem oberen waagerechten Strich von links nach rechts, bzw. von rechts nach links abgeschlossen.

Die Angleichung dieser 5 an das Modell der 3 macht nur bei dem senkrechten Strich der 5 nennenswerte Fehler. Ein Angleichung an die Modell 5 erzeugt entlang des waagerechten Strich den Hauptanteil des Gesamtfehler. Dieser Fehler hätte mit einer verfeinerten Modellselektion vermieden werden können.

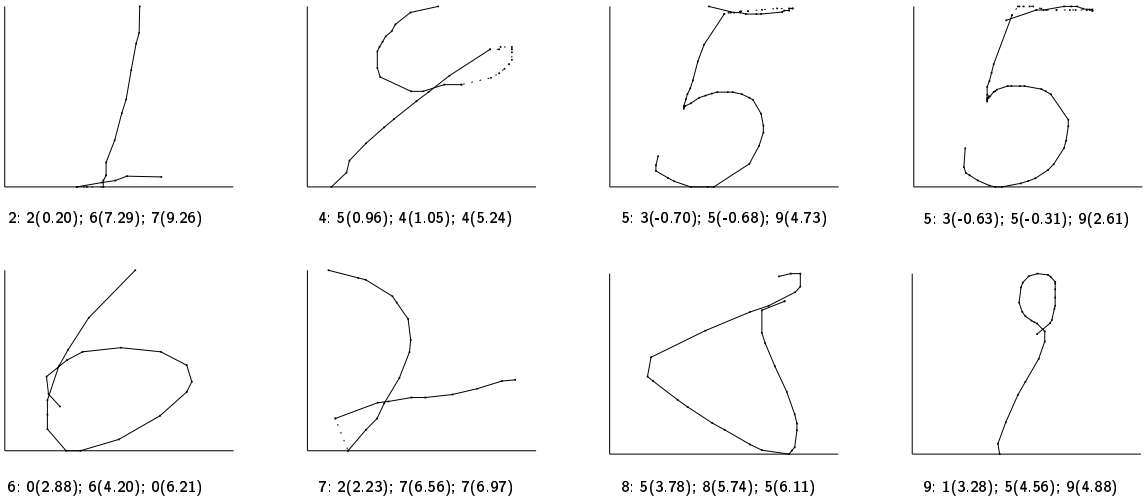


Abbildung A.1: Falsch erkannte Ziffern aus dem N_01 Benchmark. Merkmale: $(\tilde{x}, \tilde{y})$. (Erkläuterung der Bildunterschriften: die erste Ziffer gibt die tatsächliche Klasse des Musters wieder. Darauf folgt die Klassenzugehörigkeit, wie sie vom Erkenner bestimmt wurde. Jeweils in Klammer ist die Distanz zum Modell angegeben.)

Tabelle A.2 zeigt die Ergebnisse für den B_01 Benchmark. Mit einer relativ hohen Anzahl an Modellen konnte eine Klassifikationsrate von 97.12% erreicht werden. In 99.35% der Fälle war die korrekte Zeichenklasse unter den erst- und zweitbesten Modellen. Zwölf Buchstaben wurden fehlerfrei erkannt. Die häufigsten Verwechselungen waren: C $\stackrel{7}{\rightarrow}$ L[1], I $\stackrel{7}{\rightarrow}$ F, J $\stackrel{4}{\rightarrow}$ I, K $\stackrel{6}{\rightarrow}$ U, N $\stackrel{5}{\rightarrow}$ W und U $\stackrel{8}{\rightarrow}$ V.

Abbildung A.2 zeigt typische Beispiele für die häufigsten Fehlklassifikationen. In der ersten Zeile sind die falsch erkannten Muster aufgetragen. In der zweiten Zeile sind die Referenzmuster der Modelle zu sehen, die die geringste Distanz zu dem darüber stehenden Testmuster haben. Die Verwechselung J $\rightarrow$ I läßt darauf zurückführen, daß identische Schreibweisen für I und J existieren. Ähnliches gilt für das falsch klassifizierte U.

[1] C $\stackrel{7}{\rightarrow}$ L bedeutet der Buchstabe C wurde sieben mal als L erkannt.

	Klassifikationsergebnisse				Anz. der
Zeichen	C_{cor} [%]	B^2 [%]	B^3 [%]	Falsch [%]	Modelle
A	96.74	100.00	100.00	0.00	6
B	100.00	100.00	100.00	0.00	2
C	92.05	100.00	100.00	0.00	2
D	98.90	100.00	100.00	0.00	3
E	100.00	100.00	100.00	0.00	5
F	100.00	100.00	100.00	0.00	7
G	95.40	96.55	98.85	1.15	7
H	97.65	100.00	100.00	0.00	6
I	83.91	95.40	96.55	3.45	3
J	93.98	100.00	100.00	0.00	3
K	92.11	100.00	100.00	0.00	3
L	98.85	100.00	100.00	0.00	2
M	95.38	100.00	100.00	0.00	4
N	93.83	95.06	98.77	1.23	5
O	100.00	100.00	100.00	0.00	4
P	100.00	100.00	100.00	0.00	3
Q	97.53	100.00	100.00	0.00	3
R	100.00	100.00	100.00	0.00	2
S	100.00	100.00	100.00	0.00	1
T	100.00	100.00	100.00	0.00	6
U	89.87	96.20	97.47	2.53	3
V	98.82	100.00	100.00	0.00	2
W	100.00	100.00	100.00	0.00	4
X	100.00	100.00	100.00	0.00	5
Y	100.00	100.00	100.00	0.00	5
Z	100.00	100.00	100.00	0.00	3
$\sum$	97.12	99.35	99.68	0.32	99

Tabelle A.2: Benchmark: B_01, Schwellwert c: 0.2, Merkmale: $(\tilde{x}, \tilde{y})$.

