

Hausarbeit SME-PHY-B: Wahlthema 2 - Fitnesszähler

Joel Ewig

20. März 2021

Zusammenfassung

Sensorbasiert sollen ähnlich einer Fitnessuhr verschiedene Sportübungen erkannt sowie die Anzahl wie oft diese jeweils ausgeführt wurden. Benutzt werden sollen der Beschleunigungssensor und das Gyroskop des IMU-6050. Mein Lösungsansatz gliedert sich in mehrere Stufen: die Rohdaten aus dem IMU-6050 sollten durch einen mehrdimensionalen Kalman-Filter bereinigt werden. Die bereinigten Daten werden mit dem K-Means Algorithmus geclustert um daraus sogenannte „Beobachtungen“ zu machen. In der Lernphase wird mit diesen Beobachtungen pro Übung ein Hidden Markov Model erlernt. In der folgenden Detektionsphase werden die empfangenen Daten ebenfalls geclustert und die Beobachtungen an alle Hidden Markov Modelle weitergegeben. Das HMM welches die höchste Wahrscheinlichkeit für die Emittierung der beobachteten Folge bestimmt, gilt als Erkennen und die zugehörige Übung wird als ausgeführt behandelt.

Inhaltsverzeichnis

1	Konzept	4
2	Sensorik	4
3	Mehrdimensionaler Kalman-Filter	4
4	Lernphase	4
5	Detektionsphase	5
6	Baum-Welch Algorithmus	5
7	Evaluation	6

1 Konzept

1. Accelerometer und Gyrometer mittels mehrdimensionalen Kalman-Filter bereinigt
2. gefilterte Daten werden mittels kmeans geclustert
3. clusterzugehörigkeit als beobachtung in ein HMM zur Erkennung von Aktivität und Zählen

2 Sensorik

Auf dem Arduino werden die Daten des Beschleunigungssensors und des Gyroskops aus dem IMU ausgelesen. Das Gyroskop sowie der Beschleunigungssensor werden dem im Datenblatt beschriebenen Selbsttest unterzogen. Dieser soll feststellen, dass die Sensoren noch funktionsfähig sind. Die zugehörigen Daten werden über die serielle Schnittstelle an ein Pythonskript übermittelt, welches die zugehörigen Rechnungen vornimmt und bei Nichtbestehen des Selbsttests das Programm beendet.

3 Mehrdimensionaler Kalman-Filter

Der mehrdimensionale Kalman-Filter soll hier vernachlässigt werden, da er nicht unbedingt nötig ist für die Erkennung. Mit einem Kalman-Filter wäre die Erkennung vermutlich robuster, da ohne ihn entsprechende Ausreißer bei relativ kleinen k s zu unsichereren Clustern führt. Falls mit einem großen k gearbeitet wird, bekommt ein solcher Ausreißer ein eigenes Cluster und das HMM muss ihn berücksichtigen können, was bei so einem kleinen Datensatz wie hier unvorteilhaft ist. Da dies allerdings Ausnahmefälle sind, können wir diese hier vernachlässigen.

4 Lernphase

Die einkommenden 6-dimensionalen Daten werden mit einem KMeans geclustert. Experimentell wurde ermittelt, dass stabile Ergebnisse für alle $k \geq 3$ für einfache und wenige Übungen liefert. Ab $k \geq 5$ auch mit mehr beziehungsweise voneinander unterschiedlicheren Übungen, weshalb $k = 5$ als Default-Wert genutzt wird. Für sehr komplexe Übungen sollte k erhöht werden. Der kmeans wird auf den kompletten Trainingsdaten, also unabhängig von den Übungen, initialisiert. Jeder Punkt bekommt dadurch einen Clusternamen zugewiesen, hier: Nummern, welche weiterverarbeitet werden. Die Clusternamen sind die Beobachtungen die als Eingabe für das Lernen der HMMs dienen.

Ein HMM für eine Übung wird wie folgt trainiert: Zuerst werden die Emissionswahrscheinlichkeiten und die Übergänge zwischen den versteckten Zuständen

zufällig initialisiert. Anschließend werden die genannten Werte mittels des Baum-Welch Algorithmus an die in der Lernphase aufgenommenen Sequenzen angepasst, mehr dazu in Kapitel 6. Die HMMs werden mehrfach initialisiert um zu die Wahrscheinlichkeit zu vermindern, dass man nicht über ein lokales Optimum hinauskommt. Es wird das HMM gewählt, welche die größte Sicherheit bei der Erkennung der bekannten Übungen aufweist. Es wird nicht in ein Trainings- und Testdatensatz unterteilt, da der Datensatz so klein wie möglich bleiben soll.

5 Detektionsphase

In der Detektionsphase werden die empfangenen Daten mit dem K-Means aus der Lernphase geclustert. Der K-Means ist eingefroren, das heißt die Clustermittelpunkte bleiben wie sie in der Lernphase festgestellt wurden.

Die Beobachtungen einer Durchführung werden allen HMMs zugeführt, diese geben die Emissionswahrscheinlichkeit für die beobachtete Sequenz aus. Es wird die Übung erkannt, dessen zugehöriges HMM die höchste Wahrscheinlichkeit für die Emission der beobachteten Sequenz ausgibt.

