AUSTRIAN JOURNAL OF STATISTICS
Volume 42 (2013), Number 2, 101–116

# Eine empirische Studie zur Verifikation von Unterschriften und zur Indikation von Fälschungen

Olga Wälder und Tobias Kutzner

Projektgruppe zu Blended Learning, Hochschule Lausitz (FH)

**Zusammenfassung:** Eine zufällige Stichprobe von Unterschriften, die auf mobilen Geräten mit Touch-Screen Display erzeugt wurde, soll hinsichtlich ihrer Verifikation untersucht werden. Zu diesem Zweck wurden verschiedene Klassifizierungsalgorithmen verwendet und miteinander verglichen. Ein weiterer Datensatz wurde erfolgreich auf Indikation der Fälschungen untersucht. Zudem wird in diesem Beitrag eine Transformation der ursprünglichen Variablen beschrieben, die zur Erhöhung der Qualität sowohl der Verifikation, als auch der Indikation der Fälschung beigetragen hat.

**Abstract:** A random sample of signs obtained by using touch screen displays is analyzed related to their verification. Different classification algorithms are applied and compared one with another for this aim. A second data set is successfully tested related to the indication of possible manipulations. Additionally, a special transformation of the original data sets is described, that leads to a significant improvement as well of the verification as of the indication of the manipulation.

**Schlüsselwörter:** Verifikation, Klassifikation, Hauptkomponentenanalyse, Clusteranalyse

## 1 Motivation

In den letzten Jahren gewinnt die Bearbeitung biometrischer Daten immer mehr an Bedeutung, vgl. Conde u.a. (2003), Santana u.a. (2010), Gehrke u.a. (2009), Ming-Yen u.a. (2005) und Sesa-Nogueras (2011). Insbesondere in privaten und öffentlichen Bereichen werden in den letzten Jahren vermehrt biometrische Verfahren zur Erhöhung der Sicherheit eingesetzt. Zum Beispiel erfordert der Zugriff auf Bank-Systeme die höchste Sicherheitsstufe.

In Kutzner (2012) wurde eine prototypische Lösung zur Erhöhung der Sicherheit der Zugangskontrolle zu Banksystemen unter Nutzung eines Mobil-Gerätes mit Touch-Screen entwickelt. Es werden Nutzername und Passwort eingegeben, das Passwort wird handschriftlich auf einem Touch-Screen erstellt. Das System prüft zusätzlich zu der üblichen PIN-Angabe die im handschriftlichen Passwort enthaltene biometrische Information. Im Anmelde-Modus wird der User Account angelegt. Dazu gibt der Nutzer zuerst seinen Nutzernamen auf dem Mobil-Gerät ein und anschließend mehrere Male sein handschriftliches Passwort für die Bildung des Klassifikationsmodells. Aus diesen Rohdaten werden 10 Variablen für jeden Nutzer extrahiert, die zur Verifizierung der Unterschriften verwendet werden, vgl. Kutzner (2012). Unter “Rohdaten” versteht man dabei ein digitales Abbild einer an sich stetigen Unterschrift. In der Informatiker-Sprache wird dieses

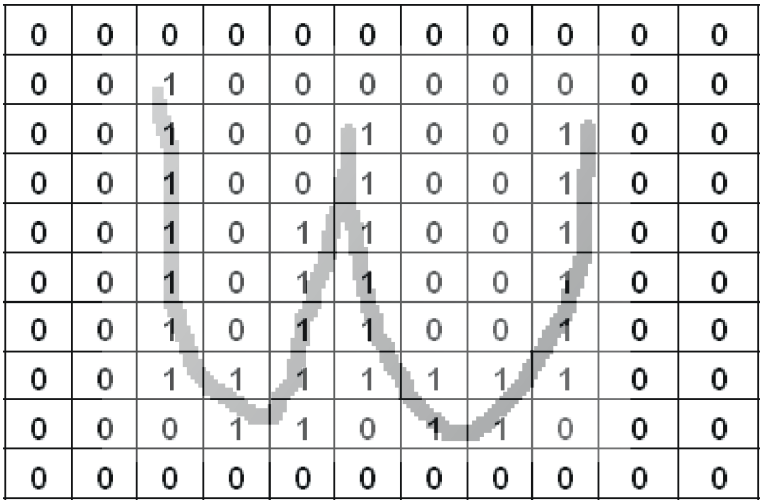

Abbildung 1: Der schematische Aufbau einer Displaymatrix. Die durchgezogene graue Linie stellt dabei ein Segment der handschriftlichen Signatur dar.

Abbild auch als “Displaymatrix” bezeichnet, die mit 0 und 1 belegt ist, vgl. Abbildung 1. Dabei steht 1 für eine belegte, d.h. von der Unterschrift markierten Zelle, und 0 für eine unbelegte Zelle.

Unter diesen 10 Variablen sind spezielle Winkel und solche geometrische Größen wie die maximale Höhe, Breite sowie der daraus abgeleitete Flächeninhalt des Segments der Unterschrift. Eine Beschreibung zur Herleitung dieser Variablen findet man in Wang u.a. (2011) und im Appendix.

In diesem Beitrag stellen wir die Ergebnisse der Analyse von zwei Datensätzen vor. Im ersten bzw. zweiten Datensatz wurden 96 bzw. 326 Unterschriften untersucht. Während der Analyse wurde eine simple Transformation der Daten durchgeführt, die im Grunde genommen der Standardisierung von Beobachtungen entspricht. Ohne Transformation ergab sich nach der Klassifikation mit dem Naive-Bayes-Ansatz für den ersten Datensatz die beste Erkennungsrate von aussagekräftigen 96.87 %. Dabei lagen die Ergebnisse für die KNN-Klassifikation ($k$-Nearest Neighbour) bei 90.62 %. Eine kurze Erläuterung zu dem Naive-Bayes-Ansatz und zu der KNN-Klassifikation findet man in Abschnitt 2.1.

Nach der Standardisierung konnte eine deutliche Erhöhung der Genauigkeit der Hauptkomponentenanalyse bei den untersuchten Datensätzen nachgewiesen werden. Für den oben erwähnten Datensatz verbesserte sich nach dieser Transformation die Erkennungsrate bei der KNN-Klassifikation von 90.62 % auf fast 99 %. Die Erkennungsrate mit dem Naive-Bayes-Ansatz ist ebenfalls angestiegen und erreichte 97.85 %.

Besonders drastisch war die Auswirkung der Transformation auf die Ergebnisse der KStar-Klassifikation (K*) beim zweiten Datensatz: Die Erkennungsrate hat sich beinahe verzehnfacht.

