Computational Intelligence: Teil 2

Vorlesungsmitschrift (Franz Pernkopf)

19. Juni 2018

Inhaltsverzeichnis

1 Einführung

Prinzipiell gibt es 3 Arten des Lernens:

1. Überwachtes Lernen:
   Gegeben sind die Daten $\mathcal{X} = \{\langle \mathbf{x}_1, t_1 \rangle, \ldots, \langle \mathbf{x}_N, t_N \rangle\}$, N entspricht der Anzahl der Samples $\langle \mathbf{x}_N, t_N \rangle$, $\mathbf{x}_n \in \mathbb{R}^d$ ist eine Beschreibung des Objekts (z.B.: Vektor von Messwerten) und t_n beschreibt den Zielwert (Target).
   Wenn $t_n \in \mathbb{N}$ spricht man von Klassifikation; z.B. $t_n \in \{1, \ldots, c\}$, c beschreibt die Anzahl der Klassen.Die wichtigsten Klassifikationsmethoden sind Neuronale Netze, Support-Vector-Maschinen, Bayes Klassifikator, usw. Bei $t_n \in \mathbb{R}$ handelt es sich um Regression.

2. Unüberwachtes Lernen:
   Beim Unüberwachten Lernen sind die Daten $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ ohne Zielwert gegeben.
   Anwendungsfälle sind:
   - explorative Datenanalyse: Dimensionsreduktion z.B. durch Hauptkomponenten-Transformation (Principal component analysis PCA)
   - Schätzen von Wahrscheinlichkeitsverteilungen: Maximum-Likelihood Schätzer, Bayes Schätzer, usw.
   - Auffinden von Zusammenhängen in Daten wie statistische Abhängigkeiten, usw.

3. Reinforcement Lernen:
   Lernen ohne unmittelbare Rückmeldung ob die Zwischenschritte korrekt sind, nur das Resultat wird bewertet.
   Zum Beispiel: Ein Roboter soll einen Würfel finden. Er bekommt aber erst Feedback (positiv oder negativ) nachdem er das Objekt gefunden hat. Aufgrund dieses Feedbacks können vorangegangene Aktionen bewertet werden.

1.1 Grundlagen der Wahrscheinlichkeitstheorie

Die Zufallsvariable kann diskret oder kontinuierlich sein. Kontinuierliche Verteilungsfunktionen werden als Wahrscheinlichkeitsdichtefunktionen bezeichnet und Verteilungsfunktionen über eine diskrete Variable sind als Wahrscheinlichkeitsmassenfunktionen bekannt. Wir bezeichnen die Wahrscheinlichkeit, dass die diskrete Zufallsvariable X den Wert x annimmt als $P(X = x)$. $P(X = x)$ wird auch als $P(x)$ abgekürzt.

Die Wahrscheinlichkeit liegt im Bereich $0 \leq P(x) \leq 1$. Die Summe über den gesamten Wertebereich einer diskreten Variable ist $\sum_x P(X = x) = 1$. Bei einer kontinuierlichen Zufallsvariable gilt: $\int_{-\infty}^{\infty} p(x)dx = 1$

Ein fairer Würfel hat eine uniforme Wahrscheinlichkeitsverteilung / Wahrscheinlichkeitsmassefunktion d.h. $P(X = x) = \frac{1}{6}$ wobei die Augenzahl $x \in \{1, \ldots, 6\}$ ist.

Handelt es sich um eine Menge von Zufallsvariablen $\mathbf{X} = \{X_1, X_2, \ldots, X_T\}$ schreiben wir die Verbundwahrscheinlichkeitsverteilung $P(X_1, X_2, \ldots, X_T) = P(\mathbf{X} = \mathbf{x})$ oder $P(\mathbf{x})$. Das gemeinsame auftreten von $X_1, X_2, \ldots, X_T$ in $P(\cdot)$ wird als Verbundwahrscheinlichkeit (Joint probability) bezeichnet. Weiters, die Wahrscheinlichkeit $P(\mathbf{X}, \mathbf{Y})$ ist die Wahrscheinlichkeit, dass $\mathbf{X}$ und $\mathbf{Y}$ gemeinsam auftreten. $P(\mathbf{X}, \mathbf{Y})$ ist die Verbundwahrscheinlichkeit für die Menge $\mathbf{X} = \{X_1, \ldots, X_M\}$ und $\mathbf{Y} = \{Y_1, \ldots, Y_N\}$ d.h. $P(\mathbf{X}, \mathbf{Y}) = P(X_1, \ldots, X_M, Y_1, \ldots, Y_N)$.

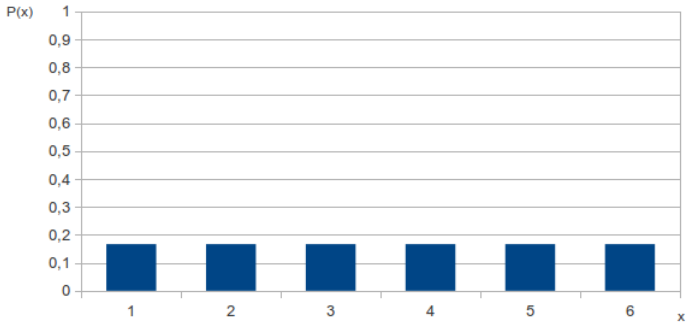

Abbildung 1: Diskrete uniforme Verteilung eines Würfels.

Die Produktregel definiert die Faktorisierung von $P(X, Y)$ wie folgt:

$$\begin{aligned} P(X, Y) =& P(X \mid Y)P(Y) \\ =& P(Y \mid X)P(X) \end{aligned}$$

$P(X \mid Y)$ und $P(Y \mid X)$ sind bedingte Wahrscheinlichkeiten (Conditional Probabilities) und $P(X)$ bzw. $P(Y)$ sind Randwahrscheinlichkeiten. Die Randwahrscheinlichkeiten können durch marginalisieren aus der Verbundwahrscheinlichkeit berechnet werden, d.h. $P(X) = \sum_y P(X, Y = y)$ bzw. $P(Y) = \sum_x P(X = x, Y)$. $P(X \mid Y)$ ist die Wahrscheinlichkeit von X unter der Bedingung, dass das Eintreten von Y bereits bekannt/gegeben ist.
Die Verbundwahrscheinlichkeitsverteilung für mehrere Variablen $X_1, \ldots, X_N$ faktorisiert anhand der Produktregel (Kettenregel) wie folgt:

$$\begin{aligned} P(\mathbf{X}) = P(X_1, \ldots, X_N) =& P(X_1)P(X_2 \mid X_1) \cdot \ldots \cdot P(X_N \mid X_{N-1}, \ldots, X_1) \\ =& P(X_N)P(X_{N-1} \mid X_N) \cdot \ldots \cdot P(X_1 \mid X_N, \ldots, X_2) \end{aligned}$$

Die Ordnung der Variablen in $\mathbf{X}$ kann bei der Kettenregel beliebig sein.
Aus dem Zusammenhang $P(X \mid Y)P(Y) = P(Y \mid X)P(X)$ der Produktregel ergibt sich direkt der Satz von Bayes

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}. \tag{1}$$

Bei statistisch unabhängigen Zufallsvariablen gilt, dass $P(X, Y) = P(X \mid Y)P(Y) = P(Y \mid X)P(X) = P(X)P(Y)$.
Wenn X statistisch unabhängig von Y ist (wird oft geschrieben als $X \perp Y$), dann enthält X keine Information über Y und umgekehrt, d.h. die bedingte Wahrscheinlichkeit $P(X \mid Y) = \frac{P(X,Y)}{P(Y)} = \frac{P(X)P(Y)}{P(Y)} = P(X)$ bzw. $P(Y \mid X) = P(Y)$.
Unter Randverteilung oder Marginalverteilung werden Wahrscheinlichkeitsverteilungen über eine Teilmenge von Zufallsvariablen bezeichnet.

- Wenn $\mathbf{X} = \{X_1, \ldots, X_M\}$ diskrete Zufallsvariablen sind, dann gilt $P(X_2, \ldots, X_M) = \sum_{x_1} P(X_1 = x_1, X_2, \ldots, X_M)$ oder $P(X_3, \ldots, X_M) = \sum_{x_1} \sum_{x_2} P(X_1 = x_1, X_2 = x_2, X_3, \ldots, X_M)$
  Das Summieren über die Wertebereiche von Variablen ist auch als Summenregel bekannt.

- Wenn $\mathbf{X} = \{X_1, \ldots, X_M\}$ kontinuierliche Zufallsvariablen sind, wird die Summe durch ein Integral ersetzt, $P(X_2, \ldots, X_M) = \int\limits_{-\infty}^{\infty} P(X_1, X_2, \ldots, X_M) dX_1$

2 Modellierung von Daten mit Wahrscheinlichkeitsverteilungen

Gegeben sind die Daten $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, wobei $\mathbf{x}_n$ ein d-dimensionaler Zufallsvektor ist, d.h.

$$\mathbf{x}_n = \begin{bmatrix} x_n^1 \\ \vdots \\ x_n^d \end{bmatrix}, \tag{2}$$

bzw. $\mathbf{x}_n \in \mathbb{R}^d$ und d beschreibt die Dimension des Vektors.
Ziel ist es, die Daten $\mathcal{X}$ mit einer Wahrscheinlichkeitsverteilung $P(\mathbf{x})$ zu modellieren.
Die Wahrscheinlichkeitsverteilung $P(\mathbf{x})$ kann parametrisch oder nicht-parametrisch dargestellt werden.

2.1 Nicht-parametrisches Modell

Im Gegensatz zum parametrischen Modell ist beim nicht-parametrischen Modell die Modellstruktur nicht a priori festgelegt, sondern wird aus den Daten bestimmt.

Beim parametrischen Modell ist die Annahme einer Wahrscheinlichkeitsverteilung für die Modellierung der Datenverteilung, z.B. Gauß, erforderlich. Beim nicht-parametrischen Modell ist die Art und Anzahl der Parameter flexibel und nicht von vornherein festgelegt.

Zur besseren Veranschaulichung nehmen wir an, dass unsere Daten in $\mathbb{R}^1$ sind. Die empirische Dichtefunktion ist eine Summe von Dirac-Impulsen $\delta(x)$ plaziert auf den Datenpunkten und normiert durch die Anzahl der Datenpunkte, d.h.

$$P^e(x) = \frac{1}{N} \sum_{n=1}^{N} \delta(x - x_n)$$

Durch die Normierung erhält man

$$\int_{-\infty}^{\infty} P^e(x) dx = 1.$$

Die Dirac Funktion $\delta(\cdot)$ ist definiert durch

$$\delta(x) = \begin{cases} +\infty & \text{wenn } x = 0 \\ 0 & \text{wenn } x \neq 0 \end{cases} \quad \text{und} \quad \int_{-\infty}^{\infty} \delta(x) dx = 1.$$

Die empirische Dichtefunktion in $\mathbb{R}^1$ für drei Samples $x_1 = 5, x_2 = 10$ und $x_3 = 15$ ist in Abbildung 2 dargestellt.

Das nicht-parametrische Modell benötigt $\mathcal{X}$ zur Repräsentation von $P^e(x)$. Die empirische Verteilung $P^e(x)$ hat maximale Auflösung, ist aber keine glatte Verteilungsfunktion. Um eine realitätsnähere Wahrscheinlichkeitsverteilung zu erhalten, wird in der Regel die Dichtefunktion mit einem geeigneten Kern (Glättungskern) geglättet. Eine gängige Wahl ist ein Gaußkern $h(x)$

$$h(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right).$$

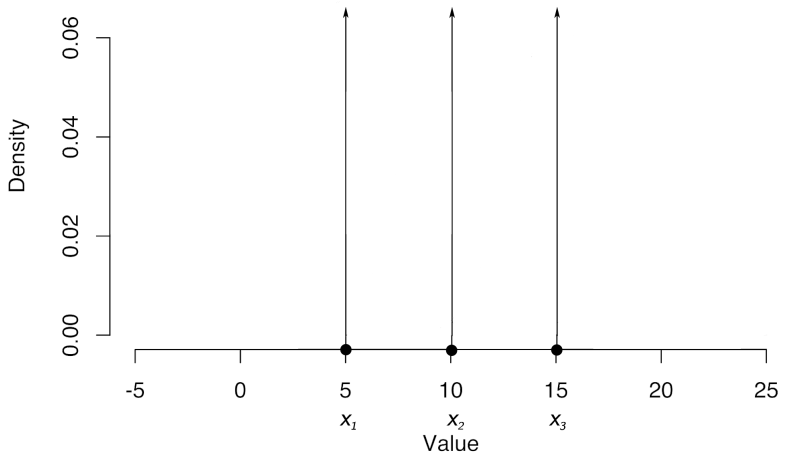

Abbildung 2: Empirische Dichtefunktion in $\mathbb{R}^1$ [Wikipedia, 2013].