6 Baum-Welch Algorithmus

Der Baum-Welch Algorithmus wendet den Maximum-Likelihood Algorithmus auf Hidden Markov Modelle an. Genutzt wird ein Hidden Markov Model (π, p, q) mit:

1. $\pi : Z \rightarrow [0, 1]$ als initiale Wahrscheinlichkeiten für die Zustände Z
2. $p : Z \times Z \rightarrow [0, 1]$ als Übergangswahrscheinlichkeiten zwischen den Zuständen Z
3. $q : Z \times B \rightarrow [0, 1]$ als Beobachtungswahrscheinlichkeiten für die Beobachtung B in den Zuständen Z

In der hier genutzten Implementierung sind p und q als Matrizen umgesetzt. Diese werden zu Beginn mit zufälligen Werten initialisiert. Im ersten Schritt werden die Wahrscheinlichkeiten mit den aktuellen Parametern ermittelt. Die Idee hinter dem Baum-Welch Algorithmus ist, die Übergänge, die der wahrscheinlichste Pfad durch das HMM nutzt, wahrscheinlicher zu machen.

Für eine beobachtete Sequenz $O = o_1, \dots, o_T$ heißt das, dass π_i in der nächsten Iteration zu $\tilde{\pi}$ und berechnet sich durch:

$$\tilde{\pi}_i = \gamma_i(1)$$

$\gamma_i(t)$ ist die erwartete Häufigkeit, sich in Zustand i zum Zeitpunkt t aufzuhalten. Diese ergibt sich aus:

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^N \alpha_j(t)\beta_j(t)}$$

α und β können rekursiv ermittelt werden über:

$$\begin{aligned}\alpha_i(1) &= \pi_i q_i(o_1) \\ \alpha_j(t+1) &= \sum_{i=1}^N \alpha_i(t) p(i, j) q_j(o_{t+1}) \\ \beta_i(T) &= 1 \\ \beta_i(t) &= \sum_{j=1}^N p(i, j) q_j(o_{t+1}) \beta_j(t+1)\end{aligned}$$

Dann werden p und q angepasst mit:

$$p(i, j) = \frac{\text{Anzahl der Übergänge von } i \text{ nach } j}{\text{Anzahl der Übergänge von } i \text{ zu einem beliebigen Zustand}}$$

$$q_i(k) = \frac{\text{Anzahl der Zeitschritte in } i \text{ in denen } k \text{ beobachtet wurde}}{\text{Anzahl der Zeitschritte in } i}$$

7 Evaluation

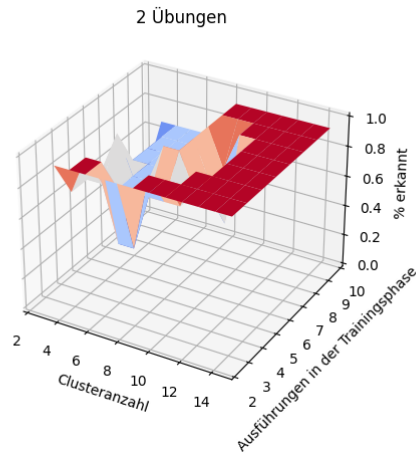


Abbildung 1: Bildunterschrift

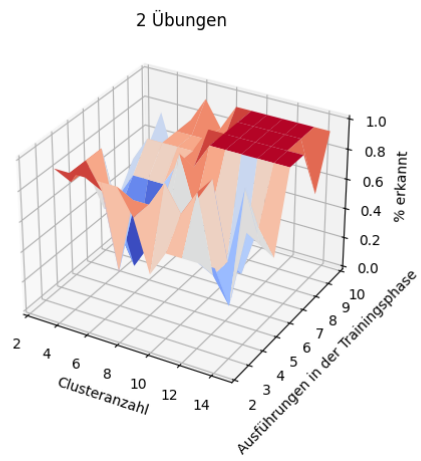


Abbildung 2: Bildunterschrift

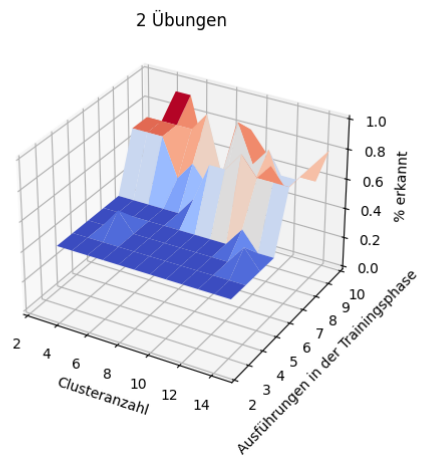


Abbildung 3: Bildunterschrift

2 Übungen

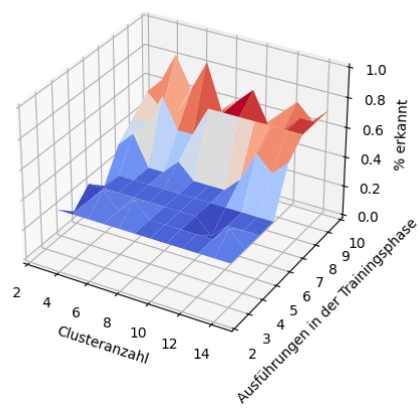


Abbildung 4: Bildunterschrift