

Übungen Musterklassifikation

Sommersemester 2006

Prof. Dr. Stefan Posch

Dipl.Inform. Andre Gohr

Abgabe am 21.4.2006



Institut für Informatik
Universität Halle

Blatt 2

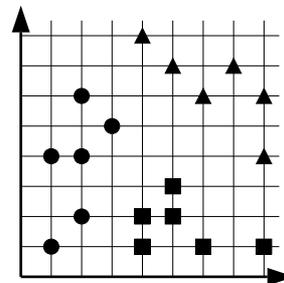
Aufgabe 2.1 (2 Punkte) Machen Sie einen Vorschlag zum Lernen von Hyperquadern für die Klassifikation beim Zwei-Klassen-Problem! (Der Problemkreis zerfällt hier in zwei Klassen.)

Aufgabe 2.2 (4 Punkte) Repräsentieren sie Klassifikator-Parameter zusammen mit ihrem Typ (z.B. Hyperquader) und der Anzahl der Klassen. Jeder Parameter-Satz soll wieder einen Namen zur Identifikation tragen. Bauen Sie in Ihr Programm einen Menüpunkt ein, der die aktuelle Stichprobe nutzt, um mit Hilfe des Algorithmus aus Aufgabe 2.1 Parameter für einen zwei Klassen Hyperquader-Klassifikator zu lernen. Ein weiterer Menüpunkt soll es gestatten einen Parametersatz auszusuchen und auf die aktuelle Stichprobe anzuwenden. Das Ergebnis ist eine neue Stichprobe, welche die klassifizierten Mustervektoren enthält.

Überlegen Sie, ob, wie und mit welchen Werkzeugen Sie in der Lage wären, eine Stichprobe (mit maximal 2 dimensional Vektoren) automatisch zu visualisieren (gnuplot, jai, qt, gl, ...?) (nur Lösungsidee).

Aufgabe 2.3 (3 Punkte) Gegeben seien drei Objektklassen, die durch zwei Merkmale charakterisiert werden. Es existiere die in der nebenstehenden Abbildung gezeigte Menge von Referenzobjekten mit bekannten Merkmalswerten.

Skizzieren Sie die sich ergebenden Klassengebiete, wenn mit Hilfe der Nächster-Nachbar-Regel klassifiziert wird. Betrachten Sie dabei die alternative Verwendung folgender Metriken: Euklidischer Abstand, Maximum Norm bzw. City-Block-Abstand. Markieren Sie auch Bereiche, in denen keine Entscheidung für eine bestimmte Klasse möglich ist



Aufgabe 2.4 (2 Punkte) Machen Sie einen Vorschlag zur Modifikation des (m)-Nächster-Nachbar-Klassifikators zur Berücksichtigung unbekannter Bereiche im Merkmalsraum. Dabei soll ein Muster, das keinem Muster der Stichprobe "ähnlich" ist, zurückgewiesen werden.

Aufgabe 2.5 (4 Punkte) Programmieren Sie einen Nearest-Neighbour-Klassifikator mit mind. 2 verschiedenen Distanzfunktionen. Testen Sie Ihr Programm.

(Source-Codes in Java, Perl, C, C++ bitte rechtzeitig an andre.gohr@informatik.uni-halle.de. Theoretische Lsgs bitte ausgedruckt zu den Abgabeterminen mitbringen.)