Parameter σ bestimmt die Breite (Varianz) des Gaußkerns.

Die geglättete Dichtefunktion berechnet sich aus der Faltung der empirischen Dichtefunktion $P^e(x)$ mit dem Kern $h(x)$, d.h.

$$\begin{aligned}
P^{gl}(x) =& h(x) * P^e(x) \\
=& \int_{-\infty}^{\infty} h(x-\xi) P^e(\xi) d\xi \\
=& \int_{-\infty}^{\infty} h(x-\xi) \frac{1}{N} \sum_{n=1}^{N} \delta(\xi - x_n) d\xi \\
=& \frac{1}{N} \sum_{n=1}^{N} \int_{-\infty}^{\infty} h(x-\xi) \delta(\xi - x_n) d\xi \\
=& \frac{1}{N} \sum_{n=1}^{N} h(x - x_n).
\end{aligned}$$

Die geglättete Dichte ist eine normalisierte Summe von N Kernen mit Mittelwert x_n der gegebenen Daten. Das Beispiel in Abbildung 3 zeigt, eine geglättete Dichtefunktion. Auf jedem Datenpunkt x_n liegt ein Gaußkern. Die Varianz σ^2 bestimmt die Stärke der Glättung, d.h. bei großer Varianz ist die Glättung stark und bei sehr kleiner Varianz ist die geglättete Dichtefunktion ähnlich der empirischen Dichtefunktion.

2.2 Parametrisches Modell

Bei einem parametrischen Ansatz trifft man in der Regel eine a-priori Annahme einer Verteilungsfunktion zur Modellierung der Daten, d.h. $\mathbf{x} \sim P(\mathbf{x} \mid \mathbf{\Theta})$. Diese Vereilungsfunktion ist durch die Parameter $\mathbf{\Theta}$ spezifiziert.
Wir verwenden zum Beispiel das Modell einer multivariaten Gaußverteilung.

$$P(\mathbf{x} \mid \mathbf{\Theta}) = \mathcal{N}(\mathbf{x} \mid \mathbf{\Theta}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right),$$

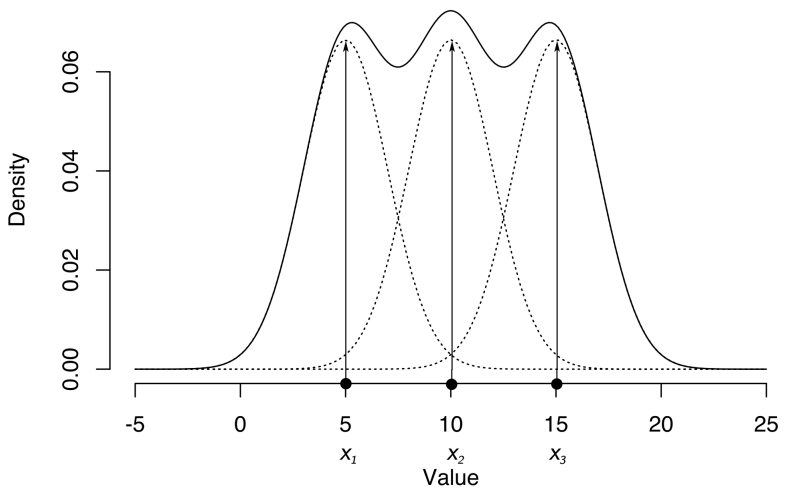

Abbildung 3: Geglättete nicht-parametrische Dichtefunktion [Wikipedia, 2013].

wobei $|\Sigma|$ die Determinante der Kovarianzmatrix ist. Die Parameter der Gaußverteilung sind $\boldsymbol{\Theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, wobei $\boldsymbol{\mu}$ der Mittelwertvektor und $\boldsymbol{\Sigma}$ die Kovarianzmatrix mit der Dimension $d \times d$ ist.
Jede Kovarianzmatrix Σ erfüllt folgende Eigenschaften: Sie ist symmetrisch, d.h. $\Sigma = \Sigma^T$ und positiv semidefinit (alle Eigenwerte von $\Sigma \geq 0$), d.h. $\mathbf{x}^T \Sigma \mathbf{x} \geq 0, \forall \mathbf{x} \neq 0$.
Im folgenden Beispiel wird die Wahrscheinlichkeitsdichte einer Gaußverteilung mit drei verschiedenen Kovarianzmatrizen im 2-dimensionalen Raum, d.h. $d = 2$ dargestellt. Punkte auf den Kreisen bzw. Ellipsen haben die gleiche Wahrscheinlichkeit bzw. Likelihood.

(a) In Abbildung 4 ist die Wahrscheinlichkeitsdichte einer Gaußverteilung mit skalierter Einheitsmatrix $a \cdot \mathbf{I}$ dargestellt.
$\Sigma = a \cdot \mathbf{I} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$
Die Varianz ist in beiden Koordinatenrichtungen gleich groß. Dies ist an den konzentrischen Kreisen um den Mittelwert ersichtlich.

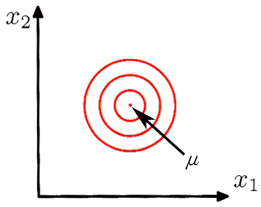

Abbildung 4: Gaußverteilung mit skalierter Einheitsmatrix als Kovarianzmatrix in $\mathbb{R}^2$ [Bishop, 2007].

(b) Als nächstes wird die Wahrscheinlichkeitsdichte einer Gaußverteilung mit diagonaler Kovarianzmatrix aber unterschiedlichen Varianzen in der Diagonale dargestellt.
$\Sigma = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$
Die Wahrscheinlichkeitsdichte ist in Abbildung 5 visualisiert. Die Hauptachsen der Ellipsen sind am Koordinatensystem ausgerichtet. Weiters ist ersichtlich, dass die Varianz in x_1 größer als in x_2 ist (d.h. $a > b$). Eine diagonale Kovarianzmatrix bedeutet das die einzelnen Dimensionen in den Daten

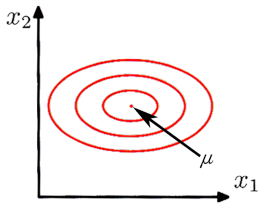

Abbildung 5: Gaußverteilung mit diagonaler Kovarianzmatrix in $\mathbb{R}^2$ [Bishop, 2007].

nicht korreliert sind.

(c) Als nächstes wollen wir uns eine Kovarianzmatrix allgemeiner Form ansehen.
$\Sigma = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$
Die Wahrscheinlichkeitsdichte ist in Abbildung 6 dargestellt. Die Hauptachsen der Ellipsen sind nicht am Koordinatensystem ausgerichtet, d.h. die einzelnen Dimensionen in den Daten sind korreliert.

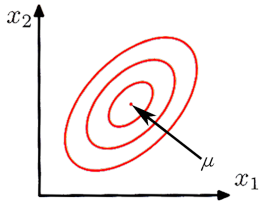

Abbildung 6: Gaußverteilung mit allgemeiner Kovarianzmatrix in $\mathbb{R}^2$ [Bishop, 2007].

2.2.1 Maximum-Likelihood (ML) Schätzer

Die Parameter des parametrischen Modells können aus Daten $\mathcal{X}$ mit den ML-Schätzer geschätzt werden. Gegeben sind die Daten $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Ziel ist es die Parameter $\mathbf{\Theta}$ des parametrischen Modells zu schätzen. Bei der Maximum-Likelihood Methode werden die Parameter $\mathbf{\Theta}$ so geschätzt, dass die Likelihood-Funktion maximal wird, dabei sind die Daten $\mathcal{X}$ fixiert und die zu schätzenden Parameter $\mathbf{\Theta}$ variieren.
Die Likelihood-Funktion ist definiert durch:

$$P(\mathcal{X} \mid \mathbf{\Theta}) = P(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \mathbf{\Theta}) = P(\mathbf{x}_1 \mid \mathbf{\Theta}) P(\mathbf{x}_2 \mid \mathbf{x}_1, \mathbf{\Theta}) \cdots P(\mathbf{x}_N \mid \mathbf{x}_{N-1,\ldots,N_1}, \mathbf{\Theta})$$

Wenn die Samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ independent identically distributed (**iid**[1]) sind, können wir die Likelihood-Funktion über das Produkt der Likelihoods aller $\mathbf{x}_n$ bestimmen, d.h.

$$P(\mathcal{X} \mid \mathbf{\Theta}) = \prod_{n=1}^{N} P(\mathbf{x}_n \mid \mathbf{\Theta}).$$

Im nächsten Schritt kann die Likelihood-Funktion logarithmiert werden. Es ist aber nicht zwingend notwendig. Das Logarithmieren hat den Vorteil, dass anstelle des Produktes über die Samples die Summe

[1] iid heißt, dass die Samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ statistisch unabhängig sind und von der gleichen Wahrscheinlichkeitsverteilung stammen.

über die Samples verwendet werden kann. Damit werden numerische Probleme bei sehr großen N vermieden.
Die log-Likelihood-Funktion ist:

$$\begin{aligned} L(\mathcal{X} \mid \boldsymbol{\Theta}) &= L(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\Theta}) = \ln P(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\Theta}) \\ &= \ln \left[\prod_{n=1}^{N} P(\mathbf{x}_n \mid \boldsymbol{\Theta}) \right] \\ &= \sum_{n=1}^{N} \ln(P(\mathbf{x}_n \mid \boldsymbol{\Theta})) \end{aligned}$$

Bei der Maximum-Likelihood Methode sind die Parameter $\boldsymbol{\Theta}$ mit der maximalen $L(\mathbf{X} \mid \boldsymbol{\Theta})$ von Interesse, d.h. wir benötigen das Argument $\boldsymbol{\Theta}$, welches die log-Likelihood maximiert:

$$\boldsymbol{\Theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\Theta}} L(\mathcal{X} \mid \boldsymbol{\Theta}).$$

Um nun diese Parameter $\boldsymbol{\Theta}_{\mathrm{ML}}$ zu bestimmen, wird die log-Likelihood-Funktion nach $\boldsymbol{\Theta}$ abgeleitet und die Ableitung wird auf 0 gesetzt

$$\frac{\partial L(\mathcal{X} \mid \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \stackrel{!}{=} \mathbf{0}.$$

Zum Beispiel im Falle einer Gaußverteilung ist die log-Likelihood-Funktion:

$$L(\mathcal{X} \mid \boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln\left(\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\Theta})\right) \tag{3}$$

Durch Ableiten von (3) nach den Parametern $\boldsymbol{\mu}$ und $\boldsymbol{\Sigma}$ und auf Null setzen der Ableitung erhalten wir für $\boldsymbol{\Theta}_{\mathrm{ML}} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ folgende Ergebnisse:

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \\ \boldsymbol{\Sigma} &= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T \end{aligned}$$

2.2.2 Bayes Schätzer

Im Unterschied zum ML-Schätzer, wo $\boldsymbol{\Theta}$ deterministisch und unbekannt ist, wird beim Bayes Schätzer $\boldsymbol{\Theta}$ als Zufallsvariable modelliert, d.h. für $\boldsymbol{\Theta}$ existiert eine Wahrscheinlichkeitsdichtefunktion, die sogenannte a-priori Verteilung. Für den Bayes Schätzer wird der Satz von Bayes verwendet. Die Verteilung von $\boldsymbol{\Theta}$ bei gegebenen Daten $\mathcal{X}$ ist also:

$$\underbrace{P(\boldsymbol{\Theta} \mid \mathcal{X})}_{\text{posterior Wahrscheinlichkeitsverteilung}} = \frac{P(\overbrace{\mathcal{X} \mid \boldsymbol{\Theta}}^{\text{Likelihood; a-priori Wahrscheinlichkeitsverteilung}} \overbrace{P(\boldsymbol{\Theta})}^{\text{Likelihood; a-priori Wahrscheinlichkeitsverteilung}}}{P(\mathcal{X})} \tag{4}$$