Beim zweiten Datensatz konnte nach dieser Transformation die Vermutung formuliert werden, dass die dort verwendeten Unterschriften manipuliert wurden. Einige Hinweise deuteten darauf hin, dass die als Rohdaten in die Berechnung eingegangenen Unterschriften nicht von zufällig ausgewählten ca. 100 Nutzern, sondern bis zu 90 % nur von einer kleinen Gruppe von Personen stammten.

Zudem wird der Möglichkeit der Variablenrestriktion nachgegangen. Dabei soll untersucht werden, inwieweit sich die Qualität der Klassifizierung mit vier statt zehn Variablen ändert. Die Variablenrestriktion kann die Geschwindigkeit des Extrahierungsalgorithmus von Variablen aus der Displaymatrix nur unwesentlich erhöhen.

# 2 Modellierung und Ergebnisse

## 2.1 Vorbearbeitung der Daten und ein kurzer Überblick über die angewendeten Verfahren

Statistische Verfahren wie Hauptkomponentenanalyse und Clusteranalyse haben sich beim breiten Anwendungskreis längst etabliert, vgl. Conde u.a. (2003). Während diese Methoden in zahlreichen Statistikbüchern wie z.B. in Mardia u.a. (1979) ausführlich behandelt werden, beschränken wir uns auf wesentliche Hauptziele beider Verfahren, die bei unseren Untersuchungen relevant waren.

Mit der Hilfe der *Clusteranalyse* wurden die aus den Rohdaten extrahierten zehn Variablen auf ihre "räumliche Nähe" analysiert. Falls man von der euklidischen Metrik in einem $N$-dimensionalen Raum ausgeht, betrachtet man die zehn Variablen zunächst als zehn eigenständige Cluster. Dabei bezeichnet $N$ die Anzahl der gesammelten Unterschriften. Falls wir uns diese Cluster als Kreise vorstellen wollen, bei denen die Radien gleichmäßig zu wachsen beginnen, so entstehen schrittweise sich überlappende Kreisgruppen, die nun als neue Zwischen-Cluster betrachtet werden. Nach der Analyse beider Datensätze konnte man nach einem bestimmten Schritt jeweils von vier etwa gleichentfernten Cluster ausgehen. Bei uns wurde das Single-Linkage-Verfahren für die hierarchische Clusteranalyse verwendet.

Mit der Hilfe der *Hauptkomponentenanalyse* wurden aus zehn Variablen ebenfalls vier Hauptkomponenten bestimmt. Dies bedeutete, dass die Information, die in zehndimensionalen Vektoren enthalten war, auf einen vierdimensionalen Raum reduziert wurde. Das angepasste Modell wurde anschließend auf seine Genauigkeit untersucht.

Um der üblichen Kritik an die Hauptkomponentenanalyse aus dem Wege zu gehen, wurde jeder der zehn Variablen der simplen Transformation (1) unterzogen, die der gewöhnlichen Standardisierung von Beobachtungen in der Statistik entspricht und als *z-Transformation* bezeichnet wird. Dadurch werden diese zehn Variablen sozusagen "dimensionslos' gemacht (es sollen keine Summen von "Äpfeln und Birnen" bei der Modellanpassung gebildet werden!)

$$z_i = \frac{x_i - \bar{x}_i}{s_i}\,, \qquad i = 1, \ldots, 10\,, \tag{1}$$

Hierbei wird mit $x_i$ die Originalvariable, mit $z_i$ die transformierte Variable und mit $\bar{x}_i$ und $s_i$ solche statistische Charakteristiken wie der Mittelwert und die Streuung der $i$-ten Variablen bezeichnet. Für jede der zehn Variablen stellt $x_i$ eine Stichprobe der Länge $N$ dar. Dabei bezeichnet $N$ die Anzahl der in den Rohdaten enthaltenen Unterschriften. Für den ersten Datensatz ist $N = 96$. Für den zweiten Datensatz entspricht $N = 326$. Bei der oben erwähnten Cluster- und Hauptkomponentenanalyse wurden alle Unterschriften aus den Datensätzen berücksichtigt.

Anhand der Untersuchung der Korrelationsmatrix für diese zehn Variablen konnte festgestellt werden, dass die Variable mit der Nummer 2 ("Anzahl der durch die Unterschrift belegten Pixel auf dem Touch-Screen") und die Variable mit der Nummer 10 ("für die Unterschrift verbrauchte Zeit") stark linear korrelieren (der Korrelationskoeffizient nach Pearson beträgt über 0.97), vgl. Tabelle 1. Nach der Transformation (1) wurde aus diesen Variablen ein arithmetischer Mittelwert gebildet. Die dabei entstandene neue Variable (die im nächsten Absatz schematisch mit $(2 + 10)/2$ bezeichnet wird) beschreibt somit den "Arbeitsaufwand des Nutzers, welchen seine Unterschrift beinhaltet".

Tabelle 1: Korrelationsmatrix der zehn Variablen: (a) erster, (b) zweiter Datensatz

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| (a) 1.00 | -0.35 | -0.36 | -0.20 | -0.60 | -0.61 | -0.56 | 0.33 | 0.16 | -0.28 |
| | 1.00 | -0.01 | -0.11 | 0.19 | 0.17 | 0.19 | -0.23 | -0.03 | 0.97 |
| | | 1.00 | -0.22 | -0.08 | 0.10 | -0.14 | 0.09 | 0.33 | 0.01 |
| | | | 1.00 | 0.63 | 0.38 | 0.60 | -0.38 | -0.22 | -0.13 |
| | | | | 1.00 | 0.75 | 0.93 | -0.69 | -0.08 | 0.08 |
| | | | | | 1.00 | 0.90 | -0.54 | 0.08 | -0.01 |
| | | | | | | 1.00 | -0.70 | -0.03 | 0.04 |
| | | | | | | | 1.00 | -0.32 | -0.16 |
| | | | | | | | | 1.00 | -0.03 |
| | | | | | | | | | 1.00 |
| (b) 1.00 | -0.34 | 0.15 | 0.06 | -0.34 | -0.25 | -0.32 | 0.28 | -0.10 | -0.34 |
| | 1.00 | -0.18 | -0.06 | 0.66 | 0.30 | 0.62 | -0.40 | 0.20 | 0.99 |
| | | 1.00 | -0.08 | -0.35 | 0.17 | -0.21 | 0.37 | -0.22 | -0.18 |
| | | | 1.00 | 0.14 | 0.25 | 0.17 | -0.22 | 0.27 | -0.06 |
| | | | | 1.00 | 0.47 | 0.94 | -0.64 | 0.31 | 0.66 |
| | | | | | 1.00 | 0.62 | -0.38 | 0.12 | 0.30 |
| | | | | | | 1.00 | -0.54 | 0.28 | 0.62 |
| | | | | | | | 1.00 | -0.44 | -0.40 |
| | | | | | | | | 1.00 | 0.20 |
| | | | | | | | | | 1.00 |

Nach analogem Prinzip wurden dann die folgenden vier Variablen $[(2+10)/2, 3, 6, 9]$ für eine Teilanalyse ausgewählt, die am schwächsten miteinander linear korrelierten. Das Wort "Teilanalyse" bezieht sich auf die statistische Untersuchung dieser vier Variablen anstatt der ursprünglichen zehn. Die Variable $(2 + 10)/2$ ist der oben erwähnte "Arbeitsaufwand des Nutzers, welchen seine Unterschrift beinhaltet". Die Variablen mit den Nummern drei und neun sind spezielle Winkel und die Variable sechs war die maximale Höhe der Unterschrift.

