

Übungen zur Vorlesung  
Grundlagen der Bilderzeugung und Bildanalyse WS 05/06

Musterlösung 8

Aufgabe 8: Merkmale für diskrete Kontouren

- a) Es sollen die Merkmale, die in Kapitel 5c, S.21 eingeführt wurden für den Datensatz 'Fish2Class.dat' berechnet werden. Nach dem Laden des Datensatzes erhalten wir zwei Variablen Spitz und Mond, die 16 bzw.13 Fischkontouren enthalten. Jede Fischkontour besteht aus 200 Punkten.

```
// load data
load('Fish2Class.dat');

// initialize variables
n = 200; // number of points on contour
m = 13+16; // number of samples
data = [ Spitz Mond ]
```

Wir wollen für jeden Fisch 8 Merkmale in Abhängigkeit der Parameter  $n_1, n_2, n_3$  berechnen. Dafür verwenden wir die Kombinationen wie in Kapitel 5c, S.21.

```
E = [1 0 0; 1 1 0 ; 1 0 1; 1 1 1 ; 2 0 0; 2 1 0; 2 0 1; 2 1 1];
```

Zudem müssen wir noch die Distanzen  $d_{i,1}, d_{i,2}, d_{i,3}$  berechnen. Dazu muss der komplette Datensatz um 1,2 bzw. 3 Stellen verschoben werden und die Distanzen zum Originaldatnesatz gebildet werden:

```
shiftdata_d1 = [ data(2:n,:); data(1,:) ] ;
shiftdata_d2 = [ data(3:n,:); data(1:2,:) ] ;
shiftdata_d3 = [ data(4:n,:); data(1:3,:) ] ;
distances_d1 = abs(data - shiftdata_d1 );
distances_d2 = abs(data - shiftdata_d2 );
distances_d3 = abs(data - shiftdata_d3 );
```

Nun können für jede Fischkontour die acht Merkmale berechnet werden:

```
for k = 1:size(E,1),
    PS(k,1:m) = ...
        sum(distances_d1.^E(k,1) .* distances_d2.^E(k,2) .* distances_d3.^E(k,3), 'r')
end;
```

- b) Hier soll die Güte der obigen Merkmale mit Hilfe des nächsten Nachbarn berechnet werden. Dazu werden die Daten `data` je nach Zugehörigkeit zur Klasse, Spitz mit 1 und Mond mit 2 gelabelt:

```
label = [ ones(1,16) 2*ones(1,13) ]; // create label vector
```

Die euklidische Distanz zwischen zwei Punkten in dem Datensatz kann einfach durch eine Distanzmatrix  $D$  repräsentiert werden. Der Eintrag  $D_{i,j}$  steht für den Abstand zwischen dem  $i$ -ten und dem  $j$ -ten Objekt. Auf der Diagonale der Distanzmatrix befindet sich der Abstand zu sich selbst. Dieser sollte natürlich bei der Auswertung nicht berücksichtigt werden:

```
// compute Distancematrix
AbsSq = ones(m,1)*sum(abs(PS).^2,'r');
Dmat = AbsSq' - 2*real(PS' * PS) + AbsSq;
Dmat = Dmat + diag(ones(1, m)*%inf); // to get a large distance to itself
```

Nun berechnen wir die Anzahl der korrekt klassifizierten Objekte:

```
// compute number of occurrences
[mm,idx] = min(Dmat,'c'); // compute nearest neighbors
correct = label(idx) == label; // same label as query?
sum(correct) // number of positives
```

Hierdurch ergeben sich für die Merkmale aus Aufgabenteil a) 16 korrekte Zuordnungen.

- c) Bei Änderungen von  $n_1, n_2, n_3$  verändern sich die Ergebnisse nicht wesentlich. Da die Konturen in Bogenlänge parametrisiert sind, ist die Distanz zum nächsten Nachbarn nicht sehr aussagekräftig. Die Merkmale hängen stark von der Skalierung ab; hohe Potenzen dominieren.

Ändert man die Skalierung, z.B. durch teilen der Originaldaten durch 1000, verbessert sich die Zuordnung zu 17.

```
data = [ Spitz Mond ]/ 1000;
```

21 Objekte werden richtig klassifiziert, indem man statt  $d_{i,1}d_{i,2}d_{i,3}$  z.B.  $d_{i,40}d_{i,80}d_{i,120}$  wählt und die Originaldaten mit 1000 skaliert:

```
d = 40;
shiftdata_d1 = [ data(d:n,:); data(1:(d-1),:) ] ;
shiftdata_d2 = [ data((d*2):n,:); data(1:((d*2)-1),:) ] ;
shiftdata_d3 = [ data((d*3):n,:); data(1:((d*3)-1),:) ] ;
distances_d1 = abs(data - shiftdata_d1 );
distances_d2 = abs(data - shiftdata_d2 );
distances_d3 = abs(data - shiftdata_d3 );
```

- d) Hier werden die Fourierdeskriptoren als Merkmal verwendet:

```
FT = mtlb_fft(data);
PS = FT(2:$,:) ./ (ones(n-1,1)* FT(2,:));
RefPhase = FT(3,:) ./abs(FT(3,:)) .* conj(FT(2,:)) ./abs(FT(2,:));
RefPhase = (ones(n-1,1) * RefPhase) .^ ( -[0:(n-2)]' * ones(1,m) );
PS = PS .* (RefPhase);
```

Hierdurch ergeben sich 26 korrekte Zuordnungen.