Die anderen Fehlklassifikationen lassen sich nicht so gut erklären. Allerdings sollte man bedenken, daß lediglich die Ortskoordinaten als Merkmale verwendet wurde. Diese Merkmale können nicht die leichte Krümmung des Testmusters C aus Abb. A.2 beschreiben, die es als C identifiziert. Ein Hinweis darauf, daß Merkmale verwendet werden sollten, die das lokale Verhalten der Schrift in der Umgebung des Abtastpunktes beschreiben.

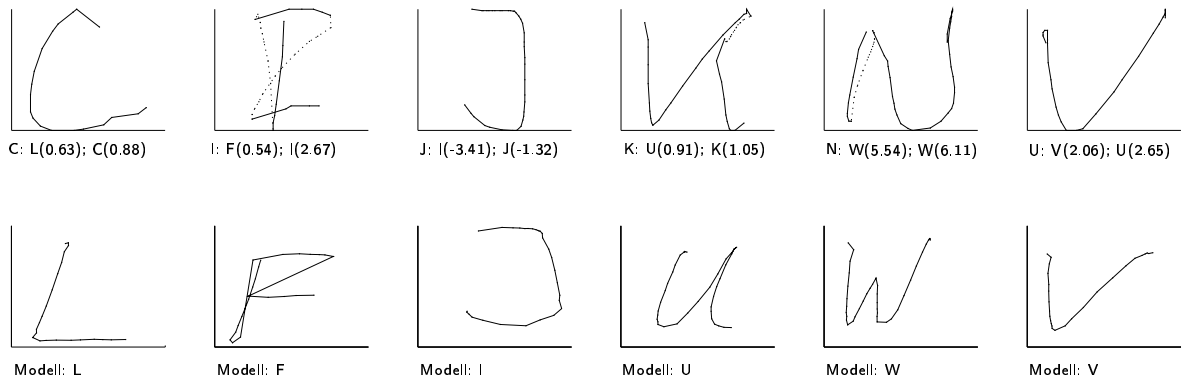


Abbildung A.2: Benchmark: B_01. Merkmale: $(\tilde{x}, \tilde{y})$. Beispiele für die häufigsten Fehlklassifikationen. Erläuterung zur Beschriftung siehe Fußnote 2.

Für den Benchmark C_01 konnte eine Klassifikationsrate von $92,89\%$ erreicht werden. Unter Berücksichtigung des zweitbesten Modells, konnten immerhin $97,77\%$ der Testmuster richtig erkannt werden. Im Schnitt wurden $1,8$ Modelle pro Zeichenklasse trainiert. Die häufigsten Verwechselungen waren: h $\overset{14}{\rightarrow}$ n, q $\overset{10}{\rightarrow}$ g, u $\overset{8}{\rightarrow}$ n, k $\overset{7}{\rightarrow}$ n, y $\overset{6}{\rightarrow}$ r und e $\overset{6}{\rightarrow}$ l.

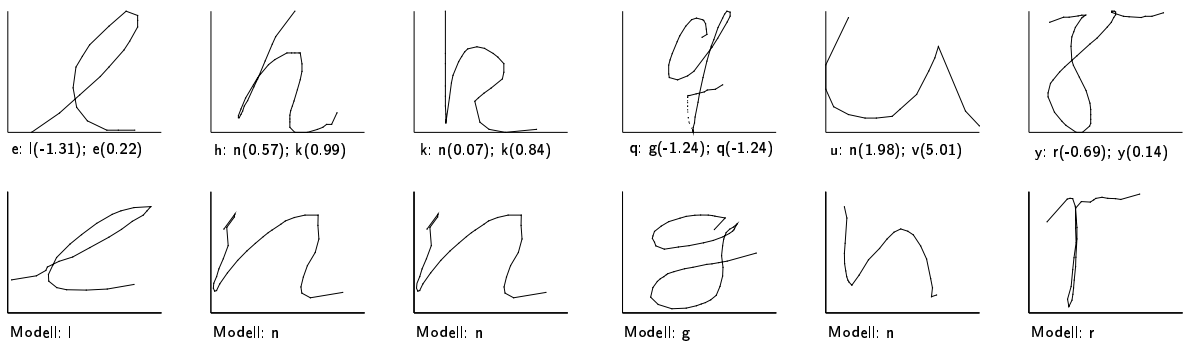


Abbildung A.3: Benchmark: C_01. Merkmale: $(\tilde{x}, \tilde{y})$. Beispiele für die häufigsten Fehlklassifikationen. Erläuterung zur Beschriftung siehe Fußnote 2.

[2]In der ersten Zeile sind die falsch erkannten Muster aufgetragen. In der zweiten Zeile sind die Referenzmuster der Modelle zu sehen, die die geringste Distanz zu dem darüber stehenden Testmuster haben. Die Bildunterschriften in der ersten Zeile, sind wie folgt zu interpretieren. Der erste Buchstabe gibt die tatsächliche Klasse wieder. Das erst- und zweitbeste Modell ist durch ein Semikolon getrennt. Die Distanzen der Referenzmodell zum Testmuster stehen jeweils in Klammern.