Nun sollen der Naive-Bayes-Ansatz, die KNN- sowie die K*-Klassifikation noch kurz erläutert werden. Unter anderem in Wikipedia findet man zahlreiche Beschreibungen der entsprechenden mathematischen Zusammenhänge, auf die wir hier verzichten möchten. Die algorithmische Implementierung dieser Verfahren findet man bei vielen Tools, die sich mit dem Problem "Data Mining" auseinandersetzen. Wir verwendetet das Tool WEKA 3.6, vgl. Hall u.a. (2009).

Ein *Bayes-Klassifikator* ist ein aus dem Satz von Bayes hergeleiteter Klassifikator. Er ordnet sozusagen jedes Objekt der Klasse zu, zu der es mit der größten Wahrscheinlichkeit gehört. Genau genommen handelt es sich um eine mathematische Funktion, die jedem Punkt eines Merkmalsraums eine Klasse zuordnet.

Es wird vorausgesetzt, dass die Wahrscheinlichkeit, dass ein Punkt des Merkmalsraums zu einer bestimmten Klasse gehört, bekannt ist. Das bedeutet, dass jede Klasse durch eine Wahrscheinlichkeitsdichte beschrieben wird. In der Realität sind diese Dichtefunktionen aber nicht bekannt. Man muss sie also approximieren. Dazu vermutet man hinter jeder Klasse einen Typ von Wahrscheinlichkeitsverteilung — in der Regel eine Normalverteilung — und versucht anhand der vorhandenen Daten, deren Parameter zu schätzen.

Der Naive-Bayes-Klassifikator ist aufgrund seiner schnellen Berechenbarkeit bei guter Erkennungsrate sehr beliebt. Mittels des naiven Bayes-Klassifikators ist es möglich, die Zugehörigkeit eines Objektes (Klassenattribut) zu einer Klasse zu bestimmen. Er basiert auf dem Satz von Bayes. Man nimmt dabei an, dass jedes Attribut nur vom Klassenattribut abhängt. Obwohl dies in der Realität selten zutrifft, erzielen naive Bayes-Klassifikatoren bei praktischen Anwendungen häufig gute Ergebnisse, allerdings nur solange die Attribute nicht zu stark miteinander korrelieren. Eine interessante praktische Anwendung dieses Verfahren zur Spam-Filterung von Emails findet man in Linke (2003).

Die *Nächste-Nachbarn-Klassifikation* ist eine parameterfreie Methode zur Schätzung von Wahrscheinlichkeitsdichtefunktionen. Der daraus resultierende $k$-Nearest-Neighbor-Algorithmus (KNN) ist ein Klassifikationsverfahren, bei dem eine Klassenzuordnung unter Berücksichtigung seiner nächsten Nachbarn vorgenommen wird. Der Teil des Lernens besteht aus einem simplen Abspeichern der Trainingsbeispiele, was auch als lazy learning ("träges Lernen") bezeichnet wird.

Die Klassifikation eines Objekts $x$ aus einem mehrdimensionalen Raum erfolgt im einfachsten Fall durch Mehrheitsentscheidung. An der Mehrheitsentscheidung beteiligen sich die $k$ nächsten bereits klassifizierten Nachbarn von $x$. Dabei sind viele Abstandsmaße denkbar (Euklidischer Abstand, Manhattan-Metrik, usw.). Das Objekt $x$ wird dann der Klasse zugewiesen, welche die größte Anzahl seiner $k$ Nachbarn enthält.

Für ein zu klein gewähltes $k$ besteht die Gefahr, dass Rauschen in den Trainingsdaten die Klassifikationsergebnisse verschlechtert. Für $k = 1$ ergibt sich ein sogenanntes Voronoi-Diagramm. Wird $k$ zu groß gewählt, besteht die Gefahr, Punkte mit großem Abstand zu $x$ in die Klassifikationsentscheidung mit einzubeziehen. Diese Gefahr ist insbesondere groß, wenn die Trainingsdaten nicht gleichverteilt vorliegen oder nur wenige Beispiele vorhanden sind. Bei nicht gleichmäßig verteilten Trainingsdaten kann eine gewichtete Abstandsfunktion verwendet werden, die näheren Punkten ein höheres Gewicht zuweist als weiter entfernten. Ein zusätzliches Problem ist auch der Speicher- und Rechenaufwand des Algorithmus bei hochdimensionalen Räumen und großen Trainingsdatensätzen. Weitere Details können Cost & Salzberg (1993) und Holte (1993) entnommen werden.

Im Beitrag von Cleary u.a. (1995) wurde ein Klassifizierungsalgorithmus vorgestellt, der auf einer so genannten Entropie-Abstandsfunktion basiert. Dieses Verfahren bezeichnet man als K*. Bei der Vergabe der Wahrscheinlichkeit für ein Objekt, einer Klasse zugeordnet zu sein, reagiert es empfindlich auf die Entfernung der Punkte sowie auf

das Vorhandensein bestimmter Strukturen im Datenmaterial. Das Verfahren stellt eine Weiterentwicklung von sogenannten "instance-based learning algorithms" dar, vgl. Aha u.a. (1991).

## 2.2 Ergebnisse der Cluster und der Hauptkomponentenanalyse

**1. Datensatz:** Es wurden 96 Unterschriften von zufällig ausgewählten Nutzern gesammelt. Dabei haben acht Personen jeweils 12-mal unterschrieben, acht Unterschriften von jedem Nutzer wurden zum Anlernen des Systems benutzt, die restlichen vier Unterschriften wurden zur Verifikation (Klassifikation) verwendet.

