

Erste Auswertung

Auswertungen

Für eine erste Auswertung gab es zwei Ansätze:

1. Ist es möglich anhand der Daten zwischen Target und Distraktoren zu unterscheiden?
Hiermit sollte untersucht werden, ob es möglich ist, mit Hilfe eines Klassifikators eine Zuordnung der Daten zu einem Target oder einer Distraktor-Klasse zu finden. Dabei wurde zuerst ein Target und eine Distraktoren-Klasse trainiert und jeweils mit einer neuen Distraktor-Klasse erweitert und das Ergebnis verfolgt.
2. Gibt es Unterschiede in den Daten bei einem Objekt, wenn es einmal die Rolle als Target und als Distraktor hatte?
Hiermit sollte untersucht werden, ob man die unterschiedliche Suchstrategie in den Daten erkennen kann. Zum einen wenn dieses Objekt gesucht wird, und zum anderen wenn dieses Objekt als Distraktor fungiert. Dabei wurden die Daten von einem Durchlauf in dem dieses Objekt Target war gegen alle anderen Durchläufe in denen es Distraktor war klassifiziert.

Datenset

Diese Auswertung wurde erstmals auf Basis einer Versuchsperson durchgeführt. Dabei gab es 5 Durchgänge mit jeweils einem der 5 Objekte als Target. Insgesamt enthält das Set dieser Person 20 500 Datenpunkte. Bei dieser Auswertung wurden nur die taktilen Daten berücksichtigt.

Die Daten wurden preprocessed, sodass nicht jeder Punkt der Zeitserie individuell betrachtet wird. Dabei wurden jeweils die Zeitfenster, in welchen die Hand über einem Objekt war zusammengefasst und mit einem Fenster der Länge 10 gemittelt. Es hat sich gezeigt, dass dies in den beiden Ansätzen die Accuracy der Klassifikatoren um 10 Prozent gesteigert hat.

Weiterhin wurden alle irrelevanten Daten aus dem Datenset bereinigt und die übrig gebliebenen Daten Z-transformiert.

Methode

Für die Auswertung wurden jeweils 3 Klassifikatoren trainiert und mit cross-validation getestet. Die resultierenden Ergebnisse der 5-fach cross-validation wurden daraufhin gemittelt um die Performance des Modells zu bestimmen.

Die Klassifikatoren inklusive Hyperparameter waren:

1. k-nearest-neighbor mit $k=5$ und euklidischen Distanzen als Abstandsmessung
2. Multilayer Perceptron mit 3 hidden Layers (65-50-10), einer ReLU Aktivierungsfunktion und dem lbfgs (Broyden-Fletcher-Goldfarb-Shanno) Verfahren für die Backpropagation

3. SVM mit rbf Kernel und penalty parameter C=1.0

Ergebnisse

Für die Auswertung wurden folgende 5 Objekte benutzt : Box (B), Kugel (K), Pyramide (P), Viertelkreis (V), Dach (D).

Target-Klasse gegen Distraktor-Klasse

```
Target = B, Distraktor = K
Length of dataset: 296
Schuffel = False

SCORES
cross_val scores: [ 0.53333333  0.48333333  0.61666667  0.67241379
0.5862069 ]
average score kNN: 0.578390804598

-----

cross_val scores: [ 0.55          0.43333333  0.78333333  0.55172414
0.68965517]
average score MLP: 0.601609195402

-----

cross_val scores: [ 0.65          0.5          0.68333333  0.65517241
0.67241379]
average score SVM: 0.632183908046
```

Das Ergebnis hier zeigt, dass sich Target und Distraktor aus den Daten unterscheiden lassen, auch wenn nur sehr gering. Hierbei wurden zwei stark unterschiedliche Objekte genommen. Als Target die Box (B) und als Distraktor-Klasse die Kugel(K).

Selbes Setting, wobei jedoch die Daten geschuffelt wurden bevor trainiert wurde:

```
Target = B, Distraktor = K
Length of dataset: 296
Schuffel = True

SCORES
cross_val scores: [ 0.61666667  0.78333333  0.75          0.72413793
0.81034483]
average score kNN: 0.736896551724

-----

cross_val scores: [ 0.78333333  0.9          0.9          0.79310345
0.9137931 ]
average score MLP: 0.858045977011
```

```
cross_val scores: [ 0.7          0.73333333  0.73333333  0.70689655
0.79310345]
average score SVM: 0.733333333333
```

Dieses Ergebnis ist viel besser als das vorherige, nur dadurch das die Daten geshuffelt wurden. Grund ist womöglich die Verteilung der Daten, die dadurch randomisiert wird. Im vorherigem Durchlauf waren die Daten immer noch in der Zeitserie angeordnet. Ich denke, die Werte ohne shuffle sind realistischer, da ganze zusammenhängende Blöcke an Daten für das Testen aufgehoben wurden. Das Ergebnis für andere Konstellationen hielt sich etwa im selben Rahmen.

