

Logikprogrammieren Hausaufgaben

Blatt 10

Morten Seemann 6945442, Tore Wiedenmann 6488837

Aufgabe 1: Nächster-Nachbar-Klassifikator 1.) Überlegen Sie sich eine nichttriviale Klassifikationsaufgabe.

```
1 %siehe Datei fuer Datenbank von Iris Blumen
2 %-----
3 %Database
4 %Merkmals Vektor
5 % sepal length, sepal width, petal length, petal width.
6 % Klassen :   Iris Setosa
7 %              Iris Versicolor
8 %              Iris Virginica
9 %
10 % data (Merkmalsvektor,Klasse)
11 % test(...) identisch, zum testen des klassifikators.
```

2.) Definieren Sie ein Prädikat, das den Abstand von zwei Punkten in einem rein numerischen Merkmalsraum mit beliebiger Dimensionalität berechnet.

```
1 %Euklidische Norm
2 metric1d(X,Y,Dist):- Dist is (X-Y)**2.
3
4 metric([],[],0).
5 metric([H|T],[H2|T2],Dist):- metric1d(H,H2,D1),
6                               metric(T,T2,D2),
7                               Dist is D1+D2.
8 %Ohne ziehen der Quadratwurzel, da dies nur Rechenaufwand verursacht und
   zum Vergleichen nicht relevant ist.
```

3.) Definieren Sie ein Prädikat, das eine Liste erzeugt, die für jedes Trainings-beispiel eine Struktur mit zwei Angaben enthält: den Abstand zwischen dem Trainingsbeispiel und dem aktuellen Testbeispiel, sowie die Klassen-zugehörigkeit des jeweiligen Trainingsbeispiels.

```
1 %Bestimmt fuer Merkmalsvektor TarVec eine Liste von (Distanz,Klasse)
   paaren. Ein Paar fuer jeden Eintrag von data().
2 all_possibilities(TarVec,List):- findall((Dist,Class),
3                                     (data(DataVec,Class),
4                                     metric(DataVec,TarVec,Dist)),
5                                     List).
```

4.) Modifizieren Sie das Prädikat aus Aufgabenteil 3 so, dass nur der Distanzminimale Eintrag ausgegeben wird.

```
1 % Wir bestimmen alle Paare, sortieren aufsteigend und waehlen den ersten
   Listeneintrag.
2 nearest(TarVec,Best_Match):- all_possibilities(TarVec,List),
3                               sort(1,@=<,List,[Best_Match|_]).
```

Aufgabe 2: k-Nächste-Nachbarn-Klassifikator 1. Definieren Sie ein Prädikat, das die k nächsten Nachbarn einer zu klassifizierenden Beobachtung ermittelt.

```

1 %Wir bestimmen alle paare, sortiere und Waehlen die ersten K Elemente als
  Liste aus.
2 k_nearest(K,TarVec,List_of_Matches):- all_possibilities(TarVec,List),
3                                     sort(1,@=<,List,Sorted),
4                                     length(List_of_Matches,K),
5                                     append(List_of_Matches,_,Sorted).

```

2. Implementieren Sie ein Abstimmungsverfahren, das aufgrund der ermittelten Klassenzuordnungen eine Mehrheitssentscheidung herbeiführt. Bei gleicher Stimmenanzahl soll eine beliebige Entscheidung getroffen werden.

```

1 %Zum Partitionieren der Stimmen
2 voted_setosa( (_,iris-setosa)).
3 voted_veri( (_,iris-versicolor)).
4 voted_virgin( (_,iris-virginica)).
5
6 % Bestimmt die Mehrheit der Liste [K1,K2,K3], wenn Eintraege
  uebereinstimmen wird zufaellig einer als Sieger ausgewaehlt.
7
8 %Falls alle Eintraege Einzigartig sind
9 majority([K1,K2,K3],iris-setosa):-max_list([K1,K2,K3],K1),not((K1=K2;K1=K3
  )).
10 majority([K1,K2,K3],iris-versicolor):-max_list([K1,K2,K3],K2),not((K2=K1;
  K2=K3)).
11 majority([K1,K2,K3],iris-virginica):-max_list([K1,K2,K3],K3),not((K3=K1;K3
  =K2)).
12
13 % Falls 2 oder mehr uebereinstimmen
14 majority([K1,K1,K2],X):- K1\=K2,random(1,100,R1),random(1,100,R2),
15                          majority([R1,R2,0],X).
16 majority([K1,K2,K1],X):- K1\=K2,random(1,100,R1),random(1,100,R2),
17                          majority([R1,0,R2],X).
18 majority([K2,K1,K1],X):- K1\=K2,random(1,100,R1),random(1,100,R2),
19                          majority([0,R1,R2],X).
20 majority([K1,K1,K1],X):- random(1,100,R1),random(1,100,R2),random(1,100,R3
  ),
21                          majority([R1,R2,R3],X).
22
23 % Majority ist der Klassenname der die Meisten Stimmen aus der Paar-Liste
  List bekommen hat.
24 vote(List,[K1,K2,K3],Majority):- partition(voted_setosa,List,Votes1,Rest),
25                                   partition(voted_veri,Rest,Votes2,Votes3),
26                                   length(Votes1,K1),
27                                   length(Votes2,K2),
28                                   length(Votes3,K3),
29                                   majority([K1,K2,K3],Majority).

```

3. Untersuchen Sie anhand von geeigneten Beispieldaten, wie die Wahl von k das Entscheidungsverhalten Ihres Klassifikators beeinflusst.

Für $K = 1$ entspricht der neue Klassifikator dem bereits vorhandenen Klassifikator. Je weiter sich K von der 1 entfernt desto mehr werden Außreißer im Datensatz falsch Klassifiziert da die Meisten sich in der nähe befindlichen Ergebnisse nicht dem Erwarteten Wert entsprechen.

Aufgabe 3: Normalisierung 1.) Schreiben Sie ein Prädikat, das die Trainingsdaten für Ihren Klassifikator mit Hilfe der Z-Transformation normalisiert.

```
1 %Summieren und durch Listenlaenge teilen
2 mean(List,Mean):- length(List,N),
3                     sum_list(List,Sum),
4                     Mean is Sum/N.
5
6 % Analog zur euklidischen Metrik d(x1,x2) mit x2=[X,X,X,...,X] und X1
   beliebig.
7 metric2([],_,0).
8 metric2([H|T],X,Dist):- metric1d(H,X,D1),
9                           metric(T,X,D2),
10                          Dist is D1+D2.
11
12 % Summe der metric2 summen geteilt durch Listenlaenge.
13 dev(List,Dev):- length(List,N),
14                  mean(List,Mean),
15                  metric2(List,Mean,Sum),
16                  Dev is Sum/N.
```