Unter "Anlernen des Systems" versteht man die folgende Vorgehensweise: Im Fall des Naive-Bayes-Klassifikators sollen diese acht Unterschriften zur Schätzung der Parameter der Normalverteilung verwendet, die dann der entsprechenden Klasse (der einzelne Nutzer, der Besitzer dieser acht Unterschriften) zugeordnet wird. Falls die KNN-Klassifikation verwendet wird, sind diese acht Unterschriften genau die acht Nachbarn des Objektes $x$. Als $x$ werden dann die restlichen vier Unterschriften verwendet.

Die Abbildung 2 zeigt den Box-Plot, die Ergebnisse der Hauptkomponentenanalyse für die nach (1) transformierten Daten. Dabei können die erste und zweite (latente) Hauptkomponente mit etwas Phantasie als "Arbeitsaufwand" und "Geometrie" der Unterschrift interpretiert werden.

Als Güte der Projektion bei der Hauptkomponentenanalyse wurde die mittlere quadratische Abweichung der theoretischen und empirischen Werte (MQA) ermittelt.

Ergebnisse: Die MQA betrug in diesem Fall 3.15; Die Verifikationsrate mit Naive-Bayes-Klassifikation betrug 97.85 %, jene mit KNN-Klassifikation 98.92 %, und jene mit K*-Klassifikation 100.00 %.

Im nächsten Schritt wurde der Datensatz auf vier Variablen reduziert, nämlich $[(2 + 10)/2, 3, 6, 9]$, die im Abschnitt 1 beschrieben wurden. Die Abbildung 3 präsentiert die Ergebnisse der Hauptfaktorenanalyse.

Ergebnisse: Die MQA betrug in diesem Fall 1.99; Die Verifikationsrate mit Naive-Bayes-Klassifikation betrug 94.62 %, jene mit KNN-Klassifikation 92.47 %, und jene mit K*-Klassifikation 81.25 %.

**2. Datensatz:** Hier wurden insgesamt 326 Unterschriften gesammelt. Diesmal haben die Personen jeweils viermal unterschrieben, drei Unterschriften von jedem Nutzer wurden zum Anlernen des Systems benutzt, die restliche Unterschrift wurde dann zur Verifikation (Klassifikation) verwendet.

Analog zum ersten Datensatz wurde auch hier zunächst die Transformation (1) der Rohdaten durchgeführt. Die Ergebnisse der nachfolgenden Hauptkomponentenanalyse sind der Abbildung 4 zu entnehmen.

Ergebnisse: Die MQA betrug in diesem Fall 3.16; Die Verifikationsrate mit Naive-Bayes-Klassifikation betrug 20.72 %, jene mit KNN-Klassifikation 47.75 %, und jene mit K*-Klassifikation 43.24 %.

Abbildung 5 zeigt die Punktwolke für zwei Variablen (Anzahl der Punkte vs. Höhe der Unterschrift) ohne und mit der Transformation (1) für den ersten und den zweiten Datensatz. Man kann aus der Gegenüberstellung sehen, dass die beiden Verteilungsmuster

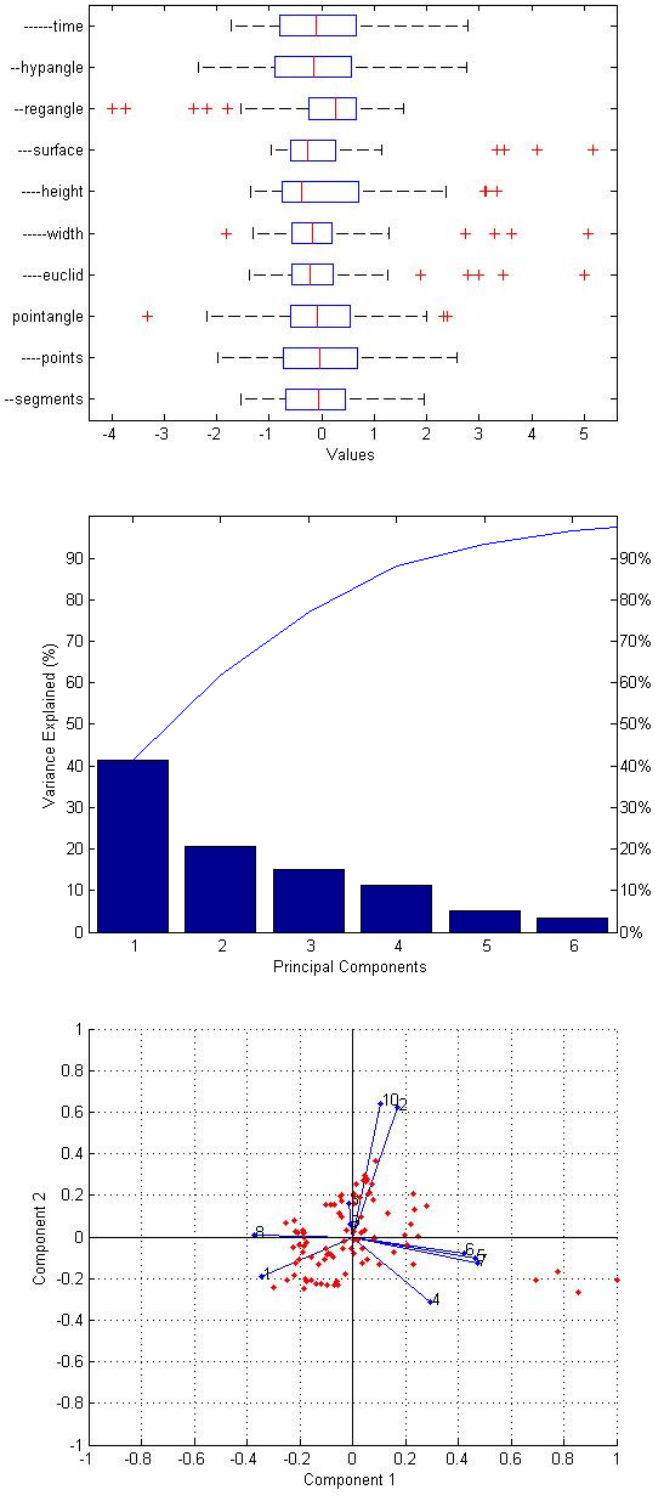


Abbildung 2: Box-Plot; die Varianzen der entsprechenden Hauptkomponenten in Prozent zu der Gesamtvarianz; Biplot für zwei Hauptkomponenten. Hier wurde der komplette Variablensatz von zehn Variablen für den ersten Datensatz verwendet.