Zeichen	Klassifikationsergebnisse				Anz. der Modelle
	C_{cor} [%]	B^2 [%]	B^3 [%]	Falsch [%]	
a	83.12	96.10	98.70	1.30	1
b	95.45	98.86	98.86	1.14	2
c	100.00	100.00	100.00	0.00	1
d	100.00	100.00	100.00	0.00	1
e	93.26	100.00	100.00	0.00	1
f	91.40	96.77	96.77	3.23	1
g	96.34	100.00	100.00	0.00	1
h	70.89	81.01	89.87	10.13	2
i	95.10	99.02	100.00	0.00	1
j	100.00	100.00	100.00	0.00	1
k	83.13	96.39	100.00	0.00	2
l	89.25	96.77	96.77	3.23	2
m	98.25	100.00	100.00	0.00	2
n	92.94	100.00	100.00	0.00	3
o	96.63	97.75	98.88	1.12	3
p	95.83	95.83	95.83	4.17	2
q	88.35	99.03	100.00	0.00	1
r	93.07	98.02	98.02	1.98	1
s	98.23	98.23	98.23	1.77	2
t	89.72	96.26	99.07	0.93	3
u	90.48	98.81	98.81	1.19	2
v	97.96	100.00	100.00	0.00	2
w	100.00	100.00	100.00	0.00	2
x	95.19	99.04	99.04	0.96	3
y	91.01	98.88	100.00	0.00	2
z	89.62	95.28	97.17	2.83	2
$\sum$	92.89	97.77	98.69	1.31	46

Tabelle A.3: Benchmark: C_01, Schwellwert c: 0.7, Merkmale: $(\tilde{x}, \tilde{y})$.

A.2 Alle Ergebnisse. Sortiert nach BENCHMARKS

A.2.1 N_01: Ziffern

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
0	100.00	100.00	100.00	5
1	98.96	98.96	98.96	3
2	100.00	100.00	100.00	2
3	100.00	100.00	100.00	1
4	98.92	100.00	100.00	6
5	97.92	100.00	100.00	3
6	99.03	100.00	100.00	2
7	98.97	100.00	100.00	3
8	98.92	100.00	100.00	3
9	97.94	98.97	100.00	3
$\sum$	99.07	99.79	99.90	31

Tabelle A.4: N_01. $(\tilde{x}, \tilde{y})$

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
0	100.00	100.00	100.00	3
1	98.96	98.96	98.96	1
2	97.62	98.81	100.00	2
3	100.00	100.00	100.00	1
4	100.00	100.00	100.00	3
5	100.00	100.00	100.00	4
6	100.00	100.00	100.00	2
7	91.75	100.00	100.00	3
8	100.00	100.00	100.00	3
9	96.91	100.00	100.00	2
$\sum$	98.52	99.78	99.90	24

Tabelle A.5: N_01. $(\tilde{x}, \tilde{y}, \theta)$

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
0	100.00	100.00	100.00	3
1	98.96	98.96	100.00	2
2	96.43	100.00	100.00	1
3	100.00	100.00	100.00	1
4	100.00	100.00	100.00	3
5	98.96	100.00	100.00	4
6	100.00	100.00	100.00	2
7	100.00	100.00	100.00	3
8	100.00	100.00	100.00	3
9	100.00	100.00	100.00	2
$\sum$	99.43	99.90	100.00	24

Tabelle A.6: N_01. $(\bar{y}, \theta)$.

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
0	100.00	100.00	100.00	2
1	96.88	97.92	98.96	2
2	98.81	100.00	100.00	2
3	100.00	100.00	100.00	1
4	98.92	100.00	100.00	3
5	98.96	100.00	100.00	5
6	100.00	100.00	100.00	2
7	95.88	98.97	98.97	3
8	100.00	100.00	100.00	3
9	100.00	100.00	100.00	3
$\sum$	98.94	99.69	99.79	26

Tabelle A.7: N_01. $(\bar{y}, \theta, \kappa)$.

A.2.2 B_01: Großbuchstaben

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
A	96.74	100.00	100.00	6
B	100.00	100.00	100.00	2
C	92.05	100.00	100.00	2
D	98.90	100.00	100.00	3
E	100.00	100.00	100.00	5
F	100.00	100.00	100.00	7
G	95.40	96.55	98.85	7
H	97.65	100.00	100.00	6
I	83.91	95.40	96.55	3
J	93.98	100.00	100.00	3
K	92.11	100.00	100.00	3
L	98.85	100.00	100.00	2
M	95.38	100.00	100.00	4
N	93.83	95.06	98.77	5
O	100.00	100.00	100.00	4
P	100.00	100.00	100.00	3
Q	97.53	100.00	100.00	3
R	100.00	100.00	100.00	2
S	100.00	100.00	100.00	1
T	100.00	100.00	100.00	6
U	89.87	96.20	97.47	3
V	98.82	100.00	100.00	2
W	100.00	100.00	100.00	4
X	100.00	100.00	100.00	5
Y	100.00	100.00	100.00	5
Z	100.00	100.00	100.00	3
$\sum$	97.12	99.35	99.68	99

Tabelle A.8: B_01. $(\tilde{x}, \tilde{y})$.