Als nächstes wird nicht nur das eine Target gegen ein Distraktor verglichen, sondern gegen mehrere, um auch zu zeigen, dass nicht nur die einzelnen Objekte unterschieden werden können, sondern die Rolle zwischen Distraktor und Target:

```
Target = B, Distraktor = K,P
Length of dataset: 520
Schuffel = False
```

SCORES

```
cross_val scores: [ 0.55238095  0.62857143  0.60576923  0.60194175
0.65048544]
average score kNN: 0.607829759238
```

```
cross_val scores: [ 0.59047619  0.64761905  0.75          0.67961165
0.81553398]
average score MLP: 0.696648173833
```

```
cross_val scores: [ 0.7047619  0.62857143  0.66346154  0.66019417
0.69902913]
average score SVM: 0.671203634553
```

Hier kann man sehen, dass sich das Ergebnis verbessert hat. Noch besser ist es, wenn man ein Target gegen alle Distraktor-Klassen vergleicht:

```
Target = B, Distraktor=K,P,V,D
Length of dataset: 836
Schuffel = False
```

SCORES

```
cross_val scores: [ 0.68452381  0.67261905  0.73809524  0.71084337
0.72289157]
average score kNN: 0.705794606999
```

```
cross_val scores: [ 0.60119048  0.63690476  0.73809524  0.65662651
0.80120482]
average score MLP: 0.686804360298
```

```
cross val scores: [ 0.75595238  0.77380952  0.78571429  0.76506024
0.78915663]
average score SVM: 0.773938611589
```

Eine Annahme ist, das die Ergebnisse durch die hinzukommenden Daten besser in die zwei Cluster Target/Distraktor zugeordnet werden können.

Objekt als Target gegen Objekt als Distraktor

Für die Auswertung wurden ebenfalls die oben genannten Objektklassen verwendet.

Hier wurde nur ein Objekt trainiert, jedoch sowohl in der Rolle eines Distraktors und eines Targets, um zu sehen, ob sich die Rolle aus den Daten eingliedern lässt:

```
Target = B, Distraktor = B
Length of dataset: 303
Schuffel = False
```

SCORES

```
cross val scores: [ 0.60655738  0.44262295  0.52459016  0.66666667
0.7          ]
average score kNN: 0.588087431694
```

```
cross val scores: [ 0.80327869  0.42622951  0.83606557  0.66666667
0.65          ]
average score MLP: 0.676448087432
```

```
cross val scores: [ 0.63934426  0.52459016  0.6557377  0.73333333
0.71666667]
average score SVM: 0.65393442623
```

Das Ergebnis von diesem Test zeigt, dass man hier bei dem Objekt Box anhand der taktilen Daten mit geringer Sicherheit die Rolle des Objektes zuordnen kann.

Auch hier waren die Ergebnisse mit schuffel=True signifikant höher.

Weiterhin konnte festgestellt werden, dass das MLP bei jedem der Objekte als bester Klassifikator hervorging:

```
Target = V, Distraktor = V
Length of dataset: 346
Schuffel = False
```

SCORES

```
cross val scores: [ 0.51428571  0.49275362  0.55072464  0.52173913
0.55072464]
average score kNN: 0.526045548654
```

```
cross_val scores: [ 0.44285714  0.63768116  0.75362319  0.57971014
0.71014493]
average score MLP: 0.624803312629

cross_val scores: [ 0.35714286  0.55072464  0.57971014  0.5942029
0.5942029 ]
average score SVM: 0.535196687371
```

Diskussion

Da das programmieren der Modell lange gedauert hat und ich noch für eine Prüfung lernen muss, blieb nicht viel Zeit für eine detaillierte Analyse der Ergebnisse. Das kommt mit der nächsten Auswertung. Was man jedoch sehen konnte war, das sowohl bei Versuch 1 als 2 die Ergebnisse besser als der Zufall waren. Daraus kann man schließen, dass es bestimmte Eigenschaften gibt, welche Targets und Distraktoren unterscheidet. Ebenfalls interessant wird zu sehen, wie sich die Ergebnisse ändern wenn man noch die Daten vom Cyberglove hinzuziehen wird oder nur die Daten vom Cyberglove nimmt.

Als ein weiteren Versuch steht noch die Frage zum Rückschluss auf die Suchstrategie. Dabei wird ein Objekt von verschiedenen Durchgängen jeweils mit dem Label des Targets in diesem Durchgang versehen und mit einem Klassifikator trainiert. Dadurch könnte man feststellen, ob man vom Distraktor auf das Target schließen kann.