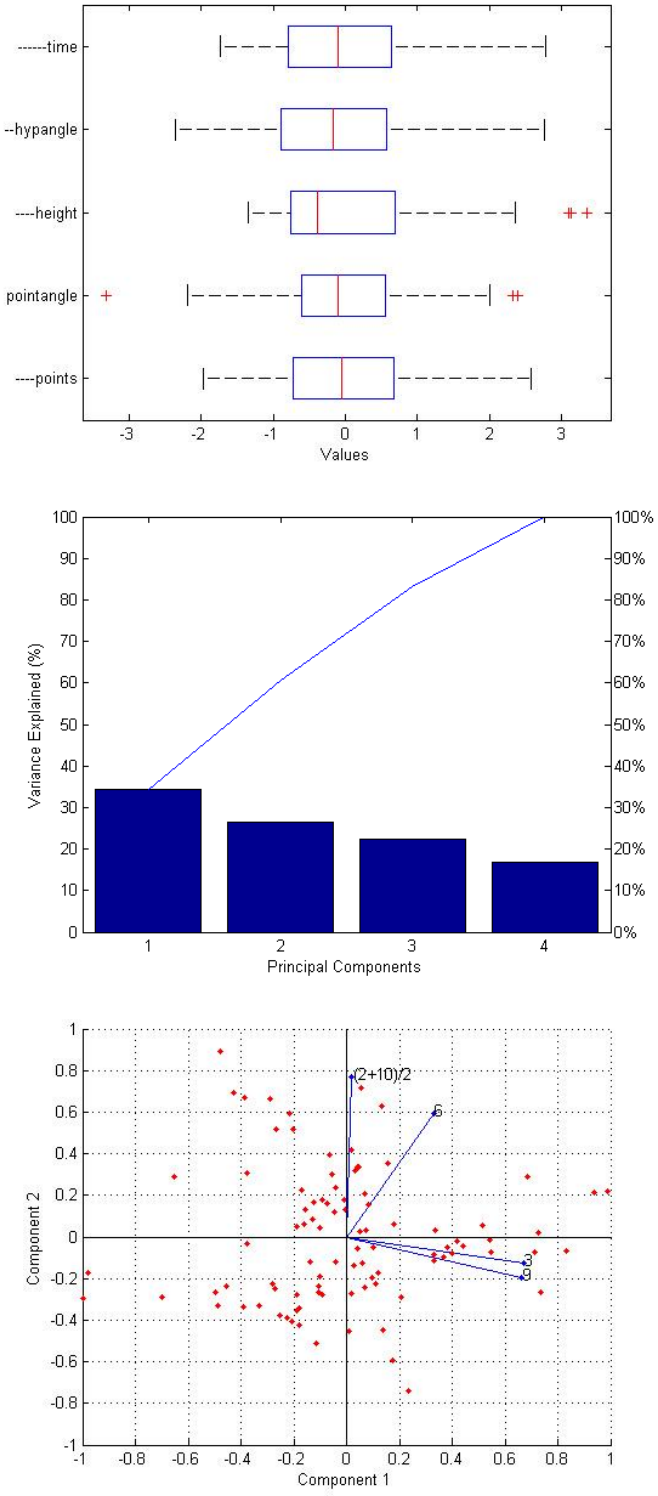


Abbildung 3: Box-Plot; die Varianzen der entsprechenden Hauptkomponenten in Prozent zu der Gesamtvarianz; Biplot für zwei Hauptkomponenten. Hier wurden nur vier Variablen von zehn verwendet.

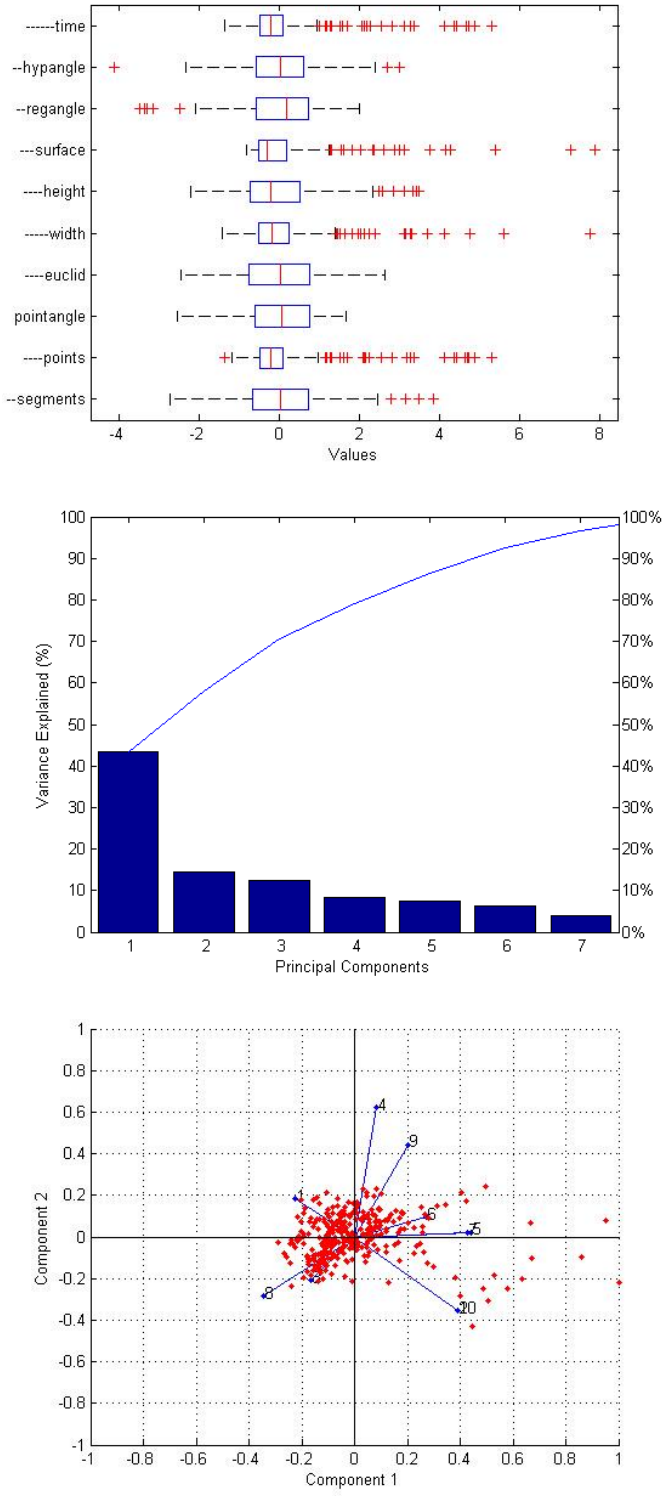


Abbildung 4: Box-Plot; die Varianzen der entsprechenden Hauptkomponenten in Prozent zu der Gesamtvarianz; Biplot für zwei Hauptkomponenten. Hier wurde der komplette Variablensatz von zehn Variablen für den zweiten Datensatz verwendet.

