Maschinelles Lernen
Master Aufgabe - Arrhytmia

Steven Brandt, *Otto-von-Guericke-Universität Magdeburg*
Fabian Witt, *Otto-von-Guericke-Universität Magdeburg*

1 Motivation

Einer der häufigsten Todesursachen, beim Transport von Patienten ins Krankenhaus, sind nicht erkannte oder falsch interpretierte Herzrhythmusstörungen. Um die Arbeit des Rettungsdienstpersonals zu erleichtern und die frühzeitige Erkennung von Herzrhythmusstörungen zu unterstützen, soll mit Hilfe von Algorithmen des maschinellen Lernens versucht werden, EKG-Daten zu interpretieren und zu klassifizieren.

2 Daten

Als Grundlage dient der Arrhythmia-Datensatz des UCI Machine Learning Repository's[1]. Dieser enthält 450 Datensätze mit jeweils 279 Attributen und einer Klasse. Attribute sind:

- Alter
- Geschlecht
- Größe
- Gewicht
- verschiedene EKG-Daten

Dabei wird unter 16 verschiedenen Klassen unterschieden (eine Klasse für normale Herzfunktionen und 15 Klassen für verschiedene Herzrhythmusstörungen).

1. http://archive.ics.uci.edu/ml/

3 Vorverarbeitung

Im ersten Schritt werden alle Probanden entfernt, die jünger als 19 Jahre sind (der BMI ist nur gültig für Personen ab 19 Jahren). Anschließend wird der Body-Mass-Index (BMI) errechnet und in nominale Gruppen unterteilt:

- kleiner als 20
- 20 bis 25
- 26 bis 30
- 31 bis 40
- größer als 40

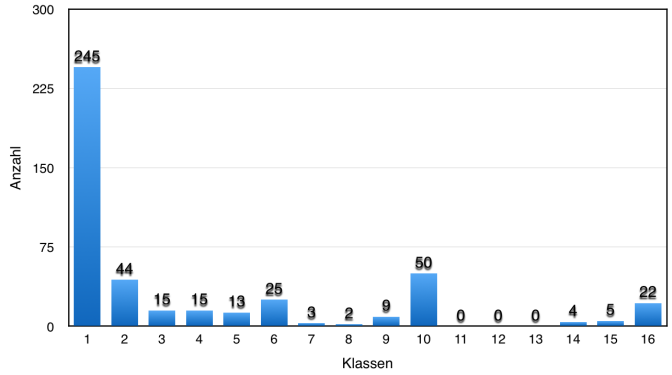
Das gleiche gilt für das Alter, welches ebenfalls in nominale Gruppen unterteilt wird:

- 19 bis 29
- 30 bis 49
- 50 bis 69
- größer als 69

Durch die ungleichmäßige Verteilung der Daten auf die 16 Klassen (siehe Fig. 1) entscheiden wir uns gegen die genaue Klassifizierung besonderer Herzrhythmusstörungen (Klassen 2 bis 16) und vereinfachen es zu einem zwei Klassen Problem (normale Herzfunktion und Herzrhythmusstörung).

Fehlende Werte, bei einzelnen Attributen, veranlassen uns diese mit Hilfe des K-Nearest-Neighbor-Algorithmus (k gleich 8) zu ersetzen. Ein Attribut entfernen wir auf Grund von 83,19 Prozent fehlender Werte.

Fig. 1. Verteilung der Klassen



4 Lernverfahren

Als Lernverfahren wählen wir den K-Nearest-Neighbor-Algorithmus und ein Neural Network.

4.1 K-Nearest-Neighbor

Der K-Nearest-Neighbor-Algorithmus klassifiziert einen neuen Datensatz anhand der k nächsten Nachbarn. Hierbei wählen wir k im Bereich von 1 bis 10.

4.2 Neural Network

Beim Neural Network hingegen wird anhand der Trainigsdaten die Gewichtung der Neuronen bestimmt. Variabel ist hier die Anzahl der Hidden-Layer, mit denen die Komplexität des Neural Network und somit auch die der Trennfunktion bestimmt wird. Dabei wird der Bereich von 1 bis 10 Hidden-Layer betrachtet.