Der Bayes Schätzer modelliert die Modellparameter $\boldsymbol{\Theta}$ als Wahrscheinlichkeitsverteilung über die gegebenen Daten $\mathcal{X}$.
Ein Spezialfall beim Bayes Schätzer ist der Maximum-a-posteriori (MAP) Schätzer. Dabei werden jene Parameter genommen, die die posterior Wahrscheinlichkeit $P(\boldsymbol{\Theta} \mid \mathcal{X})$ maximieren, d.h.:

$$\boldsymbol{\Theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\Theta}} P(\boldsymbol{\Theta} \mid \mathcal{X}) = \arg\max_{\boldsymbol{\Theta}} \left[P(\mathcal{X} \mid \boldsymbol{\Theta}) P(\boldsymbol{\Theta})\right]$$

$P(\mathcal{X})$ in (4) kann hier vernachlässigt werden, da es sich nur um einen Skalierungsfaktor handelt, welcher das Ergebnis nicht beeinflusst. Ist die prior Wahrscheinlichkeitsverteilung $P(\mathbf{\Theta})$ uniform, dann handelt es sich um eine nicht-informative a-priori Wahrscheinlichkeitsverteilung. Die MAP Parameter für diesen Fall sind $\mathbf{\Theta}_{\text{MAP}} = \mathbf{\Theta}_{\text{ML}}$. Die posterior Wahrscheinlichkeitsverteilung $P(\mathbf{\Theta} \mid \mathcal{X}) = f(\mathbf{\Theta} \mid \mathbf{y})$ ist in Abbildung 7 dargestellt. Als eine Alternative zum MAP Schätzer wird auch der Median oder der Mittelwert von $P(\mathbf{\Theta} \mid \mathcal{X})$ herangezogen. Der Median beim Bayes Schätzer liefert in vielen Fällen ein besseres Ergebnis als der MAP Schätzer.

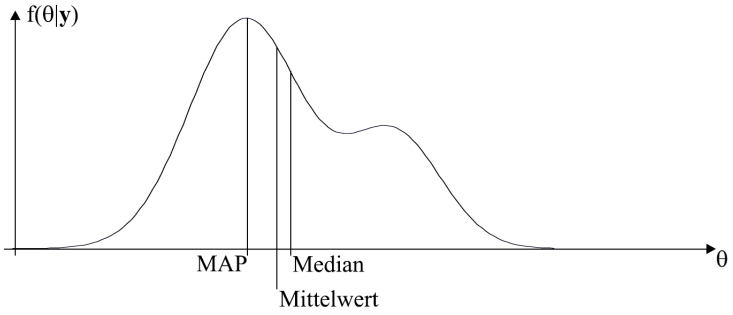

Abbildung 7: Posterior Wahrscheinlichkeitsverteilung von Bayes Schätzer [Köhler, 2005].

3 Bayes Klassifikator

Ziel der Klassifikation ist es, Objekte, beschrieben durch Features $\mathbf{x} \in \mathbb{R}^d$, einer Klasse zu zuweisen. Dafür kann die Bayes Regel verwenden werden.
Gegeben ist:

$$\begin{aligned}
&\mathcal{X} = \{\langle \mathbf{x}_1, t_1 \rangle, \ldots, \langle \mathbf{x}_N, t_N \rangle\} \\
&\mathbf{x} \in \mathbb{R}^d \\
&t \in \mathbb{N} \qquad t \in \{1, 2, \ldots, c\} \\
&c \ldots \text{Anzahl der Klasse} \\
&N \ldots \text{Anzahl der Samples}
\end{aligned}$$

Die Klassifikation erfolgt anhand der Wahrscheinlichkeit für Klasse t gegeben die Objektbeschreibung $\mathbf{x}$, d.h. $P(t \mid \mathbf{x})$ ist notwendig. Wenn nun beispielsweise bei einem 2-Klassenproblem $P(t = 1 \mid \mathbf{x}) > P(t = 2 \mid \mathbf{x})$ dann wählt man Klasse 1. Diese posterior Wahrscheinlichkeit kann über den Satz von Bayes formuliert werden:

$$\underbrace{P(t|\mathbf{x})}_{\text{posterior Wahrscheinlichkeit}} = \frac{\overbrace{P(\mathbf{x}|t)}^{\text{Likelihood; prior Wahrscheinlichkeit}} \quad \overbrace{P(t)}}{\underbrace{\sum_{t'} P(\mathbf{x}|t')P(t')}_{\sum_{t'} P(\mathbf{x}|t')P(t') = \sum_{t'} P(\mathbf{x},t') = P(\mathbf{x})}}$$

Dabei ist $P(\mathbf{x})$ ein Skalierungsfaktor (d.h. $P(\mathbf{x})$ skaliert $P(t \mid \mathbf{x})$ sodass $\sum_t P(t \mid \mathbf{x}) = 1$) welcher für alle Klassen t gleich groß ist. Um die Klassenzugehörigkeit bestimmen zu können, wählen wir jene Klasse t^* mit der größten posterior Wahrscheinlichkeit, d.h. $P(t^* \mid \mathbf{x}) > P(t \mid \mathbf{x}) \quad \forall t \neq t^*$.

Dies kann auch wie folgt formuliert werden:

$$
\begin{aligned}
t^* &= \arg\max_t P(t|\mathbf{x}) = \arg\max_t \left[P(\mathbf{x}|t)P(t)\right] \\
&= \arg\max_t \left[\underbrace{\ln P(\mathbf{x}|t) + \ln P(t)}_{g_t(\mathbf{x})\ldots\ \text{Entscheidungsfunktion}} \right]
\end{aligned}
$$

wobei $g_t(\mathbf{x})$ die Entscheidungsfunktion für die Klasse t ist.

3.1 Beispiel: Bayes Klassifikator mit 2 Klassen

Wir nehmen an, dass die Samples für beide Klassen gaußverteilt sind, d.h. $P(\mathbf{x}_n|t) = \mathcal{N}(\mathbf{x}_n, \mid \mu_t, \Sigma_t)$. Da eine Gaußverteilung verwendet wird, handelt es sich um ein parametrisches Modell.
Weiters sind die Objekte $\mathbf{x}_n \in \mathbb{R}^2$ mit 2 Merkmalen beschrieben. In Abbildung 8 sind die Daten für eine 2-Klassen-Verteilung dargestellt. Die Daten werden durch 2 Merkmale: Width und Lightness, d.h.

$$
\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} Width \\ Lightness \end{bmatrix}.
$$

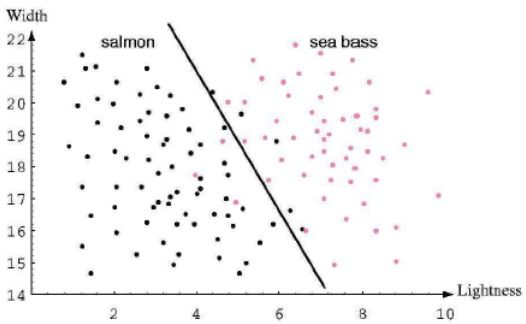

Abbildung 8: 2 Klassen Klassifikationsproblem [Duda et al., 2001].

Zuerst muss der Klassifikator trainiert werden. Dabei werden die Daten $\mathcal{X} = \{\langle \mathbf{x}_1, t_1\rangle, \ldots, \langle \mathbf{x}_n, t_n\rangle\}$ verwendet. Beim Training werden die Parameter, in unserem Fall $\boldsymbol{\mu}_{t=1}, \boldsymbol{\Sigma}_{t=1}, \boldsymbol{\mu}_{t=2}, \boldsymbol{\Sigma}_{t=2}, P(t=1)$ und $P(t=2)$ bestimmt.
Zuerst werden die Daten $\mathcal{X}$ anhand ihrer Klassenzugehörigkeit aufgeteilt, wobei $\mathcal{X}^i$ die Menge der Samples mit $t = i$ ist, d.h.

$$
\begin{aligned}
\text{Klasse } t = 1: &\qquad \mathcal{X}^1 = \{\mathbf{x}_n | t_n = 1\} \\
t = 2: &\qquad \mathcal{X}^2 = \{\mathbf{x}_n | t_n = 2\}.
\end{aligned}
$$

Von den Daten $\mathcal{X}^i$ für die Klassen t^i können über das maximum likelihood Schätzverfahren (siehe Kapitel 2.2.1) die Parameter der Gaußverteilung geschätzt werden, d.h. $\boldsymbol{\Theta}_1 = \{\boldsymbol{\mu}_{t=1}, \boldsymbol{\Sigma}_{t=1}\}, \boldsymbol{\Theta}_2 = \{\boldsymbol{\mu}_{t=2}, \boldsymbol{\Sigma}_{t=2}\}$. Weiters muss noch die Priorverteilung $P(t)$ für die Klassen geschätzt werden $P(t=i) = \frac{|\mathcal{X}^i|}{|\mathcal{X}|}$, wobei $|\mathcal{X}|$ die Anzahl der Samples in der Menge $\mathcal{X}$ bedeutet.

3.2 Entscheidungsfunktion für die Normalverteilung

Im obigen Beispiel wurde für den Likelihood-Term $P(\mathbf{x} \mid t)$ die Normalverteilung verwendet:

$$
\mathcal{N}(\mathbf{x} \mid \boldsymbol{\Theta}_t) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = \frac{1}{|2\pi|^{\frac{d}{2}} |\boldsymbol{\Sigma}_t|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1} (\mathbf{x} - \boldsymbol{\mu}_t)\right] \tag{5}
$$

Diese Likelihood-Funktion $P(\mathbf{x} \mid t)$ könnte aber auch durch eine andere Verteilung z.B. eine Gaußsche Mischverteilung (GMM) oder einem Hidden Markov Modell (HMM) modelliert werden (siehe Kapitel 4 und Kapitel 7).
Die Entscheidungsfunktion $g_t(\mathbf{x})$ für $P(\mathbf{x} \mid t) = \mathcal{N}(\mathbf{x} \mid \mathbf{\Theta}_t)$ ist:

$$g_t(\mathbf{x}) = \ln P(t) + \ln P(\mathbf{x} \mid \mathbf{\Theta}_t) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_t)^T \mathbf{\Sigma}_t^{-1}(\mathbf{x} - \boldsymbol{\mu}_t) - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{\Sigma}_t| + \ln P(t) \tag{6}$$

Nachfolgend möchten wir 3 Fälle mit verschiedenen Kovarianzmatrizen (siehe Kapitel 2.2) ableiten.

- Fall 1: Gleiche und sphärische Kovarianzmatrix für alle Klassen, d.h.

$$\Sigma_t = \sigma^2 \mathbf{I} \quad \forall t. \tag{7}$$

Unter Berücksichtigung von (7) in der Entscheidungsfunktion (6) erhält man

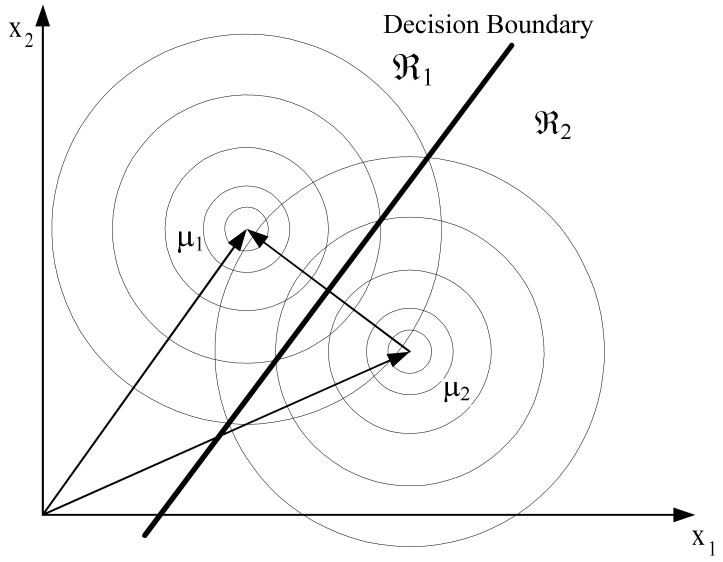

Abbildung 9: Lineare Entscheidungsgrenze bei gleichen sphärischen Kovarianzmatrizen [Duda et al., 2001].

$$g_t(\mathbf{x}) = -\frac{\overbrace{(\mathbf{x} - \boldsymbol{\mu}_t)^T(\mathbf{x} - \boldsymbol{\mu}_t)}^{euklidische Distanz}}{2\sigma^2} + \ln P(t). \tag{8}$$

Wir stellen fest, dass die Entscheidung aufgrund der euklidischen Distanz und $P(t)$ getroffen wird. Die Entscheidungsgrenze zwischen Klasse t und Klasse i ist definiert als: $g_t(\mathbf{x}) = g_i(\mathbf{x}), i \neq t$, d.h. $P(i|\mathbf{x}) = P(t|\mathbf{x})$. Die Entscheidungsgrenze ist in diesem Fall linear und in Abbildung 9 dargestellt. Wenn $P(t)$ für alle Klassen gleich ist (uniforme Verteilung), dann kann $P(t)$ in (8) und σ vernachlässigt werden. In diesem Fall wird nur Anhand der Distanz klassifiziert, d.h. jene Klasse mit Mittelwert $\boldsymbol{\mu}_t$ wird gewählt welche die minimale Distanz zu $\mathbf{x}$ hat. In diesem Fall ist die Entscheidungsfunktion $g_t(\mathbf{x}) = -(\mathbf{x} - \boldsymbol{\mu}_t)^T(\mathbf{x} - \boldsymbol{\mu}_t)$.