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
A	96.74	100.00	100.00	3
B	100.00	100.00	100.00	1
C	100.00	100.00	100.00	1
D	95.60	98.90	100.00	1
E	98.90	100.00	100.00	4
F	97.94	97.94	98.97	2
G	95.40	100.00	100.00	4
H	95.29	100.00	100.00	4
I	82.76	93.10	94.25	5
J	81.93	91.57	93.98	2
K	90.79	100.00	100.00	1
L	97.70	98.85	100.00	2
M	100.00	100.00	100.00	2
N	100.00	100.00	100.00	2
O	100.00	100.00	100.00	2
P	100.00	100.00	100.00	1
Q	100.00	100.00	100.00	1
R	97.62	97.62	98.81	1
S	100.00	100.00	100.00	1
T	100.00	100.00	100.00	3
U	91.14	100.00	100.00	2
V	100.00	100.00	100.00	2
W	98.46	100.00	100.00	2
X	100.00	100.00	100.00	4
Y	100.00	100.00	100.00	4
Z	100.00	100.00	100.00	2
$\sum$	96.93	99.15	99.46	59

Tabelle A.9: B_01. $(\tilde{x}, \tilde{y}, \theta)$.

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
A	95.65	100.00	100.00	3
B	98.86	100.00	100.00	1
C	98.86	100.00	100.00	1
D	96.70	98.90	100.00	1
E	100.00	100.00	100.00	3
F	96.91	97.94	97.94	2
G	97.70	98.85	100.00	3
H	97.65	98.82	100.00	4
I	90.80	97.70	98.85	5
J	83.13	93.98	97.59	2
K	86.84	94.74	94.74	1
L	88.51	98.85	100.00	2
M	100.00	100.00	100.00	2
N	100.00	100.00	100.00	2
O	98.78	100.00	100.00	2
P	100.00	100.00	100.00	1
Q	98.77	100.00	100.00	1
R	97.62	98.81	98.81	1
S	100.00	100.00	100.00	1
T	100.00	100.00	100.00	3
U	92.41	97.47	100.00	2
V	100.00	100.00	100.00	2
W	95.38	100.00	100.00	2
X	98.81	100.00	100.00	3
Y	100.00	100.00	100.00	3
Z	100.00	100.00	100.00	2
$\sum$	96.67	99.08	99.54	55

Tabelle A.10: B_01. $(\bar{y}, \theta)$.

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
A	89.13	100.00	100.00	3
B	100.00	100.00	100.00	1
C	100.00	100.00	100.00	1
D	97.80	100.00	100.00	1
E	95.60	95.60	98.90	4
F	91.75	91.75	92.78	2
G	96.55	98.85	100.00	1
H	95.29	98.82	100.00	3
I	89.66	100.00	100.00	4
J	75.90	91.57	93.98	1
K	78.95	92.11	93.42	1
L	93.10	100.00	100.00	1
M	98.46	100.00	100.00	2
N	100.00	100.00	100.00	2
O	98.78	100.00	100.00	2
P	99.02	100.00	100.00	1
Q	96.30	98.77	100.00	1
R	95.24	97.62	98.81	1
S	100.00	100.00	100.00	1
T	100.00	100.00	100.00	3
U	98.73	100.00	100.00	2
V	100.00	100.00	100.00	1
W	92.31	100.00	100.00	1
X	98.81	100.00	100.00	3
Y	97.56	100.00	100.00	2
Z	96.05	98.68	98.68	1
$\sum$	95.19	98.61	99.10	46

Tabelle A.11: B_01. $(\bar{y}, \theta, \kappa)$.

A.2.3 C_01: Kleinbuchstaben

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
a	83.12	96.10	98.70	1
b	95.45	98.86	98.86	2
c	100.00	100.00	100.00	1
d	100.00	100.00	100.00	1
e	93.26	100.00	100.00	1
f	91.40	96.77	96.77	1
g	96.34	100.00	100.00	1
h	70.89	81.01	89.87	2
i	95.10	99.02	100.00	1
j	100.00	100.00	100.00	1
k	83.13	96.39	100.00	2
l	89.25	96.77	96.77	2
m	98.25	100.00	100.00	2
n	92.94	100.00	100.00	3
o	96.63	97.75	98.88	3
p	95.83	95.83	95.83	2
q	88.35	99.03	100.00	1
r	93.07	98.02	98.02	1
s	98.23	98.23	98.23	2
t	89.72	96.26	99.07	3
u	90.48	98.81	98.81	2
v	97.96	100.00	100.00	2
w	100.00	100.00	100.00	2
x	95.19	99.04	99.04	3
y	91.01	98.88	100.00	2
z	89.62	95.28	97.17	2
$\sum$	92.89	97.77	98.69	46

Tabelle A.12: C_01. $(\tilde{x}, \tilde{y})$.

Z.	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	#M
a	84.42	96.10	97.40	1
b	97.73	98.86	98.86	2
c	100.00	100.00	100.00	1
d	100.00	100.00	100.00	1
e	92.13	100.00	100.00	1
f	93.55	96.77	96.77	3
g	92.68	98.78	98.78	1
h	79.75	84.81	93.67	2
i	97.06	100.00	100.00	2
j	100.00	100.00	100.00	1
k	89.16	98.80	100.00	2
l	88.17	96.77	96.77	2
m	100.00	100.00	100.00	4
n	95.29	100.00	100.00	4
o	97.75	98.88	98.88	4
p	95.83	95.83	95.83	2
q	99.03	100.00	100.00	1
r	96.04	99.01	99.01	2
s	98.23	98.23	98.23	2
t	98.13	99.07	99.07	3
u	92.86	97.62	100.00	3
v	100.00	100.00	100.00	3
w	100.00	100.00	100.00	4
x	99.04	99.04	99.04	3
y	93.26	96.63	98.88	3
z	89.62	94.34	96.23	2
$\sum$	94.99	98.06	98.75	59

Tabelle A.13: C_01. $(\tilde{x}, \tilde{y}, \theta)$.

