

Blatt 4:  $k$ -Nearest-Neighbor Klassifikator

Abgabe Montag, 27. November 2000, in der Vorlesung

6.  $k$ -NN Klassifikator: In dieser Aufgabe soll der  $k$ -Nearest-Neighbor-Klassifikator implementiert und evaluiert werden. Um einen neuen Datenpunkt  $x$  einer Klasse zuzuweisen, ermittelt er die  $k$  Trainingsvektoren, die  $x$  im Sinne einer vorgegebenen Metrik am ähnlichsten sind, und entscheidet sich für die Klasse, die unter diesen  $k$  „nächsten Nachbarn“ am häufigsten vorkommt.

Schreiben Sie eine Matlab-Funktion

```
function c = KNN(X, SAMPLE, CLASS, k) ,
```

die einen Eingabevektor  $x \in \mathbb{R}^d$  auf der Basis der Matrix von Beispielvektoren  $SAMPLE \in \mathbb{R}^{n \times d}$ ,  $CLASS \in \mathbb{N}^n$  mittels des  $k$ -NN Klassifikators einer Klasse  $c$  zuordnet. Folgen Sie in Ihrer Implementierung den folgenden Stichpunkten:

- Die Trainingsvektoren sollen zeilenweise in  $SAMPLE$  abgelegt werden. Der Vektor  $CLASS$  enthält die zugehörigen Klassenlabel der Trainingsvektoren.
- Berechnen Sie zunächst die Abstände von  $x$  zu allen Trainingsvektoren. Verwenden Sie dabei ein quadratisches Abstandsmaß  $\|x - y\| = \sum_{i=1}^d (x_i - y_i)^2$ .
- Sortieren Sie dann die Trainingsvektoren gemäß ihrer Abstände. Hierzu sollten Sie natürlich **nicht** die Vektoren physikalisch permutieren, sondern nur die Indizes sortieren. Die Matlab-Funktion `sort()` leistet hierbei gute Dienste.
- Bestimmen Sie die Klasse  $c$ , die unter den ersten  $k$  sortierten Trainingsvektoren am häufigsten vorkommt. Falls zwei oder mehrere Klassen gleich oft vorkommen, so entscheidet der nächstgelegene Vektor einer dieser Klassen.

Erstellen Sie die folgenden graphischen Darstellungen:

- (a) Benutzen Sie zum Testen den in Aufgabe 3 (a) erzeugten Datensatz. Verwenden Sie nun  $n \in \{10, 20, 50, 100, 200, 500, 1000\}$  zufällig ausgewählte Trainingsdaten als Trainingsmenge und evaluieren Sie die Klassifikationsleistung des  $k$ -NN Klassifikators mit jeweils 1000 nicht verwendeten Datenpunkten ( $k \in \{1, 5\}$ ). Plotten Sie den Klassifikationsfehler in Abhängigkeit von  $n$ . Verwenden sie dabei eine logarithmische Skala für  $n$ . Visualisieren sie das Klassifikationsergebnis, indem sie Trainings- und Testdaten als zweidimensionale Punkte eintragen und entsprechend ihrer Klassenzuordnung einfärben. (3P)
- (b) Verwenden Sie nun  $n = 1000$  Datenpunkte, um die verschiedenen  $k$ -NN Klassifikatoren zu vergleichen,  $k \in \{1, \dots, 10\}$ . Evaluieren Sie die Klassifikationsleistung wiederum mit den nicht verwendeten 9000 Datenpunkten. Plotten Sie das Klassifikationsergebnis in Abhängigkeit von  $k$ . Tragen Sie den Bayes-Fehler als Konstante in Ihren Graph ein. Ist das Klassifikationsergebnis für gerade  $k$  signifikant besser als das für  $k - 1$ ? Warum nicht? (3P)