

In den Softwaremodulen verwendete Datentypen

Vektormatrix:

- Mehrdimensionale Vektoren werden grundsätzlich in einer Matrix **zeilenweise** angeordnet, d.h. jede Zeile der Matrix bildet einen Vektor.
- Bsp.: Matrix der 8 Eckpunkte des 3 Dimensionalen Einheitswürfels mit Ecke im Ursprung
 $Wuerfel = [0,0,0;0,0,1;0,1,0;0,1,1;1,0,0;1,0,1;1,1,0;1,1,1];$

Kovarianzmatrix:

- Kovarianzmatrizen sind in den verwendeten Modulen grundsätzlich diagonal und es werden daher nur die Diagonalkomponenten in einem **Zeilenvektor** gespeichert:
 z.B. entspräche $A = [1,1,1]$; der Einheitsmatrix der Größe 3

Normalverteilungszellenobjekt:

- Normalverteilungen werden durch die Prioriwahrscheinlichkeiten, die Mittelwertsvektoren und die Kovarianzmatrizen charakterisiert. Sie werden durchgehend in einem Zellenobjekt gespeichert bei dem
 - das erste Element ein Vektor mit den Prioriwahrscheinlichkeiten ist
 - das zweite Element eine **Vektormatrix** mit den Mittelwertsvektoren
 - das dritte Element eine **Vektormatrix** die in den Zeilen die Kovarianzvektoren der Einzelkomponenten enthält.

z.B.: $Normal = \{[0.2,0.8],[1,2,3;4,5,6],[6,7,8;9,10,11]\};$

stellt eine Mischverteilung da mit 2 Komponenten dar:

1. Prioriwahrscheinlichkeit: 0.2 , Mittelwert bei [1,2,3], Kovarianzmatrix $\begin{bmatrix} 6 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 8 \end{bmatrix}$
2. Prioriwahrscheinlichkeit: 0.8 , Mittelwert bei [4,5,6], Kovarianzmatrix $\begin{bmatrix} 9 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 11 \end{bmatrix}$

Zugehörigkeitszellenobjekt:

Nach der Analyse kann es gewünscht sein, zu jeder Klasse die zugehörigen Vektoren zu ermitteln. Daher werden von den Modulen LBG und EM die Ergebnisse z.T. der Form zurückgegeben, daß in **einem** Zellenobjekt für jede Klasse eine Referenz auf die entsprechenden Vektoren gespeichert werden. D.h:

- Das erste Element des Zellenobjektes enthält die Nummern der Vektoren die in Klasse1 klassifiziert wurden.
- Das zweite Element des Zellenobjektes enthält die Nummern der Vektoren die in Klasse2 klassifiziert wurden
- usw...

Ein Beispiel für ein solches Zellenobjekt wäre: $Erg = \{[1,4,6],[2],[3,5],[]\}$

Das würde bedeuten, dass die Vektoren 1,4 und 6 in die erste Klasse, der Vektor 2 in Klasse 2, die Vektoren 3 und 5 in die Klasse 3 und kein Vektor in die Klasse 4 klassifiziert wurden.

Wären X nun unsere Merkmalsvektoren so erhielte man durch: $X(Erg\{2\}, :)$ eine Vektormatrix mit den Vektoren, die in Klasse 2 klassifiziert wurden. So erlaubt das Zellenobjekt eine sehr einfache Trennung der Vektoren.

Funktionsbeschreibung des Moduls em.m

Das Modul em ermöglicht die Datenanalyse mittels EM Algorithmus. Normalerweise ruft man das Programm mit folgenden drei Funktionen auf:

Aufruf: *em('Dateiname',P,Klassenanzahl)*

Lädt Merkmalsvektoren aus der Datei Dateiname und initialisiert die Analyse mit einer variablen Anzahl von Klassen.

Parameter	Datentyp	Bedeutung
P	String	Falls eine grafische Ausgabe gewünscht wird muss für P der String 'Paint' übergeben werden, ansonsten erfolgt die Analyse rein rechnerisch.
Klassenanzahl	Positive Ganzzahl	Anzahl der Komponenten die für die Analyse benutzt werden sollen (Default: 10)

Aufruf: *em('LoadWave','Dateiname')*

Falls die Datei Dateiname existiert, verwendet das Modul em diese Datei zur Anzeige der entsprechenden Sounddaten. Diese Funktion ist nur zur grafischen Sprachdatenanalyse sinnvoll und wenn vorher die zur Wavedatei gehörigen entsprechenden Merkmalsvektoren geladen wurden.