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
a	89.61	93.51	97.40	1
b	98.86	100.00	100.00	3
c	97.75	100.00	100.00	2
d	100.00	100.00	100.00	1
e	98.88	100.00	100.00	2
f	96.77	98.92	98.92	3
g	92.68	100.00	100.00	1
h	93.67	100.00	100.00	3
i	98.04	100.00	100.00	2
j	100.00	100.00	100.00	3
k	95.18	100.00	100.00	4
l	86.02	86.02	88.17	3
m	94.74	98.25	98.25	2
n	96.47	97.65	98.82	2
o	87.64	98.88	100.00	3
p	97.92	98.96	98.96	4
q	99.03	100.00	100.00	3
r	95.05	97.03	99.01	3
s	99.12	99.12	100.00	3
t	100.00	100.00	100.00	3
u	85.71	98.81	98.81	3
v	91.84	98.98	98.98	5
w	94.12	97.06	97.06	3
x	100.00	100.00	100.00	4
y	87.64	96.63	100.00	5
z	99.06	99.06	100.00	3
$\sum$	95.22	98.42	99.01	74

Tabelle A.14: C_01. $(\bar{y}, \theta)$.

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
a	87.01	94.81	97.40	1
b	89.77	95.45	100.00	2
c	88.76	100.00	100.00	1
d	100.00	100.00	100.00	1
e	100.00	100.00	100.00	1
f	91.40	96.77	96.77	3
g	87.80	96.34	97.56	1
h	94.94	96.20	96.20	2
i	75.49	93.14	97.06	2
j	100.00	100.00	100.00	2
k	65.06	98.80	100.00	3
l	86.02	89.25	89.25	2
m	77.19	85.96	87.72	1
n	70.59	97.65	98.82	2
o	94.38	100.00	100.00	2
p	98.96	100.00	100.00	2
q	100.00	100.00	100.00	1
r	78.22	94.06	99.01	2
s	91.15	100.00	100.00	2
t	93.46	93.46	97.20	3
u	82.14	97.62	100.00	2
v	86.73	95.92	98.98	3
w	94.12	97.06	97.06	2
x	98.08	100.00	100.00	3
y	95.51	97.75	97.75	4
z	89.62	95.28	95.28	2
$\sum$	89.09	96.75	97.93	52

Tabelle A.15: C_01. $(\bar{y}, \theta, \kappa)$.

A.2.4 X_01: Ziffern, Klein- und Großbuchstaben

Zeichen	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	Anz. der Modelle
0	72.09	93.02	98.84	5
1	98.30	98.86	98.86	3
2	92.68	98.17	99.39	2
3	99.41	100.00	100.00	1
4	95.95	97.69	98.84	6
5	97.73	100.00	100.00	3
6	93.44	98.91	100.00	2
7	96.61	97.74	98.87	3
8	97.69	98.27	99.42	3
9	88.14	97.18	98.87	3
A	91.86	97.09	98.84	6
B	99.40	100.00	100.00	2
C	65.48	93.45	97.62	2
D	98.83	99.42	100.00	3
E	90.64	97.08	97.66	5
F	99.44	100.00	100.00	7
G	84.43	91.62	93.41	7
H	92.12	95.15	96.36	6
I	66.47	91.02	92.22	3
J	83.44	92.64	96.32	3
K	82.05	93.59	95.51	3
L	79.64	89.82	91.62	2
M	88.97	94.48	96.55	4
N	93.79	93.79	94.41	5
O	64.81	95.06	98.77	4
P	72.53	100.00	100.00	3
Q	96.89	98.14	98.76	3
R	98.17	98.78	98.78	2
S	74.73	98.35	100.00	1
T	83.73	90.96	96.39	6
U	74.84	80.50	88.05	3
V	80.00	95.76	97.58	2
W	73.79	97.24	98.62	4
X	46.95	92.07	98.78	5
Y	53.70	90.12	95.06	5
Z	85.26	94.87	98.08	3
a	80.89	94.27	97.45	1
b	92.26	97.02	98.21	2
c	71.01	98.82	100.00	1
d	99.37	99.37	99.37	1
e	94.08	99.41	99.41	1
f	89.60	97.69	98.27	1
g	95.06	98.15	98.77	1
h	73.58	84.28	90.57	2
i	93.96	98.35	99.45	1
j	93.89	98.89	100.00	1
k	76.07	93.25	95.71	2
l	92.49	97.69	98.27	2
m	99.27	100.00	100.00	2
n	89.70	96.36	98.18	3
o	62.13	76.33	92.90	3
p	77.84	95.45	95.45	2
q	67.21	98.91	100.00	1
r	94.48	97.79	98.90	1
s	50.78	96.37	98.96	2
t	92.51	97.33	99.47	3
u	83.54	95.73	97.56	2
v	81.46	98.88	100.00	2
w	97.30	98.65	99.32	2
x	91.85	98.91	99.46	3
y	80.47	92.90	97.63	2
z	57.53	93.01	96.77	2
$\sum$	84.39	95.72	97.78	176

Tabelle A.16: X_01. $(\tilde{x}, \tilde{y})$.