- Fall 2: Gleiche Kovarianzmatrix für alle Klassen, d.h.

$$\mathbf{\Sigma}_t = \mathbf{\Sigma}, \quad \forall t \tag{9}$$

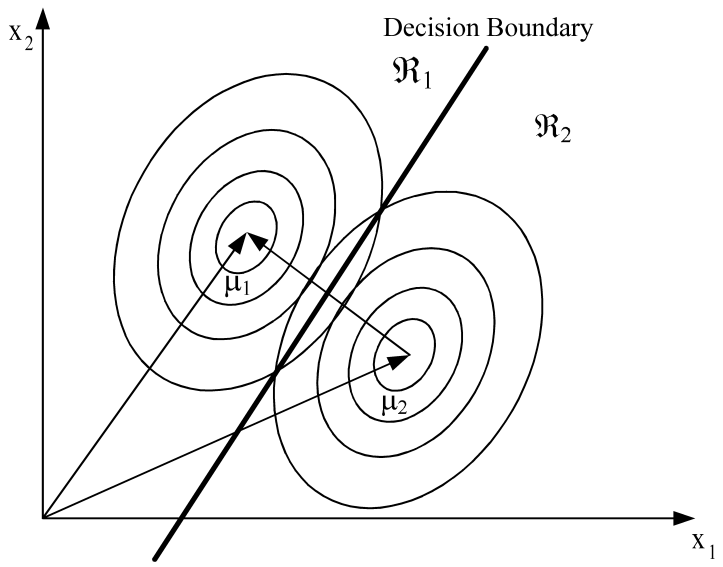

Abbildung 10: Lineare Entscheidungsgrenze bei gleichen Kovarianzmatrizen [Duda et al., 2001].

Unter Berücksichtigung von (9) vereinfacht sich die Entscheidungsfunktion in (6) zu:

$$g_t(\mathbf{x}) = -\frac{1}{2}\overbrace{(\mathbf{x}-\boldsymbol{\mu}_t)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_t)}^{\text{Mahalanobis Distanz}} + \ln P(t)$$

Die Entscheidungsgrenze ist linear und in Abbildung 10 dargestellt. Die Entscheidung wird aufgrund der Mahalanobis Distanz und $P(t)$ getroffen. Die Mahalanobis Distanz bestimmt die Entfernung zweier Punkte im mehrdimensionalen Raum unter Berücksichtigung der Kovarianzmatrix.

- Fall 3: Beliebige Kovarianzmatrix für die Klassen; $\mathbf{\Sigma}_t$ ist für jede Klasse unterschiedlich. In diesem Fall führt es zu folgender Entscheidungsfunktion:

$$\begin{aligned} g_t(\mathbf{x}) &= \ln P(t) + \ln P(\mathbf{x} \mid \mathbf{\Theta}_t) \\ &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_t)^T\mathbf{\Sigma}_t^{-1}(\mathbf{x}-\boldsymbol{\mu}_t) - \frac{1}{2}\ln|\mathbf{\Sigma}_t| + \ln P(t) \end{aligned}$$

Die Entscheidungsgrenzen sind hyperquadratische Funktionen. In Abbildung 11 sehen wir unterschiedliche Entscheidungsgrenzen. Daten die im schraffierten Bereich liegen, werden zur Klasse $t = 2$ zugewiesen. Die Kovarianzmatrix für beide Klassen sind als Ellipsen bzw. Kreise dargestellt.

4 Gaußsche Mischverteilung / Gaussian Mixture Modell (GMM)

Unter einer Gaußschen Mischverteilung versteht man eine gewichtete Summe von Gaußverteilungen. Man verwendet Gaußsche Mischverteilungen zum modellieren komplexer multimodaler Verteilungsfunktionen (siehe Abbildung 12). Multimodal bedeutet, dass die Verteilung mehr als ein Maximum aufweist.
Die Wahrscheinlichkeitsdichtefunktion ist gegeben als:

$$p(\mathbf{x} \mid \Theta) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\mathbf{x} \mid \mathbf{\Theta}_m),$$

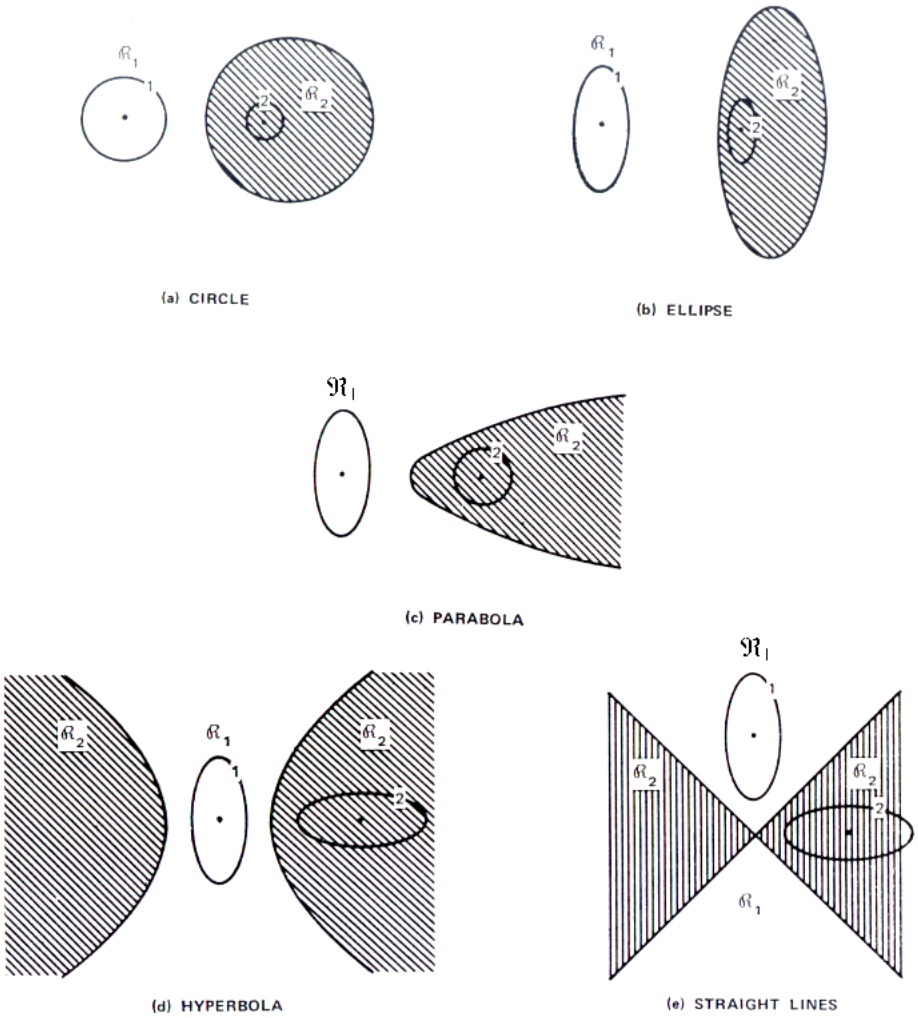

Abbildung 11: Hyperquadratische Entscheidungsgrenzen [Duda et al., 2001].

wobei $\mathbf{\Theta}_m = \{\boldsymbol{\mu}_m, \mathbf{\Sigma}_m\}$. Die Parameter der Verteilung sind $\mathbf{\Theta} = \{\alpha_1, \ldots, \alpha_M, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_M\}$, wobei M die Anzahl der Gauß-Komponenten ist. Der Parameter $\alpha_m = P(m)$ gewichtet die einzelnen Gaußkomponenten. Für die Wahrscheinlichkeit α_m gilt $0 \leq \alpha_m \leq 1$ und $\sum_{m=1}^{M} \alpha_m = 1$. Die Skalierung von $\mathcal{N}(\mathbf{x} \mid \mathbf{\Theta})$ durch α_m garantiert, dass $\int_{-\infty}^{\infty} p(\mathbf{x} \mid \mathbf{\Theta}) d\mathbf{x} = 1$.

4.1 Schätzen der Parameter Θ

Die Gaußsche Mischverteilung ist ein parametrisches Modell. Die Schätzung der Parameter $\mathbf{\Theta}$ kann durch die Maximum-Likelihood (ML) Methode (siehe Kapitel 2.2.1) erfolgen.

$$\begin{aligned} \text{Gegeben sind die Daten:} \quad & \mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}, \quad \mathbf{x} \in \mathbb{R}^d \\ \text{Gesucht sind die Parameter:} \quad & \mathbf{\Theta}_{\text{ML}} = \arg\max_{\mathbf{\Theta}} \{\ln P(\mathcal{X} \mid \mathbf{\Theta})\} \end{aligned}$$

Im ersten Schritt wird die log-Likelihood Funktion $L(\mathcal{X} \mid \mathbf{\Theta}) = \ln P(\mathcal{X}|\mathbf{\Theta})$ für die Gaußsche Mischverteilung formuliert. Anschließend wird der stationäre Punkt für $L(\mathcal{X} \mid \mathbf{\Theta})$ gesucht: $\frac{\partial \ln P(\mathcal{X}|\mathbf{\Theta})}{\partial \mathbf{\Theta}} \stackrel{!}{=} 0$.
Unter der Annahme, dass die Samples/Datenpunkte $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ **iid** sind ergibt sich folgende

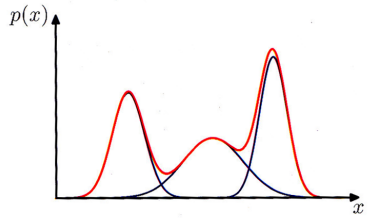

Abbildung 12: Gaußsche Mischverteilung [Bishop, 2007].

log-Likelihood Funktion

$$L(\mathcal{X} \mid \boldsymbol{\Theta}) = \ln P(\mathcal{X} \mid \boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln P(\mathbf{x}_n \mid \boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln \sum_{m=1}^{M} \alpha_m \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m). \tag{10}$$

Im letzten Schritt wird für $P(\mathbf{x}_n \mid \boldsymbol{\Theta})$ die Gaußsche Mischverteilung eingesetzt. Als nächstes werden die einzelnen Ableitung für die Parameter $\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, \boldsymbol{\alpha}_m$ formuliert und auf 0 gesetzt.