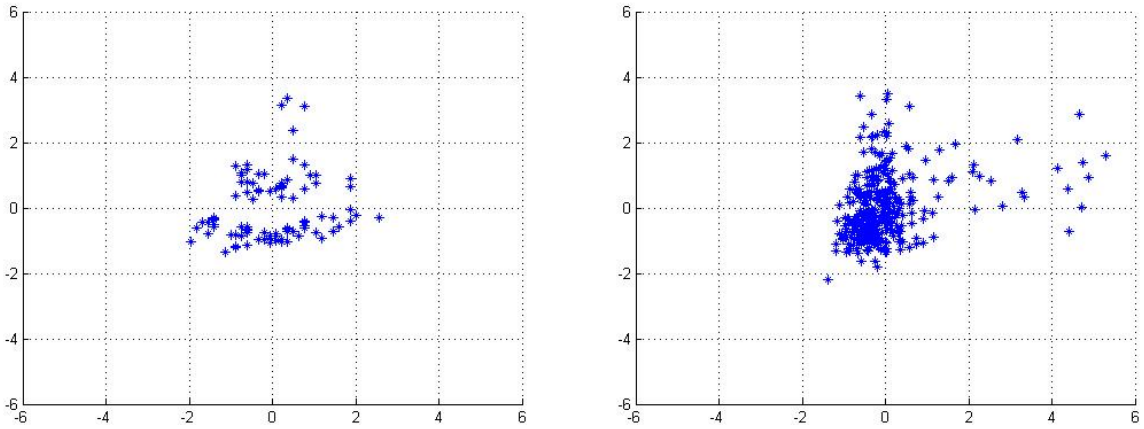


Abbildung 5: Die Punktwolken nach Transformation (1) für den ersten (links) und für den zweiten (rechts) Datensatz. Hierbei wurden die folgenden Variablen verwendet: Anzahl der Punkte ($y$-Achse) vs. Höhe der Unterschrift ($x$-Achse).

der Punkte nicht als ähnlich angesehen werden können. Es ergibt sich für den zweiten Datensatz einerseits eine signifikante Häufung der Punkte und anderseits einige deutliche Ausreißer in der rechten Hälfte, die vermuten lassen, dass die meisten Rohdaten höchstwahrscheinlich durch eine kleine Gruppe von Personen manipuliert wurden, die die unterschiedlichen Unterschriften simuliert haben. Dieses Muster kommt dabei bei vielen Variablenpaaren vor.

Abbildung 6 demonstriert das Ergebnis der Clusteranalyse für den ersten Datensatz, das mit den Ergebnissen der Hauptkomponentenanalyse in Abbildung 2 gut übereinstimmt (räumliche Nähe).

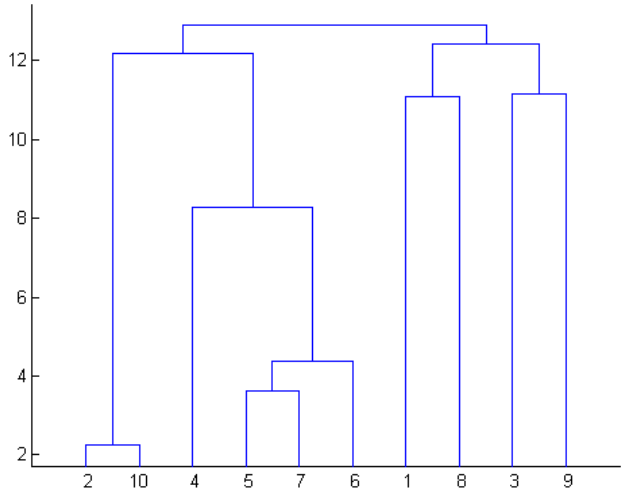


Abbildung 6: Ergebnisse der Clusteranalyse für den ersten Datensatz nach der Transformation (1).

Die bei der oben beschriebenen Analyse verwendeten Programme wurden mit Hilfe der Programmiersprache JAVA sowie der Softwaretools WEKA 3.6 und MATLAB 9.5 erstellt. Die grafische Visualisierung erfolgte dabei mit WEKA 3.6 und MATLAB 9.5.

In der Tabelle 2 werden die wichtigsten Kennwerte nochmals zusammengefasst. Die Reduktion auf vier Variablen verbesserte zwar die Genauigkeit der Projektion bei der Hauptkomponentenanalyse, verschlechterte allerdings die Erkennungsrate. Dies lag in erster Linie am Informationsverlust. Zusätzlich wirkten sich die Ausreißer in den Abbildungen 2 und 4 auf die Ergebnisse der Hauptkomponentenanalyse aus. Für den zweiten Datensatz ergab sich eine deutliche Verbesserung der Erkennungsrate nach der Transformation (fett hervorgehoben in der Tabelle 2).

Die Laufzeit spielt bei der Online-Schrifterkennung eine wesentliche Rolle. Zum Beispiel dauerte die Variablenextraktion für zehn Variablen und den ersten Datensatz nur 6.44 Millisekunden. Die Klassifikation mit dem Naive-Bayes-Ansatz erfolgte in 2.22 Millisekunden. Dagegen lag die Übertragungszeit der biometrischen Information vom Klienten bis zum Server im Intervall [500, 1500] Millisekunden. Allerdings hängen diese Angaben von der Leistungsfähigkeit der verwendeten Technik ab. Die Hauptrolle spielt dabei die Netzwerk-Übertragung.

Die Reduktion der Variablen von zehn auf vier bringt keine merklichen Unterschiede in der Laufzeit mit sich, da die Variablenextraktion stets weniger als 10 Millisekunden dauert. Mit der Entwicklung von modernen Mobile-Phones und Servern ist davon auszugehen, dass sich diese Laufzeit noch weiter reduzieren wird. Zudem kann eine zusätzliche Beschleunigung durch die geplante Optimierung des dafür entwickelten Algorithmus zur Variablenextraktion erreicht werden.