Zeichen	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	Anz. der Modelle
0	84.88	97.09	100.00	3
1	98.86	98.86	98.86	1
2	90.24	95.73	98.17	2
3	99.41	100.00	100.00	1
4	99.42	99.42	99.42	3
5	97.73	99.43	99.43	4
6	100.00	100.00	100.00	2
7	88.14	93.79	94.92	3
8	96.53	100.00	100.00	3
9	84.18	96.61	98.31	2
A	94.77	100.00	100.00	3
B	100.00	100.00	100.00	1
C	81.55	100.00	100.00	1
D	96.49	98.83	99.42	1
E	97.08	98.83	98.83	4
F	95.48	97.74	97.74	2
G	83.23	92.81	95.21	4
H	92.73	95.76	96.36	4
I	85.03	89.22	90.42	5
J	79.75	87.73	89.57	2
K	87.18	94.23	98.72	1
L	95.81	97.60	98.20	2
M	91.03	96.55	99.31	2
N	96.27	97.52	98.76	2
O	31.48	91.36	100.00	2
P	71.98	99.45	99.45	1
Q	99.38	99.38	100.00	1
R	97.56	98.17	98.17	1
S	69.78	100.00	100.00	1
T	98.80	99.40	100.00	3
U	76.73	88.68	94.97	2
V	85.45	96.97	97.58	2
W	79.31	95.17	98.62	2
X	50.00	99.39	100.00	4
Y	82.72	98.15	99.38	4
Z	66.67	93.59	100.00	2
a	80.89	92.36	96.18	1
b	94.05	98.81	98.81	2
c	67.46	100.00	100.00	1
d	97.47	99.37	99.37	1
e	91.12	100.00	100.00	1
f	94.80	97.11	98.27	3
g	91.98	98.15	98.15	1
h	83.65	87.42	89.94	2
i	95.60	99.45	99.45	2
j	99.44	100.00	100.00	1
k	73.01	93.25	94.48	2
l	92.49	98.27	98.27	2
m	100.00	100.00	100.00	4
n	90.91	95.15	99.39	4
o	64.50	80.47	98.82	4
p	81.25	94.89	96.59	2
q	81.42	100.00	100.00	1
r	95.58	98.90	98.90	2
s	68.91	98.96	98.96	2
t	96.79	99.47	99.47	3
u	86.59	98.17	98.78	3
v	72.47	95.51	100.00	3
w	97.97	99.32	100.00	4
x	96.74	99.46	99.46	3
y	73.96	90.53	97.63	3
z	73.66	94.09	94.62	2
$\sum$	86.59	96.72	98.31	142

Tabelle A.17: X_01. $(\tilde{x}, \tilde{y}, \theta)$.

Zeichen	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	Anz. der Modelle
0	93.02	99.42	100.00	3
1	96.02	97.73	98.86	2
2	81.10	88.41	93.29	1
3	99.41	99.41	100.00	1
4	96.53	98.27	100.00	3
5	98.86	100.00	100.00	4
6	97.81	98.91	100.00	2
7	96.61	97.74	98.31	3
8	100.00	100.00	100.00	3
9	96.61	98.31	99.44	2
A	96.51	100.00	100.00	3
B	99.40	100.00	100.00	1
C	98.81	100.00	100.00	1
D	97.66	98.83	99.42	1
E	100.00	100.00	100.00	3
F	97.18	97.74	97.74	2
G	89.82	97.01	99.40	3
H	97.58	99.39	99.39	4
I	86.83	95.81	98.20	5
J	84.05	96.93	98.16	2
K	85.26	96.79	97.44	1
L	90.42	97.01	100.00	2
M	100.00	100.00	100.00	2
N	99.38	99.38	99.38	2
O	22.84	96.91	100.00	2
P	100.00	100.00	100.00	1
Q	97.52	98.76	99.38	1
R	98.17	98.78	98.78	1
S	100.00	100.00	100.00	1
T	100.00	100.00	100.00	3
U	93.71	95.60	98.74	2
V	98.79	100.00	100.00	2
W	96.55	99.31	99.31	2
X	99.39	100.00	100.00	3
Y	95.06	100.00	100.00	3
Z	96.79	99.36	100.00	2
a	92.99	95.54	98.73	1
b	98.81	100.00	100.00	3
c	98.82	100.00	100.00	2
d	98.10	99.37	99.37	1
e	98.22	100.00	100.00	2
f	97.69	97.69	98.27	3
g	95.06	98.77	98.77	1
h	94.34	98.74	100.00	3
i	97.80	99.45	99.45	2
j	100.00	100.00	100.00	3
k	73.62	95.09	100.00	4
l	95.38	97.11	98.27	3
m	94.89	97.81	99.27	2
n	95.15	97.58	99.39	2
o	85.21	94.67	99.41	3
p	98.86	99.43	99.43	4
q	99.45	100.00	100.00	3
r	95.03	97.79	98.90	3
s	99.48	99.48	100.00	3
t	97.33	98.40	98.93	3
u	87.80	99.39	99.39	3
v	94.38	99.44	99.44	5
w	95.95	98.65	98.65	3
x	99.46	100.00	100.00	4
y	90.53	97.04	99.41	5
z	99.46	99.46	100.00	3
$\sum$	94.38	98.50	99.36	153

Tabelle A.18: X_01. $(\bar{y}, \theta)$.

