

Audio Content Analysis
Klassifikation: K-Nearest Neighbours

Abgabetermin: 29.05.2013

Allgemeine Hinweise:

Die Bearbeitung der Aufgaben findet in Gruppen von je zwei oder drei Studenten statt. Abzugeben sind die Matlab Skripte mit Namen und Matrikelnummern aller Teilnehmer per E-Mail an:

von_coler@mailbox.tu-berlin.de.

Hinweise zum Matlab-Code:

- Stellen Sie die Ausführbarkeit der Skripte sicher.
- Die Skripte sind mit Kommentaren zu versehen, sodass jeder Schritt nachvollziehbar ist.
- Achten Sie bei eventuellen Plots auf korrekte Achsenbeschriftungen.

K-Nearest Neighbours

Der k-nearest neighbours Algorithmus (KNN) zählt zu den einfachsten nichtparametrischen Klassifikatoren. Ziel dieser Übung ist es, eine Klassifikation von einzelnen Instrumentenklängen in die Klassen *percussive* und *tonal* mittels KNN zu implementieren und diese zu testen. Dafür stehen ein Trainings- und ein Testdatensatz zur Verfügung. Die Programmierung ist ohne Verwendung der Matlab-Funktionen zur KNN-Klassifikation (z.B. `knnclassify()`) vorgesehen.

1 Training

Die Trainingsphase eines KNN Klassifikators besteht hauptsächlich in der Zuweisung der Samples im Trainingset zu Klassen. Dafür steht ein Datensatz von Instrumentenklängen (die Samples), getrennt in die Klassen *percussive* und *tonal*, zur Verfügung. Als Grundlage für das Clustering dienen die spektralen Merkmale *Spectral Rolloff*, *Spectral Flux* und *Spectral Centroid*, wie sie im vorherigen Aufgabenblatt implementiert wurden.

Erstellen Sie ein Skript `training.m`, welches das Trainingset (getrennt in Klassen) einliest¹ und berechnen sie diese Merkmale für alle Samples mit der Funktion `blockwise_features()` aus Aufgabenblatt 3. Dafür wird eine Hop-Size von 512 Samples und eine Frame-Size von 1024 Samples verwendet. Aus den resultierenden zeitlichen Verläufen soll nun ein statistischer Wert ('Meta

¹z.B. mit `dir('Pfad-mit-Audiofiles')`

Feature') pro Merkmal berechnet werden: Für den Rolloff und den Centroid das arithmetische Mittel², für den Flux die Varianz³. Somit wird jedes Sample durch einen dreidimensionalen Merkmalsvektor beschrieben. Diese drei Dimensionen sind für das gesamte Trainingset (tonal+percussive) mit deren Varianz v_n zu normieren. Plotten Sie in einem Scatterplot⁴ die 3 möglichen Merkmalspaare (1+2,1+3,2+3), wobei die Klassen eindeutige Marker erhalten. So erhalten Sie einen Eindruck von der Trennbarkeit der Klassen.

Erzeugen sie eine Datenstruktur KNN, welche die normierten Merkmalsvektoren für die perkussiven und die tonalen Samples sowie die Varianz v_n aller drei Merkmale enthält.

2 Klassifikation

Schreiben Sie eine Funktion `classify()`, welche den Merkmalsvektor S_n eines zu klassifizierenden Samples, die Datenstruktur KNN aus der Analyse sowie den Parameter k als Argumente erhält. S_n muss zunächst mit der im Training bestimmten Varianz v_n normalisiert werden. Finden Sie dann innerhalb dieser Funktion die k Samples des Trainingssets (tonal+percussive), welche dem Sample S nach euklidischer Distanz am nächsten sind⁵. Anschließend wird ausgewertet, aus welcher der Klassen die Mehrzahl der k nächsten Nachbarn stammt⁶. Diese Klasse wird dann von der Funktion als Rückgabewert ausgegeben.

3 Test

Schreiben Sie ein Skript `test.m`, welches alle Dateien des Testordners einliest und klassifiziert. Wählen Sie k sinnvoll. Weiterhin soll es möglich sein, für die Klassifikation auf einzelne Merkmale zu verzichten. Evaluieren Sie die Güte des Klassifikators für verschiedene Merkmalskombinationen, indem Sie das Ergebnis der Klassifikation mit den Sollwerten vergleichen. Visualisieren Sie die beste Lösung in einem Scatter-Plot (Trainingset + Testset).

²`mean()`

³`var()`

⁴`scatter()`

⁵z.B. `sort()`

⁶z.B. `mode()`