

Programmentwurf Wissensbasierte Systeme

Prof. Dr. Dirk M. Reichardt

**Definition eines Ähnlichkeitsmaßes zur Klassifizierung von Handzeichen**

Studiengang Angewandte Informatik

Duale Hochschule Baden-Württemberg, Stuttgart

von

**Marco Klein** 9513693

**Matthias Bidlingmeyer** 1613581

Inhaltsverzeichnis

[Dokumentation 1](#_Toc471239496)

[Aufgabenstellung 1](#_Toc471239497)

[Entwurf 1](#_Toc471239498)

[Ergebnis 1](#_Toc471239499)

[Umsetzung 1](#_Toc471239500)

[Bedienung 2](#_Toc471239501)

# Dokumentation

## Aufgabenstellung

Definieren und implementieren Sie ein Ähnlichkeitsmaß, mit dem es möglich ist, die drei Handsymbole „Schere“, „Stein“ und „Papier“ des bekannten Spiels zu klassifizieren. Eingabe sind die Rohdaten des LEAP Motion Sensors (nicht die vordefinierten Handzeichen).

## Entwurf

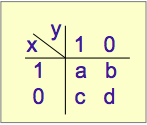
Das Handsymbol Schere wird durch ausgestreckten Mittelfinger und ausgestreckten Zeigefinger definiert. Papier wiederum ist gegeben, wenn alle Finger ausgestreckt sind und Stein, wenn kein Finger ausgestreckt ist. Daher genügt es zu testen, welche Finger ausgestreckt sind und diese mit den unterschiedlichen Handsymbolen zu vergleichen.

Die LEAP Motion API besitzt bereits eine solche Funktion, die wir daher nutzen werden. Diese Funktion besitzt die zwei Zustände „ausgestreckt“ oder „nicht ausgestreckt“. Daher können keine Übergänge, wie zum Beispiel „halb ausgestreckt“ erkannt werden. Daher arbeiten wir mit 5 booleschen Werten, wobei „wahr“ für ausgestreckt und „falsch“ für nicht ausgestreckt steht. Der erste Finger ist dabei der Daumen und der letzte der kleine Finger.

Die Konstellation an ausgestreckten Fingern soll als Zustand bezeichnet werden. Der Zustand einer Schere wäre also der boolesche Vektor „011000“. Dabei beschreibt eine „1“ einen ausgestreckten Finger und eine „0“ einen nicht ausgestreckten Finger. In diesem Fall wären also der Zeigefinger und der Mittelfinger ausgestreckt. Für einen Stein ergibt sich demensprechend „00000“ und für ein Papier „11111“.

Diese Zustände ergeben unsere Wissensbasis. Durch Versuche mit LEAP Motion konnten wir bereits feststellen, dass das System nicht immer zuverlässig arbeitet und wir teilweise falsche Handdaten ausgegeben wurden. Um Fehler tolerieren zu können ist ein Ähnlichkeitsmaß sinnvoll.

Da wir all unsere Zustände in booleschen Vektoren beschreiben können, bietet sich die formale Festlegung von 4 Beschreibungsgrößen an. In unserem Fall beschreibt der Vektor x die Wissensbasis und der Vektor y das gemessene Ergebnis von LEAP Motion. Diese beiden Vektoren sollen nun gegenübergestellt und verglichen werden, wie die folgende Matrix zeigt:



Die vier Beschreibungsgrößen lassen sich unmittelbar aus der Matrix ableiten:

a = Sum(xi,yi)

b = Sum(xi, ¬yi)

c = Sum(¬xi,yi)

d = Sum(¬xi, ¬yi).

Die Beschreibungsgröße a gibt dabei alle Übereinstimmungen an und d alle nicht Übereinstimmungen.

Zu aller erst wollen wir unser Ähnlichkeitsmaß über den Simple Matching Coefficient (SMC) definieren. Dieser vergleicht alle 4 Beschreibungsgrößen und liefert einen Ähnlichkeitswert zurück. Stimmen beide Vektoren überein, wir der Wert 1 geliefert. Stimmen keine Werte überein, wir der Wert 0 geliefert. Eine optimistische bzw. pessimistische Auslegung macht hier keinen Sinn, da wir nur 5 Vergleichswerte haben. Eine veränderte Auslegung würde das Ergebnis dadurch entweder weit nach unten oder weit nach oben treiben und uns wenig nutzen.