Parameter	Datentyp	Bedeutung
Dateiname	String	Name der zu den Daten gehörigen Wavedatei

z.B.: initialisieren diese beiden Zeilen Matlab zur Analyse der Datei s11.wav mit den zugehörigen Merkmalsvektoren es11.txt mit 6 Mischverteilungskomponenten.

em('es11.txt','Paint',6);

em('LoadWave','s11.wav');

Die Ergebnisse nach Drücken des Run-Buttons werden mit der folgenden Funktion ermittelt:

Aufruf: *[V,R,S] = em('Result')*

Liefert das Ergebnis nach der Analyse der Daten. Gibt die geschätzten Klassen zurück. Man kann die Zuordnung der Vektoren zu den Klassen über das Modul Klassifizieren mittels folgendem Aufruf erhalten: *[G,A,C]=Klassifizieren(V,R);*

Parameter	Datentyp	Bedeutung
Keine		
Wiedergabewert	Datentyp	Bedeutung
V	Vektormatrix	Enthält die im Modul geladenen (zu klassifizierenden) Vektoren
R	Normalverteilungs Zellenobjekt	Enthält die geschätzten Normalverteilungen für die initialisierte Anzahl von Klassen
S	Normalverteilungs Zellenobjekt	Nur bei zufälliger Initialisierung belegt. Enthält die wahren Werte der von EM zufällig generierten Normalverteilungen.

Funktionsbeschreibung des Moduls em.m

Die folgenden Funktionen dienen zur erweiterten Bedienung des Moduls und können bei Bedarf verwendet werden (z.B. zur automatischen Analyse) :

Aufruf: <i>em('Init',A,Klassenanzahl)</i>		
Initialisiert das Modul EM. Schließt das Bild, setzt Statusvariablen auf Defaultwerte.		
Parameter	Datentyp	Bedeutung
A		Nicht verwendet
Klassenanzahl	Positive Ganzzahl	Anzahl der für die Schätzung anzusetzenden Normalverteilungskomponenten (Default: 10)
Aufruf: <i>em('InitVecs',Vektoren,Klassenanzahl)</i>		
Parameter	Datentyp	Bedeutung
Vektoren	Vektormatrix	Stichprobenvektoren mit denen em initialisiert werden soll.
Klassenanzahl	Positive Ganzzahl	Anzahl der Komponenten die für die Analyse benutzt werden sollen (Default: 10)
Aufruf: <i>em('Initfigure')</i>		
Initialisiert die grafische Ausgabe des Programms. Kann jederzeit aufgerufen werden.		
Aufruf: <i>em('Rnd',Anzahl)</i>		
Erzeugt zur Analyse eine zufällige mischverteilte Stichprobe mit einer Variablen Anzahl an Komponenten. Pro Komponente werden 50 zufällige Vektoren erzeugt.		
Parameter	Datentyp	Bedeutung
Anzahl	Positive Ganzzahl	Anzahl der zufällig zu generierenden Mischverteilungskomponenten (Default: 2)
Aufruf: <i>em('Dim')</i>		
Zeigt eine Dialogbox in der die angezeigten Dimensionen eingestellt werden können (nur bei grafischer Ausgabe)		
Aufruf: <i>em('Paint')</i>		
Zeichnet das Bild neu. Diese Funktion braucht nur nach <i>em('Initfigure')</i> aufgerufen zu werden um das Bild einmalig zu zeichnen.		
Aufruf: <i>em('Ende')</i>		
Schliesst das Bild		
Aufruf: <i>em('Step')</i>		
Berechnet einen EM-Schritt		
Aufruf: <i>em('Run')</i>		
Berechnet das Endergebnis. Wenn die Änderung aller Werte kleiner als 1E-3 ist wird konvergenz angenommen.		