4.1.1 Ableitung für den Mittelwert $\boldsymbol{\mu}_m$

Im ersten Schritt müssen wir die log-Likelihood Funktion $L(\mathcal{X} \mid \boldsymbol{\Theta})$ in(10) nach $\boldsymbol{\mu}_m$ ableiten:

$$\frac{\partial \ln P(\mathcal{X}|\boldsymbol{\Theta})}{\partial \boldsymbol{\mu}_m} = \sum_{n=1}^{N} \frac{1}{\sum\limits_{m'=1}^{M} \alpha_{m'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{m'}, \boldsymbol{\Sigma}_{m'})} \frac{\partial \sum\limits_{m=1}^{M} \alpha_m \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\partial \boldsymbol{\mu}_m}$$

$$= \sum_{n=1}^{N} \underbrace{\frac{\alpha_m \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum\limits_{m'=1}^{M} \alpha_{m'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{m'}, \boldsymbol{\Sigma}_{m'})}}_{r_m^n} \frac{\partial \left[\ln(\alpha_m) + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\right]}{\partial \boldsymbol{\mu}_m}. \tag{11}$$

Die Wahrscheinlichkeit $r_m^n = P(m|\mathbf{x}_n, \Theta)$ ist die posterior Wahrscheinlichkeit für Komponente m gegeben $\mathbf{x}_n$ und die Parameter.
Die Ableitung der Normalverteilung (siehe (5)) ergibt:

$$\frac{\partial \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\partial \boldsymbol{\mu}_m} = -\frac{1}{2} \left(\boldsymbol{\Sigma}_m^{-1} + (\boldsymbol{\Sigma}_m^{-1})^T\right) (\mathbf{x}_n - \boldsymbol{\mu}_m),$$

wobei $(\boldsymbol{\Sigma}_m^{-1} + (\boldsymbol{\Sigma}_m^{-1})^T) = 2\boldsymbol{\Sigma}_m^{-1}$, da $\boldsymbol{\Sigma}$ symetrisch ist, d.h. $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T$. Daraus folgt

$$\frac{\partial \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\partial \boldsymbol{\mu}_m} = -\boldsymbol{\Sigma}_m^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_m). \tag{12}$$

Einsetzen von (12) in (11) ergibt

$$\frac{\partial \ln P(\mathbf{X}|\Theta)}{\partial \boldsymbol{\mu}_m} = -\sum_{n=1}^{N} r_m^n \boldsymbol{\Sigma}_m^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_m).$$

Im nächsten Schritt setzen wir die Ableitung auf 0 und multiplizieren beide Seiten mit $\boldsymbol{\Sigma}_m$

$$\sum_{n=1}^{N} (r_m^n \mathbf{x}_n - r_m^n \boldsymbol{\mu}_m) \stackrel{!}{=} 0.$$

Nach weiterer Umformunlierung erhalten wir die Formel zum berechnen von $\boldsymbol{\mu}_m$

$$\begin{aligned}\boldsymbol{\mu}_m \sum_{n=1}^{N} r_m^n &= \sum_{n=1}^{N} r_m^n \mathbf{x}_n \\ \boldsymbol{\mu}_m &= \frac{\sum\limits_{n=1}^{N} r_m^n \mathbf{x}_n}{\underbrace{\sum\limits_{n=1}^{N} r_m^n}_{N_m}}.\end{aligned}$$

Der Mittelwert berechnet sich aus den mit der Zugehörigkeitswahrscheinlichkeit r_m^n gewichteten Daten $\mathbf{x}_n$. Wir bezeichnen $N_m = \sum\limits_{n=1}^{N} r_m^n$ als die effektive Anzahl von Datenpunkten die von Komponente m modelliert werden. Zur Berechnung von $\boldsymbol{\mu}_m$ braucht man r_m^n, dass wiederum von $\boldsymbol{\Theta}$ abhängt. Dies führt zur klassischen Henne-Ei-Problematik. Die Konsequenz daraus ist, dass die Berechnung von $\boldsymbol{\mu}_m$ iterativ bewerkstelligt wird, d.h. man initialisiert $\boldsymbol{\Theta}$. Dies ermöglicht die iterative Berechnung von r_m^n und $\boldsymbol{\mu}_m$.

4.1.2 Ableitung für $\boldsymbol{\Sigma}_m$

Nun wird $L(\mathcal{X} \mid \boldsymbol{\Theta})$ nach $\boldsymbol{\Sigma}_m$ abgeleitet und die Ableitung anschließend auf 0 gesetzt:

$$\frac{\partial \ln P(\mathbf{X} \mid \boldsymbol{\Theta})}{\partial \boldsymbol{\Sigma}_m} \stackrel{!}{=} 0.$$

Diese Ableitung ist in [Bishop, 2007] zu finden. Die Lösung ist:

$$\boldsymbol{\Sigma}_m = \frac{1}{N_m} \sum_{n=1}^{N} r_m^n (\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^T$$

Es wird wieder die posterior Verteilung r_m^n zur Gewichtung der Daten $\mathcal{X}$ benötigt. Für nur eine Gaußverteilung d.h. $M = 1$ würde sich eine Zugehörigkeitswahrscheinlichkeit $r_m^n = P(m \mid \mathbf{x}_n, \boldsymbol{\Theta}) = 1$ ergeben, womit sich unsere Parameter $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ wie folgt berechnen lässt:

$$\begin{aligned}\boldsymbol{\mu} &= \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \\ \boldsymbol{\Sigma} &= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T\end{aligned}$$

Wir erhalten die ML-Lösung einer Gaußverteilung wie sie im Kapitel 2.2.1 gezeigt wurde.

4.1.3 Ableitung für α_m

Weiters müssen wir $L(\mathcal{X} \mid \boldsymbol{\Theta})$ noch nach α_m ableiten und auf 0 setzen:

$$\frac{\partial \ln P(\mathcal{X} \mid \boldsymbol{\Theta})}{\partial \alpha_m} \stackrel{!}{=} 0.$$

Hier handelt es sich um ein Optimierungsproblem in α mit der Nebenbedingung $\sum\limits_{m=1}^{M} \alpha_m = 1$.
Die Lösung erhalten wir über Lagrange Multiplikatoren (siehe Anhang A). Dabei wird die Lagrange

Funktion aus Kombination der log-Likelihood Funktion $L(\mathcal{X} \mid \mathbf{\Theta})$ mit der Nebenbedingung wie folgt konstruiert

$$J(m) = \ln P(\mathcal{X} \mid \mathbf{\Theta}) + \lambda \left(\sum_{m=1}^{M} \alpha_m - 1 \right). \tag{13}$$

Die Lagrange Funktion $J(m)$ besteht aus der log-Likelihood Funktion plus einem Term bestehend aus dem Lagrangemultiplikator λ und der Bedingung $\sum\limits_{m=1}^{M} \alpha_m = 1$. Die Funktion $J(m)$ wird nach α_m abgeleitet. Als erstes berechnen wir die Ableitung von $L(\mathcal{X} \mid \mathbf{\Theta})$ nach α_m, d.h.

$$\frac{\partial \ln P(\mathcal{X} \mid \mathbf{\Theta})}{\partial \alpha_m} = \sum_{n=1}^{N} \frac{1}{\sum\limits_{m'=1}^{M} \alpha_{m'} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{m'}, \mathbf{\Sigma}_{m'})} \frac{\partial \sum\limits_{m=1}^{M} \alpha_m \mathcal{N}(\mathbf{x}_n \mid \mathbf{\Sigma}_m, \boldsymbol{\mu}_m)}{\partial \alpha_m}$$

$$= \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_m, \mathbf{\Sigma}_m)}{\sum\limits_{m'=1}^{M} \alpha_{m'} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{m'}, \mathbf{\Sigma}_{m'})} \tag{14}$$

Die Ableitung von $J(m)$ nach α_m erhalten wir durch (14) und der Ableitung der Bedingung $\sum\limits_{m=1}^{M} \alpha_m = 1$, d.h.

$$\frac{\partial J(m)}{\partial \alpha_m} = \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_m \mathbf{\Sigma}_m)}{\sum\limits_{m'=1}^{M} \alpha_{m'} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_{m'} \mathbf{\Sigma}_{m'})} + \lambda.$$

Nun setzten wir die Ableitung auf 0

$$\frac{\partial J(m)}{\partial \alpha_m} \stackrel{!}{=} 0$$

und multiplizieren beide Seiten der Gleichung mit α_m. Somit erhalten wir folgende Formel

$$\sum_{n=1}^{N} r_m^n + \lambda \alpha_m \stackrel{!}{=} 0 \tag{15}$$

Anschließend summieren wir auf beiden Seiten über m Komponenten und setzen $N_m = \sum\limits_{n=1}^{N} r_m^n$. Wir erhalten folgende Gleichung:

$$\sum_{m=1}^{M} N_m + \sum_{m=1}^{M} \lambda \alpha_m = 0$$

Da $\lambda \sum\limits_{m=1}^{M} \alpha_m = \lambda$ unter der Bedingung $\sum\limits_{m=1}^{M} \alpha_m = 1$ und $\sum\limits_{m=1}^{M} N_m = N$ ist folgt daraus, dass $\lambda = -N$. Wir setzen $\lambda = -N$ in (15) ein und erhalten

$$N_m - N\alpha_m = 0$$
$$\alpha_m = \frac{N_m}{N}.$$

Ähnlich wie bei der Ableitung für $\boldsymbol{\mu}_m$ und $\boldsymbol{\Sigma}_m$ ist auch hier immer r_m^n notwendig, da durch r_m^n immer alle Daten gewichtet zu den Updates beitragen. Dies resultiert in einem iterativen Algorithmus zur Schätzung von $\boldsymbol{\Theta}_{\mathrm{ML}}$. Dieser Algorithmus ist als Expectation Maximization für GMMs bekannt und wird im folgenden Kapitel vorgestellt.

4.2 Expectation-Maximization (EM) Algorithmus zum Lernen von Gaußschen Mischverteilungen

Der EM Algorithmus zum Lernen von GMMs ist iterativ. Es werden zuerst die Parameter $\boldsymbol{\Theta}$ initialisiert. Im E-Step können auf Grund der Parameter $\boldsymbol{\Theta}$ die Zugehörigkeitswahrscheinlichkeiten r_m^n berechnet werden. Im maximierenden Schritt (M-Step) werden die Parameter $\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m$ und α_m unter Zuhilfenahme von r_m^n neu berechnet. Der E und der M-Step werden abwechselnd durchgeführt bis die log-Likelihood-Funktion $L(\mathcal{X}|\boldsymbol{\Theta})$ konvergiert.
Die einzelnen Schritte sind im Folgenden kurz zusammengefasst.

1. Initialisierung: $t \ldots$ Iterationszähler

   $$\boldsymbol{\Theta}^{t=0} = \{\alpha_m^{t=0}, \boldsymbol{\mu}_m^{t=0}, \boldsymbol{\Sigma}_m^{t=0}\}_{m=1}^M$$

2. E-Step: Klassenzugehörigkeit ausrechnen

   $$r_m^n = \frac{\alpha_m^t \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m^t, \boldsymbol{\Sigma}_m^t)}{\sum\limits_{m'=1}^{n} \alpha_m^t \cdot \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{m'}^t, \boldsymbol{\Sigma}_{m'}^t)} = P(m \mid \mathbf{x}_n, \boldsymbol{\Theta}^t)$$

   $P(m \mid \mathbf{x}_n, \boldsymbol{\Theta}^t)$ gibt die Zugehörigkeitswahrscheinlichkeit für m gegeben $\mathbf{x}_n$ und $\boldsymbol{\Theta}$ an. Diese Posteriorverteilung ist faktisch gleich dem Posterior beim Bayes-Klassifikator mit der Annahme einer Normalverteilung als Likelihoodmodell.

3. M-Step: Berechnen der Parameter Θ

   $$\boldsymbol{\mu}_m^{t+1} = \frac{1}{N_m} \sum_{n=1}^{N} r_m^n \mathbf{x}_n$$
   $$\boldsymbol{\Sigma}_m^{t+1} = \frac{1}{N_m} \sum_{n=1}^{N} r_m^n (\mathbf{x}_n - \boldsymbol{\mu}_m^{t+1})(\mathbf{x}_n - \boldsymbol{\mu}_m^{t+1})^T$$
   $$\alpha_m^{t+1} = \frac{N_m}{N}$$
   $$t = t + 1$$

4. Evaluiere:

   $$L(\mathcal{X} \mid \boldsymbol{\Theta}^t) = \log(P(\mathcal{X} \mid \boldsymbol{\Theta}^t))$$
   $\rightarrow$ falls konvergiert Abbruch $\boldsymbol{\Theta}_{\mathrm{ML}} = \boldsymbol{\Theta}^t$
   $\rightarrow$ falls nicht konvergiert $\Rightarrow$ E-Step

4.2.1 Initialisierung

Eine Möglichkeit $\boldsymbol{\Theta}^0$ zu initialisieren ist:

1. α_m^0 auf uniforme Verteilungsfunktion $\alpha_m^0 = \frac{1}{M}$

2. $\mathbf{\Sigma}_m^0$ wird auf die Kovarianzmatrix $\mathbf{\Sigma}$ der Daten $\mathbf{X}$ gesetzt d.h. $\mathbf{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T$ wobei $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n$.