Als weiteres Ähnlichkeitsmaß ließe sich der Jaccard S-Koeffizient heranziehen. Dieser Betrachtet nur die positiven Übereinstimmungen. Für unseren Fall ist dieser jedoch komplett ungeeignet, da dieser Koeffizient den Zustand Stein überhaupt nicht erkennen könnte, da Stein nur aus negativen Übereinstimmungen besteht.

## Ergebnis

Vergleiche wurden nur mit dem Simple Matching Coefficient durchegeführt und im Folgenden sollen einige exemplarische Ergebnisse aufgeführt werden. Um komplett sinnfreie Ergebnisse zu vermeiden haben wir uns auf den Schwellenwert von 0,6 festgelegt. Werte darüber werden als gültiges Ergebnis betrachtet und Ähnlichkeitswerte darunter als nicht ähnlich genug.

Als erstes betrachten wir das Ergebnis für den Vektor „00000“, sprich den Faust-Zustand:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d |  | SMC |
| Schere | 0 | 0 | 2 | 3 |  | 0,6 |
| Stein | 0 | 0 | 0 | 5 |  | 1 |
| Papier | 0 | 0 | 5 | 0 |  | 0 |

Ergebnis für Vergleich mit Zustand „00000“.

Wie erwartet haben wir eine 100% Übereinstimmung bei Stein. Bei Schere sind es bereits 60%. In diesem Fall würden wir uns also selbstverständlich für Stein entscheiden.

Betrachten wir nun den Zustand „00110“. Hier wären Mittelfinger und Ringfinger gestreckt:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d |  | SMC |
| Schere | 1 | 1 | 1 | 2 |  | 0,6 |
| Stein | 0 | 2 | 0 | 3 |  | 0,6 |
| Papier | 2 | 0 | 3 | 0 |  | 0,4 |

Ergbenis für den Vergleich mit Zustand „00110“.

Auch wenn der Wert für Schere und Stein der Selbe ist, kommen wir hier zu einem nicht definierten Ergebnis, da der Schwellenwert 0,6 beträgt. Wir hätten diesen Zustand aber auch als eine „fehlerhafte“ Schere auslegen können, sprich wenn der Nutzer seine Schere mit dem Mittelfinger und dem Ringfinger machen würde. Dies ist aber eher unwahrscheinlich, daher ist der Schwellenwert von 0,6 gut gewählt.

Interessanter wird es bei dem Zustand „11100“. Hier sind sowohl Daumen, als auch Zeige- und Mittelfinger gestreckt.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d |  | SMC |
| Schere | 2 | 1 | 0 | 2 |  | 0,8 |
| Stein | 0 | 3 | 0 | 2 |  | 0,4 |
| Papier | 3 | 0 | 2 | 0 |  | 0,6 |

Ergbenis für den Vergleich mit Zustand „11100“.

Hier würden wir auf das Ergebnis Schere kommen, was durchaus berechtigt ist, da der Daumen oft fehlerhaft von LEAP Motion erkannt wird.

## Umsetzung

Das Programm ist in Java implementiert und funktioniert nach dem typischen EVA-Prinzip (Eingabe, Verarbeitung, Ausgabe).

Als Eingabe wird die Datei „input.csv“ aus dem Stammverzeichnis der Applikation gelesen. Dabei werden die einzelnen Datensätze in eine Liste geladen. Ein Datensatz besteht aus fünf boole’schen Werten für den Zusand des jeweiligen Fingers, angefangen vom Daumen bis zum kleinen Finger. Der Wert 1 steht für einen ausgestreckten, die 0 für einen angewinkelten Finger.

Für jeden Datensatz wird nun der Simple Matching Coefficient (SMC) im Bezug zu den drei Handzeichen Schere, Stein und Papier berechnet. Die so genannte Knowledge Base repräsentiert diese Handzeichen in der Form von drei „perfekten“ Datensätzen. Zu diesem Ideal werden nun die vier verschiedenen Beschreibungsgrößen (Anzahl positiver Übereinstimmungen , negativer Übereinstimmungen etc.) ermittelt. Daraus ergibt sich dann der SMC, also die Ähnlichkeit zu dem Handzeichen.

Abschließend wird eine Ausgabe-CSV Datei mit den Datensätzen, den drei Ähnlichkeitswerten und dem jeweiligen „Gewinner“ erstellt. Der Gewinner, das ähnlichste Handzeichen, ist das mit dem höchsten SMC größer als 0,6.

## Bedienung

Zum Ausführen des Programms muss eines der drei beiliegenden input-Files als „input.csv“ im selben Verzeichnis wie die „similarity.jar“ gespeichert werden. Das Ausführen der similarity.jar generiert eine neue CSV-Datei mit den berechneten Ähnlichkeitswerten.