Funktionsbeschreibung des Moduls lbg.m

Das Modul lbg ermöglicht die Datenanalyse mittels lbg Algorithmus. Normalerweise ruft man das Programm mit folgenden drei Funktionen auf:

Aufruf: *lbg('Dateiname',P,Klassenanzahl)*

Lädt Merkmalsvektoren aus der Datei Dateiname und initialisiert die Analyse mit einer variablen Anzahl von Klassen.

Parameter	Datentyp	Bedeutung
P	String	Falls eine grafische Ausgabe gewünscht wird muss für P der String 'Paint' übergeben werden, ansonsten erfolgt die Analyse rein rechnerisch.
Klassenanzahl	Positive Ganzzahl	Anzahl der Klassen die für die Analyse benutzt werden sollen (Default: 10)

Aufruf: *lbg('LoadWave','Dateiname')*

Falls die Datei Dateiname existiert, verwendet das Modul lbg diese Datei zur Anzeige der entsprechenden Sounddaten. Diese Funktion ist nur zur grafischen Sprachdatenanalyse sinnvoll und wenn vorher die zur Wavedatei gehörigen entsprechenden Merkmalsvektoren geladen wurden.

Parameter	Datentyp	Bedeutung
Dateiname	String	Name der zu den Daten gehörigen Wavedatei

z.B.: initialisieren diese beiden Zeilen Matlab zur Analyse der Datei s11.wav mit den zugehörigen Merkmalsvektoren es11.txt mit 6 Mischverteilungskomponenten.

lbg('es11.txt','Paint',6);

lbg('LoadWave','s11.wav');

Die Ergebnisse nach Drücken des Run-Buttons werden mit der folgenden Funktion ermittelt:

Aufruf: *[V,R,S] = lbg('Result')*

Liefert das Ergebnis nach der Analyse der Daten. Gibt die geschätzten Klassen zurück. Man kann die Zuordnung der Vektoren zu den Klassen über das Modul Klassifizieren mittels folgendem Aufruf erhalten: *[G,A,C]=Klassifizieren(V,R);*

Parameter	Datentyp	Bedeutung
Keine		
Wiedergabewert	Datentyp	Bedeutung
V	Vektormatrix	Enthält die im Modul geladenen (zu klassifizierenden) Vektoren
R	Vektormatrix	Enthält die Mittelpunktsvektoren für die initialisierte Anzahl von Klassen
S	Zugehörigkeits Zellenobjekt	Enthält die Zugehörigkeiten der Vektoren zu den Klassen als ZugehörigkeitsZellenobjekt (s. Datentypen)

Funktionsbeschreibung des Moduls lbg.m

Die folgenden Funktionen dienen zur erweiterten Bedienung des Moduls und können bei Bedarf verwendet werden (z.B. zur automatischen Analyse) :

Aufruf: <i>lbg('Init',A,Klassenanzahl)</i>		
Initialisiert das Modul lbg. Schließt das Bild, setzt Statusvariablen auf Defaultwerte.		
Parameter	Datentyp	Bedeutung
A		Nicht verwendet
Klassenanzahl	Positive Ganzzahl	Anzahl der für die Schätzung anzusetzenden Klassen (Default: 10)
Aufruf: <i>lbg('InitVecs',Vektoren,Klassenanzahl)</i>		
Parameter	Datentyp	Bedeutung
Vektoren	Vektormatrix	Stichprobenvektoren mit denen lbg initialisiert werden soll.
Klassenanzahl	Positive Ganzzahl	Anzahl der Klassen die für die Analyse benutzt werden sollen (Default: 10)
Aufruf: <i>lbg('Initfigure')</i>		
Initialisiert die grafische Ausgabe des Programms. Kann jederzeit aufgerufen werden.		
Aufruf: <i>lbg('Rnd')</i>		
Erzeugt zur Analyse eine zufällige gleichverteilte Stichprobe mit 100 2 dimensional Vektoren.		
Aufruf: <i>lbg('Dim')</i>		
Zeigt eine Dialogbox in der die angezeigten Dimensionen eingestellt werden können (nur bei grafischer Ausgabe)		
Aufruf: <i>lbg('Paint')</i>		
Zeichnet das Bild neu. Diese Funktion braucht nur nach <i>lbg('Initfigure')</i> aufgerufen zu werden um das Bild einmalig zu zeichnen.		
Aufruf: <i>lbg('Ende')</i>		
Schliesst das Bild		
Aufruf: <i>lbg('Step')</i>		
Berechnet einen lbg-Schritt		
Aufruf: <i>lbg('Run')</i>		
Berechnet das Endergebnis. Wenn die Änderung aller Werte kleiner als 1E-3 ist wird konvergenz angenommen.		