3. Für $\boldsymbol{\mu}_m^0$ wählt man m Samples zufällig aus oder man verwendet den k-means Algorithmus.

4.2.2 Eigenschaften des EM Algorithmus

1. Die log-Likelihood $\log(P(\mathcal{X} \mid \mathbf{\Theta}^t))$ wird in der Regel mit jeder Iteration monoton größer, siehe Abbildung 13.

2. Der EM Algorithmus findet lokale Optima d.h. ein lokales Maximum der Likelihood Funktion. Wenn es mehrere lokale Maxima gibt, wird das globale Optimum in der Regel nicht gefunden.

3. Daraus folgt, dass die Lösung von der Initialisierung von $\mathbf{\Theta}^0$ abhängt.

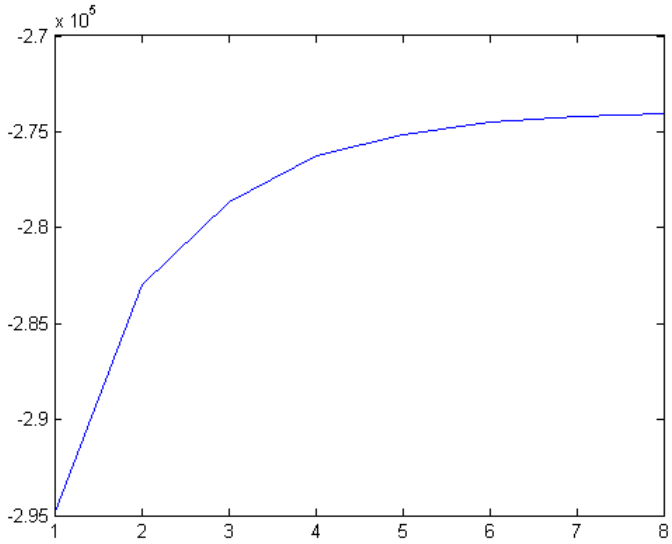

Abbildung 13: Verlauf der log-Likelihood $\log(P(\mathcal{X} \mid \mathbf{\Theta}^t))$ über die Iterationen.

5 Der K-means Algorithmus

Das Ziel des K-means Algorithmus ist es, die Daten in Cluster einzuteilen. Dabei werden die Anzahl der Cluster durch K bezeichnet. Eine Möglichkeit den K-means Algorithmus herzuleiten, ist das Modifizieren des EM-Algorithmus für GMMs in Kapitel 4.2. Unter folgenden Annahmen wird aus dem EM-Algorithmus für GMMs der K-means Algorithmus:

1. $\alpha_m = P(m) = 1/M \qquad \forall m$; Es wird α_m durch eine uniforme Wahrscheinlichkeitsverteilung modelliert und nicht modifiziert, d.h. α_m kann vernachlässigt werden; siehe Fall 1 in Kapitel 3.2.

2. $\mathbf{\Sigma}_m = \sigma^2\mathbf{I} \qquad \forall m$; Es werden alle Komponenten durch die gleiche sphärische Kovarianzmatrix dargestellt; siehe Fall 1 in Kapitel 3.2.

3. Klassifikation der Samples $\mathbf{x}_n$ zu Komponente m; d.h. $m = \arg\max_{m'}[r^n_{m'}]$; Jedes Sample wird von *einer* Komponente modelliert.

Nun modifizieren wir den E-step

$$r^n_m = \frac{\alpha_m \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum\limits_{m=1}^{n} \alpha_m \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}$$

unter oben genannten Bedingungen. Die posterior Wahrscheinlichkeit r^n_m ist ähnlich wie bei unserem Bayes-Klassifikator. Annahme 2 führt zur Entscheidungsfunktion $g_m(\mathbf{x}_n)$ (siehe Kapitel 3.2)

$$g_m(\mathbf{x}_n) = -\frac{(\mathbf{x}_n - \boldsymbol{\mu}_m)^T(\mathbf{x}_n - \boldsymbol{\mu}_m)}{2\sigma^2} + \ln P(m).$$

Durch Annahme 1 kann $\ln P(m)$ vernachlässigt werden. Weiters ist $2\sigma^2$ nur ein Skalierungsfaktor und kann ebenfalls vernachlässigt werden. So erhalten wir die Euklidische Distanz $g_m(\mathbf{x}_n) = -(\mathbf{x}_n - \boldsymbol{\mu}_m)^T(\mathbf{x}_n - \boldsymbol{\mu}_m)$ als Entscheidungsfunktion. Annahme 3 führt zur Klassifikation von $\mathbf{x}_n$ zur Komponente m

$$\begin{aligned} m^* &= \arg\max_m \left[g_m(\mathbf{x}_n)\right] \\ &= \arg\min_m \left[(\mathbf{x}_n - \boldsymbol{\mu}_m)^T(\mathbf{x}_n - \boldsymbol{\mu}_m)\right]. \end{aligned}$$

5.1 Funktionsweise von K-means

Der K-means Algorithmus zum Clustern der Daten $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ in K Cluster wird in diesem Abschnitt dargestellt. Dabei repräsentiert $\mathbf{Y}_m$ die Menge aller Datenpunkte $\mathbf{x}_n$ die zur Komponente m klassifiziert werden, d.h. $\mathbf{Y}_m = \{\mathbf{x}_n \mid m = \arg\min_m[(\mathbf{x}_n - \boldsymbol{\mu}_m)^T(\mathbf{x}_n - \boldsymbol{\mu}_m)]\}$. Im Folgenden werden die Schritte des K-means Algorithmus präsentiert. Dabei wird die Variable für Komponente m bei GMMs durch die Variable K ersetzt.

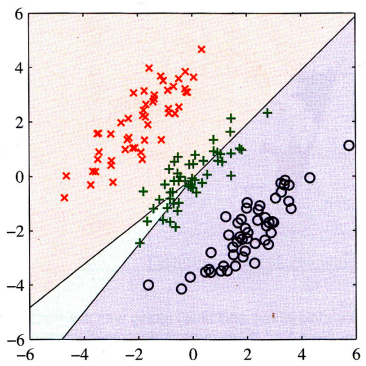

Abbildung 14: K-means bei Initialisierung [Bishop, 2007].

1. Initialisierung: $t \ldots$ Iterationszähler

   Wähle K Samples zufällig für die Clusterzentren $\boldsymbol{\mu}_k$ aus: $\boldsymbol{\Theta}^0 = \{\boldsymbol{\mu}_k^{t=0}\}_{k=1}^K$, $t = 0$. Die Initialisierung ist in Abbildung 14 dargestellt. Visuell sind in den Daten 3 Cluster ersichtlich. Die farblichen Regionen geben die Cluster bei der Initialisierung wieder.

2. Step 1: Klassifikation der Samples zu den Komponenten ($\rightarrow$ modifizierter E-step)

$$\mathbf{Y}_k = \{\mathbf{x}_n \mid k = \arg\min_{k'} \left[(\mathbf{x}_n - \boldsymbol{\mu}_{k'}^t)^T(\mathbf{x}_n - \boldsymbol{\mu}_{k'}^t)\right]\} \qquad \forall k = 1, \ldots, K$$

3. Step 2: Neuberechnung der Mittelwertvektoren (entspricht Schwerpunkt der Cluster) aufgrund der Zuweisung in $\mathbf{Y}_k$

$$\boldsymbol{\mu}_k^{t+1} = \frac{1}{\mid \mathbf{Y}_k \mid} \sum_{\mathbf{x}_n \in \mathbf{Y}_k} \mathbf{x}_n$$
$$t = t + 1$$

4. Evaluieren der kumulativen Distanz

$$J^t = \sum_{k=1}^{K} \sum_{\mathbf{x}_n \in \mathbf{Y}_k} (\mathbf{x}_n - \boldsymbol{\mu}_k^t)^T (\mathbf{x}_n - \boldsymbol{\mu}_k^t)$$

$\rightarrow$ falls J^t konvergiert, d.h. $|J^t - J^{t-1}| < \epsilon$, dann sind die optimalen Clusterzentren $\{\boldsymbol{\mu}_1^t, \ldots, \boldsymbol{\mu}_K^t\}$

$\rightarrow$ falls J^t nicht konvergiert, d.h. $|J^t - J^{t-1}| > \epsilon$, $\Rightarrow$ Step 1

In Abbildung 15 werden die Cluster am Ende der Iterationen dargestellt. Es werden alle 3 Cluster gut identifiziert.

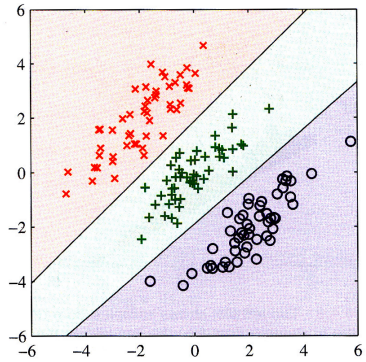

Abbildung 15: K-means nach Konvergenz [Bishop, 2007].

5.2 K-means Eigenschaften

1. K-means konvergiert zu lokalem Minimum der kumulativen Distanz J^t.

2. Mit jeder Iteration wird die kumulative Distanz J^t kleiner, d.h. $J^t \geq J^{t+1}$.

3. Ergebnis ist von der Initialisierung von $\boldsymbol{\Theta}^0$ abhängig, d.h. es wird in der Regel kein globales Optimum gefunden.

4. Entscheidungsgrenzen zwischen den Clustern sind stückweise linear.

Anwendungsbeispiele für K-means sind die Initialisierung von EM Algorithmus für eine Gaußsche Mischverteilung, die Diskretisierung von Daten oder Vektorquantisierung.

6 Markov Modell (MM)

Das Markov Modell ist zur Modellierung von Sequenzen geeignet, d.h. es modelliert explizit die Abhängigkeit zwischen den Samples. Die **iid** Annahme wird (zum Teil) vernachlässigt.

6.1 Beispiel: Grammatikmodell für Spracherkennung

Wir bezeichnen Q_n als Zustandsvariable wobei n der Zeitpunkt ist. In unserem Fall ist Q_n eine diskrete Zufallsvariable, wobei $Q_n \in \mathcal{W}$ einen Zustand aus einer endlichen Menge $\mathcal{W}$ annehmen kann.
Hier steht $\mathcal{W}$ für das Wörterbuch und jeder Zustand ist ein bestimmtes Wort.

Ein Beispiel für $\mathcal{W}$ ist $\mathcal{W} = \begin{pmatrix} \text{ich} \\ \text{gehe} \\ \vdots \end{pmatrix}$, wobei $|\mathcal{W}|$ die Anzahl der Wörter in $\mathcal{W}$ ist.

Ziel des Grammatikmodells ist es, Sätze (= Wortsequenzen) $Q_1, \ldots, Q_N$, wobei N die Satzlänge ist, mit einer Wahrscheinlichkeit $P(Q_1, \ldots, Q_N)$ zu bewerten, d.h. die Wahrscheinlichkeit für ein grammatikalisch richtigen Satz soll höher sein als für einen grammatikalisch falschen Satz.

Durch die Anwendung der Kettenregel erhalten wir folgende Faktorisierung:
$P(Q_1, ..., Q_N) = P(Q_1)P(Q_2|Q_1)P(Q_3|Q_2, Q_1) \cdot \ldots \cdot P(Q_N|Q_{N-1}, \ldots, Q_1)$.
Die Anzahl der Wahrscheinlichkeitsparameter für jeden dieser Terme ist:

$$\begin{aligned} \#P(Q_1) =& |\mathcal{W}| \\ \#P(Q_2|Q_1) =& |\mathcal{W}|^2 \\ \#P(Q_3|Q_2, Q_1) =& |\mathcal{W}|^3 \\ \#P(Q_N|Q_{N-1}, \ldots, Q_1) =& |\mathcal{W}|^N \end{aligned}$$

Konkret bedeutet das, dass die Anzahl der Wahrscheinlichkeiten exponentiell mit der Satzlänge N steigt. Bei einer Wörterbuchgröße von $|\mathcal{W}| = 1000$ und einer Satzlänge $N = 10$ hätten wir $(1000)^{10}$ Wahrscheinlichkeiten. Daraus folgt, dass das Schätzen und Speichern der Wahrscheinlichkeiten schnell unmöglich wird. Abhilfe schafft die Einschränkung des Kontext. Dies hat auch den Vorteil, dass Sequenzen unterschiedlicher Länge modelliert werden können.