Tabelle 2: Gegenüberstellung der Erkennungsraten in Prozent ohne und mit Transformation (1) für beide Datensätze.

| | Erkennungsrate in Prozent | | | | | |
|---|---|---|---|---|---|---|
| | ohne Standardisierung (1) | | | mit Standardisierung (1) | | |
| Klassifikator | Naive-Bayes | KNN | K* | Naive-Bayes | KNN | K* |
| 1. Datensatz $N = 96$ | 96.87 | 90.62 | 90.63 | 97.85 | 98.20 | 100.0 |
| 2. Datensatz $N = 326$ | 19.82 | 47.75 | **4.50** | 20.72 | 47.75 | **43.24** |

Tabelle 3: Einfluss der Variablenreduktion auf die Genauigkeit der Projektion und auf die Erkennungsrate beim ersten Datensatz mit Transformation (1).

| Anzahl der Variablen | Genauigkeit | Erkennungsrate in Prozent | | |
|---|---|---|---|---|
| | der Projektion, MQA | Naive-Bayes | KNN | K* |
| 10 | 3.15 | 97.85 | 98.20 | 100.0 |
| 4 | 1.99 | 94.62 | 92.47 | 81.25 |

# 3 Ausblick und Diskussion

Mit der im Beitrag vorgestellten Studie wird die Möglichkeit der Schreiberidentifikation mit nur einem handschriftlichen Passwort auf einem Touch-Screen eines Mobil-Phones bestätigt. Das bedeutet, dass man diese Methodik zur Nutzer-Authentifizierung erfolgreich einsetzen kann.

Die besten Ergebnisse der Kombination von Online- und Offline-Identifikation mit gerundet 99 % wurden mit der KNN-Klassifikation sowie mit acht Passwörtern erreicht. Diese Methode wurde im aktuellen Prototyp des Systems integriert.

Die im Abschnitt 2.2 beschriebene Standardisierung (1) der Variablen führt unter anderem zur deutlichen Erhöhung der Genauigkeit der Projektion bei der Hauptkomponentenanalyse. Außerdem wird die grafische Präsentation der Punktwolke von Variablenpaaren so weit geschärft, dass die möglichen Manipulationen während der Erstellung von Rohdaten (zweiter Datensatz im Abschnitt 2.2) leicht aufgedeckt werden können.

Die Ergebnisse der Clusteranalyse stimmten mit den Ergebnissen der Hauptkomponentenanalyse überein. Allerdings ist der Einsatz beider Methoden auch etwas vorsichtig zu genießen: Die verwendeten Variablen weisen Ausreißer auf und sind nicht stochastisch unabhängig. Wie man beim zweiten Datensatz gesehen hat, verschlechtert die Manipulation des größten Teils des Datenmaterials durch eine kleine Gruppe von Nutzern die Erkennungsrate. Diese Vermutung konnte auch nach einem visuellen Abgleich mit den originalen biometrischen Daten bestätigt werden. Der K*–Klassifikator reagierte auf diese Manipulation besonders empfindlich: Die Erkennungsrate hat sich nach der Transformation (1) für den zweiten Datensatz beinahe verzehnfacht.

Selbstverständlich hat auch die Auflösung des Touch-Screens eines mobilen Gerätes einen Einfluss auf die Qualität der Klassifizierung/Verifizierung von den zu untersuchenden biometrischen Daten (handschriftliche Unterschriften in unserem Fall). Abbildung 1 zeigt den schematischen Aufbau einer Displaymatrix. Auch die Geschwindigkeit, mit der ein Nutzer seine Signatur auf dem Bildschirm hinterlässt, spielt eine Rolle. Falls diese Geschwindigkeit einen bestimmten Schwellenwert überschreitet, entstehen lückenhafte digitale Abbildungen. Diesbezügliche Untersuchungen stellen eine Schnittstelle zur digitalen Bildverarbeitung dar.

Eine Laufzeitreduktion durch Verringerung der Variablenanzahl ist nicht sinnvoll, da sie sowohl die Schreiberidentifikation als auch die Betrüger-Erkennung verschlechtert und dabei keine merklichen Verbesserungen in der Laufzeit mit sich bringt.

Um die Anzahl der Passworte für das Klassifikationsmodell auf ein für den Nutzer zumutbares Maß reduzieren zu können, soll im nächsten Schritt geprüft werden, ob weitere Variablen (bzw. ihre, sinnvoll gewichtete Kombination mit den aktuellen Variablen) zur Verifikation extrahiert werden müssen. Ebenfalls Gegenstand der weiteren Untersuchungen ist die Analyse des Einflusses der Größe der Stichprobe, der Anzahl der Unterschriften (sowohl zum Anlernen, als auch zur Klassifikation), der Auswahl des Klassifizierungsalgorithmus und der Optimierung des Algorithmus zur Variablenextraktion auf die Güte der Verifikation von Nutzern. In weiteren Studien soll zudem ein Modell mit Kreuzvalidierung mit der besten Vorhersagekraft und der Ermittlung der Erkennungsraten berücksichtigt werden.

## Danksagung

Die Aktivitäten der Projektgruppe zu Blended Learning der Hochschule Lausitz, die in diesem Artikel vorgestellt werden, werden vom Bundesministerium für Bildung und Forschung (BMBF) im Rahmen des Projekts "Anfangshürden erkennen und überwinden: Blended Learning zur Unterstützung der fachspezifischen Studienvorbereitung und des Lernerfolges im ersten Studienjahr" gefördert.

# Appendix: Beschreibung der verwendeten Variablen

Es werden folgende zehn Variablen verwendet und hier näher beschrieben:

- points
- segments
- time
- pointangle
- hypangle
- regangle
- width
- height
- surface
- euclid

Die ersten drei Variablen beziehen sich auf:

**points** Anzahl der für die Signatur belegten Pixel auf dem Display.

**segments** Anzahl der Segmente der Signatur. Ein neues Segment beginnt dabei immer dann, wenn der Schreiber das Display des Mobil-Phones berührt, um mit dem Schreibvorgang zu beginnen, und endet, wenn er die Hand (kapazitiven Stift) wieder vom Display abhebt, um an einer anderen Stelle weiter zu schreiben bzw. den Schreibvorgang zu beenden.