Zeichen	Klassifikationsergebnisse C_{cor} [%]	B^2 [%]	B^3 [%]	Anz. der Modelle
0	97.09	100.00	100.00	2
1	90.34	92.05	92.05	2
2	95.73	96.34	97.56	2
3	94.67	98.22	98.82	1
4	94.80	95.95	95.95	3
5	90.91	97.16	97.73	5
6	98.91	100.00	100.00	2
7	91.53	92.09	92.66	3
8	99.42	99.42	99.42	3
9	92.09	96.05	96.05	3
A	70.93	97.67	100.00	3
B	97.02	100.00	100.00	1
C	99.40	100.00	100.00	1
D	96.49	99.42	99.42	1
E	93.57	96.49	96.49	4
F	83.62	92.09	92.66	2
G	64.67	87.43	97.01	1
H	87.88	99.39	99.39	3
I	86.23	98.20	99.40	4
J	63.80	90.18	91.41	1
K	62.18	86.54	89.74	1
L	83.23	99.40	100.00	1
M	97.93	100.00	100.00	2
N	98.76	98.76	99.38	2
O	12.96	98.15	100.00	2
P	97.25	100.00	100.00	1
Q	95.03	99.38	99.38	1
R	95.73	97.56	98.17	1
S	96.15	100.00	100.00	1
T	99.40	99.40	99.40	3
U	97.48	98.74	99.37	2
V	98.18	100.00	100.00	1
W	93.10	100.00	100.00	1
X	99.39	100.00	100.00	3
Y	90.12	99.38	100.00	2
Z	66.03	94.23	99.36	1
a	91.72	96.82	98.09	1
b	88.10	96.43	98.21	2
c	89.94	100.00	100.00	1
d	98.73	99.37	99.37	1
e	99.41	99.41	99.41	1
f	94.22	98.27	98.27	3
g	91.36	96.91	97.53	1
h	94.34	96.23	96.86	2
i	78.02	90.66	97.80	2
j	100.00	100.00	100.00	2
k	57.67	95.71	99.39	3
l	96.53	100.00	100.00	2
m	81.75	90.51	93.43	1
n	74.55	95.76	98.18	2
o	95.86	99.41	100.00	2
p	99.43	100.00	100.00	2
q	99.45	100.00	100.00	1
r	73.48	88.40	96.69	2
s	92.75	100.00	100.00	2
t	89.84	91.98	91.98	3
u	84.76	98.78	100.00	2
v	84.27	96.63	99.44	3
w	95.95	97.97	98.65	2
x	96.74	98.37	98.37	3
y	93.49	98.22	98.82	4
z	90.32	96.24	96.24	2
$\sum$	88.79	97.13	98.19	124

Tabelle A.19: X_01. $(\bar{y}, \theta, \kappa)$.

A.2.5 M_01: aus Wörtern segmentierte Buchstaben

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
a	74.53	89.31	94.34	1
c	84.21	86.84	94.74	1
d	94.92	98.31	100.00	1
e	74.29	95.05	96.70	1
f	86.11	88.89	93.06	2
g	90.00	98.46	99.23	1
h	95.58	96.94	98.30	3
i	79.61	83.91	88.20	2
l	84.12	95.28	97.85	2
m	87.50	93.75	96.88	3
n	52.01	80.59	87.91	3
o	89.53	94.76	97.01	1
r	73.09	88.36	92.36	2
s	80.12	91.99	95.85	1
t	89.10	96.66	98.95	2
u	76.13	89.03	93.55	4
w	93.94	96.97	96.97	4
y	93.29	96.95	98.17	1
$\sum$	83.23	92.34	95.56	35

Tabelle A.20: M_01. $(\tilde{x}, \tilde{y})$.

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
a	72.96	86.48	94.65	1
c	92.11	94.74	97.37	2
d	94.92	98.31	98.31	1
e	74.76	93.63	97.17	1
f	81.94	86.11	87.50	2
g	86.15	96.15	98.46	2
h	81.29	86.39	92.18	2
i	70.82	78.76	82.40	3
l	75.54	86.70	91.42	2
m	84.38	90.63	93.75	3
n	80.95	97.07	98.90	2
o	92.27	97.26	98.00	1
r	81.82	94.55	97.45	2
s	87.54	93.47	94.36	1
t	83.66	95.96	98.95	3
u	62.58	85.81	94.19	3
w	100.00	100.00	100.00	4
y	93.90	99.39	100.00	2
$\sum$	83.20	92.30	95.28	37

Tabelle A.21: M_01. $(\tilde{x}, \tilde{y}, \theta)$.

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
a	63.21	89.94	93.71	1
c	89.47	97.37	97.37	3
d	99.15	99.15	100.00	1
e	81.37	95.28	98.11	1
f	86.11	97.22	97.22	4
g	73.08	93.85	96.92	1
h	84.01	91.84	95.92	2
i	79.61	92.70	95.92	3
l	97.00	99.14	100.00	3
m	84.38	96.88	96.88	2
n	61.90	87.18	95.24	2
o	89.78	95.51	98.25	3
r	84.36	94.91	96.73	2
s	91.39	94.66	95.55	1
t	91.39	96.13	97.36	3
u	65.81	86.45	92.90	3
w	93.94	96.97	96.97	3
y	92.07	100.00	100.00	3
$\sum$	83.78	94.73	96.95	41

Tabelle A.22: M_01. $(\bar{y}, \theta, \kappa)$.

Z.	Klassifikationsergebnisse			#M
	C_{cor} [%]	B^2 [%]	B^3 [%]	
a	81.76	92.45	96.23	1
c	84.21	92.11	94.74	1
d	96.61	98.31	99.15	1
e	94.81	97.41	97.88	1
f	88.89	93.06	94.44	2
g	84.62	100.00	100.00	2
h	77.55	89.80	93.54	2
i	71.03	81.97	88.41	3
l	93.99	98.71	99.57	2
m	93.75	93.75	96.88	2
n	71.79	88.64	91.21	2
o	94.51	98.75	99.75	1
r	77.09	94.55	97.82	1
s	94.07	96.74	98.22	1
t	72.58	82.25	87.52	3
u	56.13	75.48	85.16	3
w	96.97	100.00	100.00	1
y	91.46	99.39	100.00	2
$\sum$	84.55	92.96	95.58	31

Tabelle A.23: M_01. $(\bar{y}, \theta, \kappa)$.