- Unigramm Grammatik Modell: Hier wird die Annahme getroffen, dass keine Abhängigkeit zwischen den Wörtern besteht, d.h. die Wörter sind iid.
  $P(Q_1, \ldots, Q_n) = \prod_{i=1}^{N} P(Q_i)$
  Das Problem anhand des Beispielsatzes „Ich gehe bald einkaufen“ist, dass die Wahrscheinlichkeit für alle Permutationen der Wörter gleich ist, d.h. P(Ich gehe bald einkaufen) = P(einkaufen bald Ich gehe). Somit wird der grammatikalisch richtige Satz nicht mit einer höheren Wahrscheinlichkeit bewertet.

- Bigramm Modell: Hier wird die Annahme getroffen, dass $P(Q_1, \ldots, Q_N) = P(Q_1) \prod_{i=2}^{N} P(Q_i|Q_{i-1})$.
  Hierbei handelt es sich um ein Markov Modell 1. Ordnung. Die Parameter dieses Modells sind die prior Wahrscheinlichkeiten $P(Q_1)$ für das erste Wort am Satzanfang und die Übergangswahrscheinlichkeiten $P(Q_i|Q_{i-1})$. Diese Parameter können durch Maximum-Likelihood Schätzung aus einem Textkorpus gelernt werden. In der Regel durch einfaches Zählen der Wortkombinationen und deren Normalisierung. Dieses Modell beschränkt den Kontext auf das Vorgänger Wort (State) und die Auswahl ist überschaubar.

- Trigramm Modell: Hier wird die Annahme getroffen, dass $P(Q_1, \ldots, Q_N) = P(Q_1)P(Q_2|Q_1) \prod_{i=3}^{N} P(Q_i|Q_{i-1}, Q_{i-2})$ ist. Dies entspricht einem Markov Modell 2. Ordnung, der Kontext wird auf 2 vorangegangene States ausgeweitet, während bei einem Markov Modell 1. Ordnung nur ein vorangegangener State berücksichtigt wird (siehe Bigramm Modell).

6.2 Beispiel für ein MM 1. Ordnung zur Modellierung von Wettersequenzen

Ein Wettermodell hat 3 Zustände für das Wetter $Q_n \in \{S, R, F\}$, wobei die Zustände sonnig (S), regnerisch (R) und neblig (F) sind. Wenn das Wetter am ersten Tag $Q_1 = S$ gegeben ist, dann kann die Wahrscheinlichkeit für $P(Q_3 = R, Q_2 = S|Q_1 = S)$ mittels MM berechnet werden. Diese Wahrscheinlichkeit für ein MM 1. Ordnung faktorisiert wie folgt:

$$\begin{aligned} P(Q_3 = R, Q_2 = S|Q_1 = S) =& P(Q_3|Q_2, Q_1)P(Q_2|Q_1) \\ =& P(Q_3|Q_2)P(Q_2|Q_1). \end{aligned}$$

Nachdem $P(Q_3|Q_2, Q_1)$ statistisch unabhängig von Q_1 ist, folgt $P(Q_3|Q_2, Q_1) = P(Q_3|Q_2)$. Und $P(Q_3 = R, Q_2 = S|Q_1 = S)$ kann einfach als Produkt der Übergangswahrscheinlichkeiten ausgerechnet werden.

6.3 Parameter des Markov Modells

Ein Markov Modell besteht aus folgenden Parametern $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}\}$:

- Menge der Zustände $\mathcal{S} = \{s_1, \ldots, s_{N_S}\}$ wobei N_S die Anzahl der States ist.

- $\pi_i = \pi_{Q_1} = \pi_{Q_1=i} = \pi_{q1} = P(Q_1 = i)$ ist die Wahrscheinlichkeit für den State $i \in \mathcal{S}$ zum Zeitpunkt 1. Diese Wahrscheinlichkeit wird als Anfangswahrscheinlichkeit bezeichnet und ist in $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_{N_S}\}$ zusammengefasst.
- $a_{ij} = P(Q_n = j|Q_{n-1} = i) = a_{Q_{n-1},Q_n} = a_{q_{n-1},q_n} = a_{Q_{n-1}=i,Q_n=j}$ spezifiziert die Übergangswahrscheinlichkeit von State $i \in \mathcal{S}$ auf $j \in \mathcal{S}$. Die Übergangswahrscheinlichkeiten sind in $\mathbf{A} = [a_{ij}]_{N_S \times N_S}$ zusammengefasst. Eine vollbesetzte Matrix $\mathbf{A}$ bedeutet, dass man zu beliebigen Zeitpunkten von jedem Zustand in jeden anderen Zustand innerhalb eines Zeitschrittes wechseln kann. Dies wird auch als ergodisches Markov Modell bezeichnet. Bei einem Links-Rechts Modell kann man mit jedem Zeitschritt etweder im gleichen Zustand beleiben oder zu einem bisher noch nicht besuchten Zustand wechseln. In diesem Fall ist Matrix $\mathbf{A}$ eine obere Dreiecksmatrix.

7 Hidden Markov Modell

Im Unterschied zum Markov Modell (MM) ist beim Hidden Markov Model (HMM) der State nicht direkt beobachtbar (=hidden). Aber es gibt Beobachtungen $\mathbf{X}_n$, die stochastisch mit dem State Q_n zum selben Zeitpunkt n zusammenhängen und als Zufallsvariable modelliert werden. Die Beobachtungen $\mathbf{X}_n$ können diskret oder kontinuierlich sein bzw. ein Skalar oder ein Vektor. $P(\mathbf{X}_n|Q_n)$ wird als Beobachtungswahrscheinlichkeit oder Emissionswahrscheinlichkeit bezeichnet.
Bei den Beobachtungen wird die Annahme getroffen, dass $\mathbf{X}_n$ nur vom aktuellen State Q_n abhängt und nicht von anderen States oder Beobachtungen, d.h. $P(\mathbf{X}_n|Q_1, \ldots, Q_n, \ldots, Q_N, \mathbf{X}_1, \ldots, \mathbf{X}_{n-1}, \mathbf{X}_{n+1}, \ldots, \mathbf{X}_N) = P(\mathbf{X}_n|Q_n)$. Weitere Informationen sind bei [Rank and Pernkopf, 2004] zu finden.

7.1 Parameter von HMM

Die Parameter sind: $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$. Zusätzlich zum Markov Modell (siehe 6.3) gelten für das Hidden Markov Modell noch folgende Erweiterungen.

- Das HMM hat zusätzlich zu den Parametern des Markov Models noch die Beobachtungswahrscheinlichkeit. Die Beobachtungswahrscheinlichkeiten (Emissionswahrscheinlichkeiten) können diskret oder kontinuierlich, bzw. einzelne Werte oder eine Menge von Werten (Vektor) sein. Die Beobachtungswahrscheinlichkeiten werden mit Symbol $\mathbf{B}$ zusammengefasst.
  - diskrete Beobachtungen: $\mathbf{x}_n \in \{\vartheta_1, \ldots, \vartheta_k\}$: $b_{Q_n=i,\mathbf{X}_n=\mathbf{x}_n} = b_{i,\mathbf{x}_n} = P(\mathbf{X}_n = \mathbf{x}_n | Q_n = i)$
  - kontinuierliche Beobachtungen: $\mathbf{x}_n \in \mathbb{R}^d$: $b_{Q_n=i,\mathbf{x}_n} = b_{i,\mathbf{x}_n} = P(\mathbf{x}_n | Q_n = i)$; Die kontinuierlichen Beobachtungen können wieder mit einer Normalverteilung oder einer Gaußschen Mischverteilung modelliert werden.

Weiters gelten folgende statistische Unabhängigkeitsannahmen:

1. MM 1. Ordnung: Der aktuelle Zustand hängt nur vom vorherigen Zustand ab, d.h. $P(Q_n|Q_{n-1}, \ldots, Q_1) = P(Q_n|Q_{n-1})$.

2. Beobachtung $\mathbf{X}_n$ hängt nur von Q_n ab, d.h. $P(\mathbf{X}_n|Q_1, \ldots, Q_N, \mathbf{X}_1, \ldots, \mathbf{X}_{n-1}, \mathbf{X}_{n+1}, \ldots, \mathbf{X}_N) = P(\mathbf{X}_n|Q_n)$.

Nachdem die Parameter $\mathbf{\Theta}$ Wahrscheinlichkeitsverteilungen sind gelten folgende Normierungsbedingungen:

$$
\begin{aligned}
0 \leq \pi_i \leq 1 \quad &\text{und} \quad \sum_{i=1}^{N_s} \pi_i = 1 \\
0 \leq a_{ij} \leq 1 \quad &\text{und} \quad \sum_{j=1}^{N_s} a_{ij} = 1 \quad \forall i \\
0 \leq b_{i,\mathbf{x}_n} \leq 1 \quad &\text{und} \quad \mathbf{x}_n \text{ ist kontinuierlich: } \int_{-\infty}^{\infty} b_{i,\mathbf{x}_n} d\mathbf{x}_n = 1 \qquad \forall i \\
&\phantom{\text{und}} \quad \mathbf{x}_n \text{ ist diskret: } \sum_{\mathbf{x}_n \in \{\vartheta_1 \ldots \vartheta_k\}} b_{i,\mathbf{x}_n} = 1 \qquad \forall i
\end{aligned}
$$

In weiterer Folge benötigen wir folgende Terme: State Sequenz $\mathbf{Q} = \{Q_1, \ldots, Q_N\}$ und Beobachtungssequenz $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ bzw. deren Realisierungen $\mathbf{Q} = \{Q_1 = q_1, \ldots, Q_N = q_N\} = \{q_1, \ldots, q_N\}$ und $\mathbf{X} = \{\mathbf{X}_1 = \mathbf{x}_1, \ldots, \mathbf{X}_N = \mathbf{x}_N\} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

7.2 3 Problemstellungen

Es gibt 3 grundlegende Problemstellungen für den Einsatz von HMMs:

- Evaluierungsproblem (Klassifikationsproblem): Gegeben ist das Modell $\mathbf{\Theta}_t$ und eine Beobachtungssequenz $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Gesucht ist die Likelihood (Produktionswahrscheinlichkeit) $P(\mathbf{X}|\mathbf{\Theta}_t)$ einer Beobachtungssequenz. Diese Likelihood kann zur Klassifikation im Bayes Klassifikator herangezogen werden, d.h. der Bayes Klassifikator kann Sequenzen $\mathbf{X}$ unterschiedlicher Länge klassifizieren:

$$
P(t|\mathbf{X}) = \frac{P(\mathbf{X}|t)P(t)}{\sum_{t'=1}^{T} P(\mathbf{X}|t')P(t')}
$$

Für die Likelihood Funktion $P(\mathbf{X}|t)$ kann das HMM verwendet werden: $P(\mathbf{X}|t) = P(\mathbf{X}|\mathbf{\Theta}_t)$ wobei $\mathbf{\Theta}_t$ das Modell für Klasse t repräsentiert. Man wählt jene Klasse t^* mit der größten posterior Wahrscheinlichkeit $P(t|\mathbf{X})$:

$$t^* = \arg\max_t \left[P(\mathbf{X}|t)P(t)\right] = \arg\max_t \left[(\mathbf{X}|\mathbf{\Theta}_t)P(t)\right]$$

Die Berechnung von $P(\mathbf{X}|\mathbf{\Theta}_t)$ erfolgt mit dem Forward/Backward Algorithmus (siehe 7.2.1).