**time** Die Zeit, die der Schreiber für das Schreiben der Signatur benötigt.

Zudem werden die folgenden sieben geometrischen Variablen verwendet:

**pointangle** beschreibt den Winkel $\alpha$ der Geraden zwischen End- und Anfangspunkt des Segmentes in Bezug auf die Horizontale des Displays (Abbildung 7 links), wobei $(x_1, y_1)$ die Koordinaten des ersten Punktes des Segmentes sind und $(x_2, y_2)$ die Koordinaten des letzten Punktes des Segments sind,

$$\alpha = \arctan\left(\frac{|y_2 - y_1|}{|x_2 - x_1|}\right) .$$

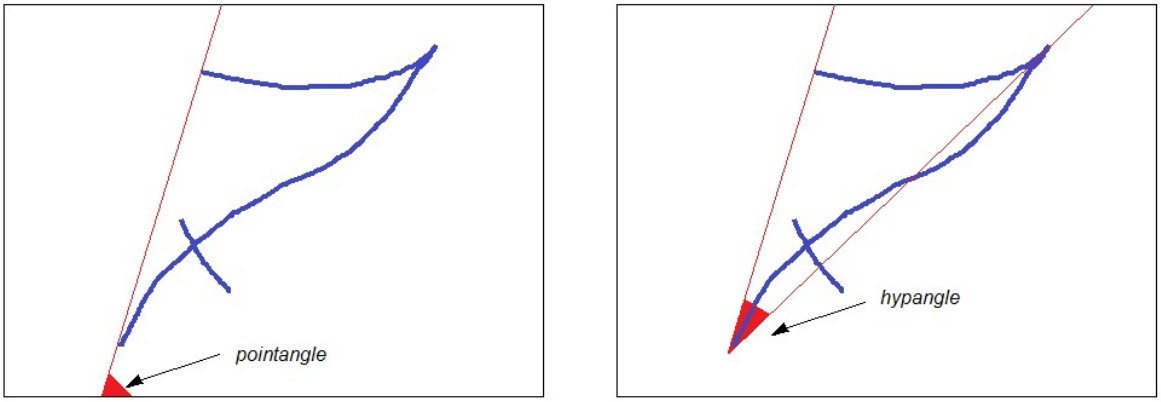


Abbildung 7: Zur Definition der Variablen “pointangle“ (links) und “hypangle” (rechts).

**hypangle** beschreibt den Winkel $\phi$ der Geraden zwischen End- und Anfangspunkt des Segmentes in Bezug auf die Hypotenuse des Segmentes (Abbildung 7 rechts),

$$\phi = \arccos\left(\frac{a_1 \cdot a_2}{|a_1| \cdot |a_2|}\right)$$

mit

$$a_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \qquad a_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix},$$

und

$$|a_1| = \sqrt{x_1^2 + y_1^2}\,, \qquad |a_2| = \sqrt{x_2^2 + y_2^2}\,.$$

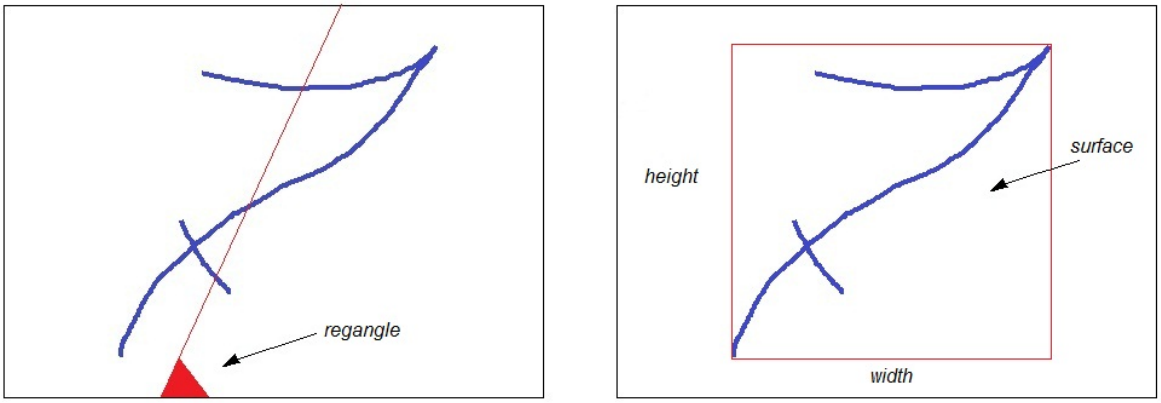


Abbildung 8: Zur Definition der Variablen “regangle“ (links) und der Variablen “width”, “height” und “surface” (rechts).

**regangle** beschreibt den Winkel der Geraden zwischen End- und Anfangspunkt des Segmentes in Bezug auf die Regressionsgerade des Segmentes (Abbildung 8 links), wobei $n$

die Anzahl der Punkte des Segmentes und $(x_i, y_i)$ der $i$te Punkt des Segmentes ist,

$$\alpha = \arctan\left(\frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}\right).$$

**width** und **height** beziehen sich auf die Extrempunkte des Segmentes und **surface** ist der daraus errechnete Flächeninhalt (Abbildung 8 rechts).

**euclid** beschreibt den Euklidischen Abstand zwischen den einzelnen Punkten des Segmentes. Falls die Signatur schneller geschrieben wird, vergrößert sich der Abstand zwischen den Punkten.

# Literatur


Aha, D. W., Kibler, D. and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*, 37-66.

Cleary, J. G. and Trigg, L. E. (1995). An instance-based learner using an entropic distance measure. In *12th International Conference on Machine Learning* (S. 108-114).

Conde, C., Ruiz, A. and Cabello, E. (2003). PCA vs low resolution images in face verification. In *Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'03).*

Cost, S. and Salzberg, S. (1993). A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, *10*, 57-78.

Gehrke, M., Steinke, K. and Dzido, R. (2009). Writer recognition by characters, words and sentences. *IEEE International Carnahan Conference on Security Technology*, 281-288.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The WEKA Data Mining Software: An Update [Software-Handbuch]. (Volume 11, Issue 1)

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*, 63-91.

Linke, A. (2003). Spam oder nicht Spam? E-Mail sortieren mit Bayes Filtern. *c't*, *17*, 150-153.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979). *Multivariate Analysis*. London, New York: Academic Press.

Ming-Yen, T. and Leu-Shing, L. (2005). Online writer identification using the point distribution model. *IEEE International Conference on Systems, Man and Cybernetics*, 1264-1268.

Santana, O., Travieso, C. M., Alonso, J. B. and Ferrer, M. A. (2010). Writer identification based on graphology techniques. *IEEE Aerospace and Electronic System Magazine*, 35-42.

Sesa-Nogueras, E. (2011). Discriminative power of online handwritten words for writer recognition. *IEEE International Carnahan Conference on Security Technology (ICCST)*, 1-8.

Wang, D., Zhu, B. and Nakagawa, M. (2011). A digitale ink recognition server for handwritten Japanese text. *IEEE International Conference on Document Analysis and Recognition*, 146-150.

Authors' address:
Olga Wälder und Tobias Kutzner
E-Learning Team
Hochschule Lausitz
Großenhainer Str. 57
D-01968 Senftenberg
Deutschland

E-Mail: olga.waelder@hs-lausitz.de, tobias.kutzner@hs-lausitz.de
WebPage: http://www.hs-lausitz.de/studium/elearning.html