Funktionsbeschreibung der Hilfsprogramme

Aufruf: *emload('Dateiname', Klassenanzahl)*

Initialisiert das Modul em mit der in der durch *KsaveVar* gespeicherten Mischverteilung mit einer variablen Anzahl von Klassen.

Parameter	Datentyp	Bedeutung
Klassenanzahl	Positive Ganzzahl	Anzahl der zur Analyse zu benutzenden Mischverteilungskomponenten.
Dateiname	String	Name der zu ladenden Merkmalsvektoren (Endung wird angehängt !)

Aufruf: *emsave(Nr, 'Dateiname')*

Speichert die von em in eine variable Klasse klassifizierten Vektoren in eine Datei.

Parameter	Datentyp	Bedeutung
Nr	Positive Ganzzahl	Nummer der Klasse, deren Vektoren zu speichern sind
Dateiname	String	Name der Datei, in die die Vektoren gespeichert werden sollen.

Aufruf: *emrauschen('Dateiname')*

Initialisiert und startet die grafische Analyse der Merkmalsvektoren einer Datei mit 2 Klassen. Diese Programm verdeutlicht, dass man bei mehrmaligem Starten unterschiedliche Ergebnisse erhält. z.B. starten sie mehrmals *emrauschen('es11.txt')* und beobachten sie die Ergebnisse.

Parameter	Datentyp	Bedeutung
Dateiname	String	Name der zu Analysierenden Datei.

Aufruf: *KsaveVar(X, 'Dateiname')*

Speichert den Inhalt der Matrix X in eine Datei. (Benutzt der Matlab Befehl *Save*)

Parameter	Datentyp	Bedeutung
Dateiname	String	Name der Datei, in den die Matrix X geschrieben werden soll.

Aufruf: *lbgload('Dateiname', Klassenanzahl)*

Lädt Dateien zur Sprachdatenanalyse mit variabler Klassenanzahl. Es wird automatisch die den Merkmalsvektoren entsprechende Wavedatei mitgeladen. z.B.: *lbgload('es13', 10)* lädt die Datei es13.txt und die Datei s13.wav und initialisiert den LBG Algorithmus mit 10 Klassen.

Parameter	Datentyp	Bedeutung
Dateiname	String	Name der zu ladenden Merkmalsvektoren (.txt wird angehängt !)
Klassenanzahl	Positive Ganzzahl	Anzahl der zur Analyse zu verwendenden Klassen

Funktionsbeschreibung der Hilfsprogramme

Aufruf: *PlotStichprobe(Vektoren,'Farbstring')*

Plottet die ersten beiden Dimensionen einer Vektormenge in variabler Form in ein Diagramm.

z.B. *PlotStichprobe(X,'rx');*

Parameter	Datentyp	Bedeutung
Vektoren	Vektormatrix	Vektoren die geplottet werden sollen.
Farbstring	String	Formatcode wie er im <i>plot</i> Befehl von Matlab verwendet wird. Gibt Farbe und Form an, z.B. 'rx','go','b', usw...

Aufruf: *[V,L] = readvec('Dateiname')*

Liest Merkmalsvektoren, die von HTK in spezieller Weise in eine Textdatei exportiert wurden aus.

Parameter	Datentyp	Bedeutung
Dateiname	String	Name der Datei, die die Merkmalsvektoren enthält.
Wiedergabewert	Datentyp	Bedeutung
V	Vektormatrix	Enthält die in der Datei gespeicherten Vektoren
L	Positive Ganzzahl	Enthält die Dimensionalität der Vektoren

Aufruf: *Matrix3D = lbglauf(Vec, Dim, Class)* oder *emlauf(Vec, Dim, Class)*

Misst die Laufzeit der Analyse von zufällig generierten Vektoren im Modul *lbgl.m* / *em.m*. Das Hilfsprogramm permutiert alle in den Vektoren *Vec*, *Dim* und *Class* gespeicherten Zahlwerte und misst die Laufzeiten für all diese Kombinationen. Diese werden in einer 3 Dimensionalen Matrix gespeichert. Soll nur eine Abhängigkeit getestet werden muss daher für die anderen Werte Konstanten angegeben werden.