5 Auswertung

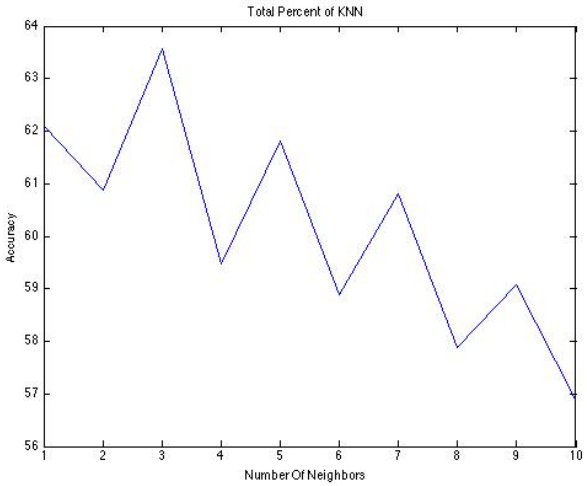
In der Auswertung wird auf eine abgeänderte Version des Kreuzvalidierungsverfahrens zurückgegriffen. Die Datenmenge wird dabei in jedem Durchlauf in Trainings- und Testdaten aufgeteilt. Die Aufteilung geschieht dabei immer zufällig in einem Verhältnis von 70 Prozent Trainingsdaten zu 30 Prozent Testdaten. Nach einer vorher festgelegten Anzahl an Durchläufen wird die Gesamtfehlerquote als Durchschnitt aus den Einzelfehlerquoten errechnet.

Um den jeweiligen Fehler des Algorithmus zu bestimmen, werden für den K-Nearest-Neighbor-Algorithmus 100 und für das Neural Network 50 Durchläufe betrachtet. Auf Grund des hohen Trainingsaufwandes musste die Anzahl der Zyklen beim Neural Network verringert werden.

6 Ergebnisse

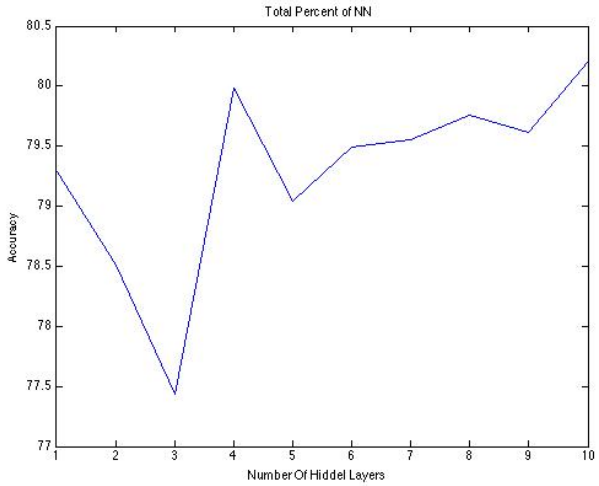
Beim K-Nearest-Neighbor-Algorithmus erreichten wir maximal 63,70 Prozent mit 3 Nachbaren. Ab 4 Nachbaren verschlechterte sich das Ergebnis zunehmends. Dies liegt an den stark verrauschten Daten und der geringen Beispielanzahl. Die Ergebnisse des K-Nearest-Neighbor-Algorithmus sind in Figure 2 zu sehen.

Fig. 2. Result: K-Nearest-Neighbor



Das Neural Network verzeichnet bessere Ergebnisse aufgrund komplexerer Trennfunktionen und erzielt somit eine Genauigkeit von 80,30 Prozent (bei 10 Hidden-Layern). Die schlechtere Klassifizierung bei weniger Hidden-Layern deutet auf eine zu simple Trennfunktion hin (siehe fig. 3).

Fig. 3. Result: Neural Network



7 FAZIT

Es hat sich gezeigt, dass weder die Qualität noch die Quantität der gegebenen Daten für eine präzise Klassifizierung ausreichend sind. Durch die Reduktion auf 2 Klassen konnten trotzdem gute Ergebnisse erzielt werden. Weiterhin wäre denkbar, eine Support-Vector-Machine auf das Problem anzusetzen und die Resultate zu vergleichen. Möglicherweise lassen sich somit noch bessere Ergebnisse erzielen.