A.3 Beispiele aus M_01

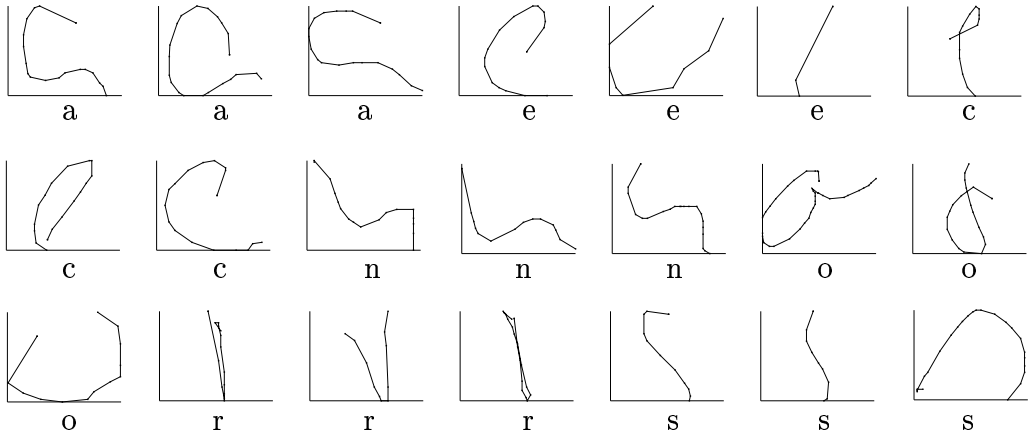


Abbildung A.4: Musterbeispiele aus dem Benchmark M_01. Die Beschriftung der einzelnen Muster gibt die tatsächliche Klassenzugehörigkeit wieder.

Literaturverzeichnis

[1] Bozinovic, Radmilo M. und Sargur N. Srihari: *Off-Line Cursive Script Word Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(1):68–83, Januar 1989.

[2] Bunke, H., M. Roth und E.G. Schukat-Talamazzini: *Off-Line Cursive Handwriting Recognition Using Hidden Markov Models.* Pattern Recognition, Seiten 1399–1413, 1995.

[3] Caesar, T., J. Gloger und E. Mandler: *Preprocessing and Feature Extraction for a Handwriting Recognition System.* In: *Proceedings of the 2nd ICDAR*, Seiten 408–411, Tsukuba City, 1993.

[4] Carmo, Manfredo: *Differentialgeometrie von Kurven und Flächen.* Vieweg, Braunschweig, 1983.

[5] Cormen, Thomas H., Charles E. Rivest und Ronald L. Rivest: *Introduction to Algorithms.* MIT Press, 1990.

[6] Döker, Rolf, Till Maurer, Werner Kremer, K.-P. Neidig und Hans Robert Kalbitzer: *Determination of Mean and Standard Deviation of Dihedral Angles.* Biochem. Biophys. Res. Com., 257(2):348–350, April 1999.

[7] Duda, Richard O. und Peter E. Hart: *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[8] Forney, Jr. G. D.: *The Viterbi Algorithm.* Proc. IEEE, Seiten 268–278, März 1973.

[9] Guyon, I., P. Albrecht, Y. LeCun, J. S. Denker und Hubbard W.: *Design Of A Neural Network Character Recognizer For A Touch Terminal.* Pattern Recognition, 24(2):105–119, 1991.

[10] Guyon, Isabelle, Lambert Schomaker, Stan Janet, Mark Liberman und Rejan Plamondon: *First UNIPEN Benchmark of On-line Handwriting Recognizers Organized by NIST.* Technical Commitee 11 of the IAPR, 1994.

[11] (iUF), International UNIPEN Foundation: *The UNIPEN Project.* http://hwr.nici.kun.nl/unipen/.

[12] Jäger, Stefan: *Recovering Dynamic Information from Static, Handwritten Word Images.* Doktorarbeit, University of Freiburg, 1998.

[13] Merhav, N. und Y. Ephraim: *Hidden Markov Modeling Using a Dominant State Sequence with Application to Speech Recognition.* Computer Speech & Language, 5(4):327–339, 1991.

[14] Nartker, Thomas A.: *Benchmarking DIA Systems.* In: Bunke, H. und P.S.P. Wang (Herausgeber): *Handbook of Character Recognition and Document Image Analysis*, Seiten 801–820. World Scientific Publishing Company, 1997.

[15] Rabiner, L. und B. Juang: *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[16] Scattolin, Patrice: *Recognition of Handwritten Numerals using Elastic Matching.* Diplomarbeit, Concordia University Montréal, Québec, Canada, November 1995.

[17] Schukat-Talamazzini, Ernst Günter: *Automatische Spracherkennung—Grundlagen, statistische Modelle und effiziente Algorithmen.* Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, 1995.

[18] Schürmann, Jürgen: *Pattern Classification: a unified view of statistical and neural approaches.* John Wiley & Sons, Inc., New York, 1996.

[19] Tappert, C. C., C.Y. Suen und T. Wakahara: *The State of the Art in On-Line Handwriting Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(8):787–808, August 1990.

[20] Tappert, Charles: *Cursive Script Recognition by Elsatic Matching.* IBM J. Res. Develop., 26(6):765–771, November 1982.

[21] Theodoridis, S. und K. Koutroumbas: *Pattern Recognition.* Academix Press, 1999.

[22] Vuurpijl, Louis: *The HCLUS software environment a set of tools and dataset descriptions for performing character recognition.* http://hwr.nici.kun.nl/~vuurpijl/hclus/, Juni 1997.