z.B. Testen der Abhängigkeit der Anzahl der Vektoren:

Z = LBGLauf([1:25:1000],2,5); plot([1:25:1000],Z);

der Dimensionsanzahl: *Z = LBGLauf(100,[1:25:1000],5); plot([1:25:1000],Z(1,,:));*

der Klassenanzahl: *Z = LBGLauf(100,2,[1:25:1000]); plot([1:25:1000],Z(1,1,:));*

Parameter	Datentyp	Bedeutung
Vec	Positive Ganzzahl	Ein Vektor mit den zu testenden Anzahlen von Vektoren z.B. [1 10 100 1000 10000]
Dim	Positive Ganzzahl	Ein Vektor mit der Anzahl der zu testenden Dimensionen z.B. [10 50 100 150 200 250 300 350]
Class	Positive Ganzzahl	Ein Vektor mit der Anzahl der zur Analyse zu verwendenden Klassen z.B. [2 10 20 30 40 50 60 70 80]
Wiedergabewert	Datentyp	Bedeutung
Matrix3D	3 Dimensionale Matrix	Sie enthält die Laufzeiten des Programmes pro Durchgang in Sekunden.

Beschreibung des Moduls Klassifizieren.m

Das Modul Klassifizieren ist in der Lage sowohl Vektoren nach einer gegebenen Mischverteilung zu klassifizieren (Normalverteilungsklassifikator), als auch nach einem Minimum-Distance Verfahren alleinig über die Klassenmittelpunkte. Welcher Klassifikator angewandt wird hängt von den übergebenen Parametern ab. Die unterschiedlichen Parameter sind in den Tabellen grau schraffiert.

Parameter für Verwendung eines Normalverteilungsklassifikators:

Aufruf: $[G \ A \ C] = \text{Klassifizieren}(\text{Vektoren}, \text{Normal})$		
Parameter	Datentyp	Bedeutung
Vektoren	Vektormatrix	Enthält die zu klassifizierenden Vektoren in Form einer Vektormatrix.
Normal	Normalverteilungszellenobjekt	Enthält die Komponenten der Mischverteilung, nach der die Vektoren klassifiziert werden sollen.
Wiedergabewert	Datentyp	Bedeutung
G	Vektor	Enthält einen Vektor mit den Gewinnerklassen der einzelnen Vektoren. G(1) ist also die Gewinnerklasse des Vektors 1, etc...
A	Matrix	Enthält die logarithmierten Wahrscheinlichkeiten der einzelnen Vektoren für jede Klasse in ihren Zeilen. d.h.: A(i,j) ist die Wkt. dass Vektor i in Klasse j fällt.
C	Zugehörigkeitszellenobjekt	Zellenobjekt, dass für jede Klasse einen Vektor mit den zugehörigen Vektornummern enthält.

Parameter für Verwendung eines Minimumdistanceklassifikators:

Aufruf: $[G \ A \ C] = \text{Klassifizieren}(\text{Vektoren}, \text{Mittelwerte})$		
Parameter	Datentyp	Bedeutung
Vektoren	Vektormatrix	Enthält die zu klassifizierenden Vektoren in Form einer Vektormatrix.
Mittelwerte	Vektormatrix	Enthält die Klassenmittelpunkte als Vektormatrix, nach denen die Vektoren klassifiziert werden sollen.
Wiedergabewert	Datentyp	Bedeutung
G	Vektor	Enthält einen Vektor mit den Gewinnerklassen der einzelnen Vektoren. G(1) ist also die Gewinnerklasse des Vektors 1, etc...
A	Matrix	Enthält die euklidischen Abstände der einzelnen Vektoren für jede Klasse in ihren Zeilen. d.h.: A(i,j) ist Abstands des Vektors i vom Klassenmittelpunkt der Klasse j fällt.
C	Zugehörigkeitszellenobjekt	Zellenobjekt, dass für jede Klasse einen Vektor mit den zugehörigen Vektornummern enthält.