- Dekodierungsproblem: Gegeben ist das Modell $\mathbf{\Theta}$ und die Beobachtungssequenz $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Gesucht ist $\mathbf{Q}^* = \arg\max_{\mathbf{Q}} \left[P(\mathbf{Q}|\mathbf{X}, \mathbf{\Theta})\right]$. $\mathbf{Q}^*$ bezeichnet eine Zustandsfolge (= Statesequenz) die eine Beobachtungssequenz $\mathbf{X}$ bei gegebenen Parametern $\mathbf{\Theta}$ am besten erklärt. Der Viterbi Algorithmus (siehe 7.2.3) dient zur Ermittlung von $\mathbf{Q}^*$.

- Schätzproblem (Lernproblem): Gegeben sind R Beobachtungssequenzen: $\mathbf{X}^{1:R} = \{\mathbf{X}^1, \ldots, \mathbf{X}^R\}$. Gesucht sind die Parameter des Modells, d.h.

$$\mathbf{\Theta}_{\mathrm{ML}} = \arg\max_{\mathbf{\Theta}} \left[P(\mathbf{X}^{1:R}|\mathbf{\Theta})\right] = \arg\max \left[\prod_{r=1}^{R} P(\mathbf{X}^r|\mathbf{\Theta})\right], \tag{16}$$

  wobei $P(\mathbf{X}^r|\mathbf{\Theta})$ die Likelihood des Evaluierungsproblems darstellt. Die ML Lösung $\mathbf{\Theta}_{\mathrm{ML}}$ kann mit dem EM Algorithmus (ähnlich zu 4.2) gefunden werden.

Die oben angeführten Algorithmen für die Lösung des jeweiligen Problems werden in den folgenden Abschnitten genauer erläutert.

7.2.1 Evaluierungsproblem / Klassifizierungsproblem

Gegeben sind die Beobachtungssequenz $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ und die Modellparameter $\mathbf{\Theta}$. Gesucht ist die Produktionswahrscheinlichkeit $P(\mathbf{X}|\mathbf{\Theta})$. Die Wahrscheinlichkeit für die Statesequenzen $\mathbf{Q} = \{q_1, \ldots, q_N\}$ ($\hat{=}$ Markov Modell 1. Ordnung) bei gegebenen Modell ist:

$$\begin{aligned} P(\mathbf{Q}|\mathbf{\Theta}) &= P(q_1, \ldots, q_N|\mathbf{\Theta}) = \\ &= P(q_1) \prod_{n=2}^{N} P(q_n|q_{n-1}) = \pi_{q_1} \prod_{n=2}^{N} a_{q_{n-1},q_n} \end{aligned}$$

Die Wahrscheinlichkeit für eine Beobachtungssequenz $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ bei gegebener Statesequenz $\mathbf{Q} = \{q_1, \ldots, q_N\}$ und verwendeten $\mathbf{\Theta}$ ist:

$$P(\mathbf{X}|\mathbf{Q}, \mathbf{\Theta}) = P(\mathbf{x}_1, \ldots, \mathbf{x}_N|q_1, \ldots, q_N, \mathbf{\Theta}) = \prod_{n=1}^{N} P(\mathbf{x}_n|q_n) = \prod_{n=1}^{N} b_{q_n,\mathbf{x}_n}$$

Die Verbund-Wahrscheinlichkeit ist

$$\begin{aligned} P(\mathbf{X}, \mathbf{Q}|\mathbf{\Theta}) &= P(\mathbf{X}|\mathbf{Q}, \mathbf{\Theta})P(\mathbf{Q}|\mathbf{\Theta}) \\ &= \prod_{n=1}^{N} P(\mathbf{x}_n|q_n)P(q_1) \prod_{n=2}^{N} P(q_n|q_{n-1}) \\ &= P(q_1)P(\mathbf{x}_1|q_1) \prod_{n=2}^{N} P(q_n|q_{n-1})P(\mathbf{x}_n|q_n) \\ &= \pi_{q_1} b_{q_1,\mathbf{x}_1} \prod_{n=2}^{N} a_{q_{n-1},q_n} b_{q_n,\mathbf{x}_n} \end{aligned}$$

Die Produktionswahrscheinlichkeit errechnet sich durch marginalisieren von $P(\mathbf{X}, \mathbf{Q}|\mathbf{\Theta})$, d.h. $P(\mathbf{X}|\mathbf{\Theta}) = \sum_{\mathbf{Q}\in\mathcal{Q}} P(\mathbf{X}, \mathbf{Q}|\mathbf{\Theta})$, wobei $\mathcal{Q}$ die Menge aller Statesequenzen durch das HMM / Trellis ist. Das Trellis Diagramm dient zur zeitlichen Darstellung des HMMs (Abbildung 16).

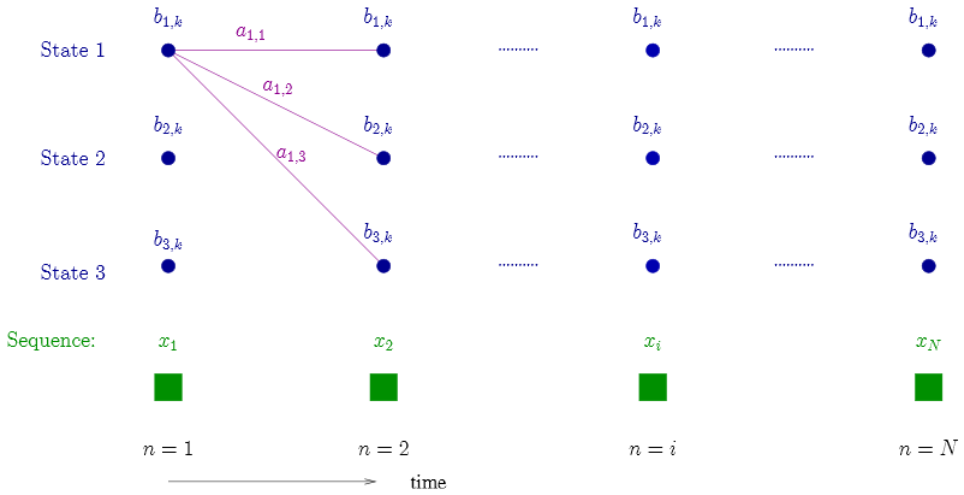

Abbildung 16: Trellis Diagramm [Rank and Pernkopf, 2004].

Die Anzahl der Statesequenzen ist $|\mathcal{Q}| = (N_s)^N$. D.h., dass naive Marginalisierung $O(2N(N_s)^N)$ Rechenoperationen benötigt.
Der Forward / Backward Algorithmus nutzt die Faktorisierungseigenschaft von $P(\mathbf{X}, \mathbf{Q}|\Theta)$ und kann mit Hilfe einer Rekursion die Anzahl der Rechenoperationen auf $O(2(N_s)^2N)$ reduzieren. Der Grund für die Reduktion liegt in der Nutzung von Zwischenergebnissen auf partiellen Pfaden (Distributiv Gesetz: $a(b + c) = ab + ac$). Im Folgenden ist die Berechnung der *Forward-Wahrscheinlichkeit* mittels Forward Algorithmus dargestellt. Die Marginalisierung der Forward-Wahrscheinlichkeit zum Zeitpunkt N über die States liefert die gewünschte Produktionswahrscheinlichkeit.

Berechnen der Forward-Wahrscheinlichkeiten: Die Forward-Wahrscheinlichkeit ist wie folgt definiert: $\alpha_n(j) = P(\mathbf{x}_1, \ldots, \mathbf{x}_n, Q_n = j|\mathbf{\Theta})$

- Initalisierung

$$\alpha_1(j) = \pi_j b_{j,\mathbf{x}_1} \quad \forall j = 1, \ldots, N_s$$

- Rekursion

$$\alpha_n(j) = \left(\sum_{i=1}^{N_s} \alpha_{n-1}(i) a_{i,j}\right) b_{j,\mathbf{x}_n} \quad \forall n = 2, \ldots, N; \quad \forall j = 1 \ldots N_s$$

- Berechnung der Produktionswahrscheinlichkeit:

$$P(\mathbf{X}|\mathbf{\Theta}) = P(\mathbf{x}_1, \ldots, \mathbf{x}_N|\mathbf{\Theta}) = \sum_{j=1}^{N_s} \alpha_N(j)$$

Ähnlich kann die Produktionswahrscheinlichkeit auch mittels Backward-Wahrscheinlichkeiten berechnet werden. Diese sind wie folgt definiert: $\beta_n(j) = P(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N|Q_n = j, \mathbf{\Theta})$.

- Initalisierung

$$\beta_N(i) = 1 \quad \forall i = 1, \ldots, N_s$$

- Rekursion

$$\beta_n(i) = \sum_{j=1}^{N_s} a_{i,j} b_{j,\mathbf{x}_{n+1}} \beta_{n+1}(j) \quad \forall n = N-1, \ldots, 1; \quad \forall i = 1, \ldots, N_s$$

- Berechnung der Produktionswahrscheinlichkeit:

$$P(\mathbf{X}|\mathbf{\Theta}) = P(\mathbf{x}_1, \ldots, \mathbf{x}_N|\mathbf{\Theta}) = \sum_{j=1}^{N_s} \pi_j b_{j,\mathbf{x}_1} \beta_1(j)$$

7.2.2 Decodierungsproblem

Das Decodierungsproblem wird durch den Viterbi-Algorithmus gelöst. Gegeben sind $\mathbf{\Theta}$ und $\mathbf{X}$. Gesucht ist die Statesequenz $\mathbf{Q}^* = \{q_1^*, \ldots, q_N^*\}$ die $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ am besten erklärt, d.h.

$$Q^* = \arg\max_{\mathbf{Q}\in\mathcal{Q}} P(\mathbf{Q}|\mathbf{X}, \mathbf{\Theta}) = \arg\max_{\mathbf{Q}\in\mathcal{Q}} \frac{P(\mathbf{X}, \mathbf{Q}|\mathbf{\Theta})}{P(\mathbf{X}|\mathbf{\Theta})} = \arg\max_{\mathbf{Q}\in\mathcal{Q}} P(\mathbf{X}, \mathbf{Q}|\mathbf{\Theta}).$$

Der Term $P(\mathbf{X}|\mathbf{\Theta})$ ist unabhängig von $\mathbf{Q}$ und entspricht nur einer Skalierung von $P(\mathbf{Q}|\mathbf{X}, \mathbf{\Theta})$, d.h. Das Argument des Maximierungsoperators wird dadurch nicht verändert. Wir definieren die Wahrscheinlichkeit entlang des Pfades $\mathbf{Q}^*$ als $P^*(\mathbf{X}|\mathbf{\Theta})$.

$$P^*(\mathbf{X}|\mathbf{\Theta}) \stackrel{!}{=} P(\mathbf{Q}^*, \mathbf{X}|\mathbf{\Theta}) = \underbrace{\max_{\mathbf{Q}\in\mathcal{Q}} P(\mathbf{X}, \mathbf{Q}|\mathbf{\Theta})}_{\text{wird bei Viterbi rekursiv berechnet}}$$

Als $\delta_n(i)$ bezeichnet man die maximal erzielbare Wahrscheinlichkeit für eine partielle Beobachtungssequenz $\mathbf{x}_1, \ldots, \mathbf{x}_n$ entlang eines einzigen Pfades $Q_1, \ldots, Q_{n-1}$ der zum Zeitpunkt n im State $Q_n = i$ mündet, d.h. $\delta_n(i) = \max_{Q_1,\ldots,Q_{n-1}} P(Q_1, \ldots, Q_{n-1}, Q_n = i, \mathbf{x}_1, \ldots, \mathbf{x}_n|\Theta)$.
Die Berechnung von $\delta_n(i)$ erfolgt rekursiv, d.h. $\delta_n(j) = \max\left[\delta_{n-1}(i)a_{ij}\right] b_{j,\mathbf{x}_n}$, wobei $\delta_1(i) = \pi_1 b_{i,\mathbf{x}_1}$ ist. Bei $n = N$ entspricht:

$$\begin{aligned}
\delta_N(i) &= \max_{Q_1,\ldots,Q_{N-1}} P(Q_1, \ldots, Q_N = i, \mathbf{x}_1, \ldots, \mathbf{x}_N|\Theta) \\
P^*(\mathbf{X}|\mathbf{\Theta}) &= \max_{1\le j\le N_s} \delta_N(j)
\end{aligned}$$

Weiters wird für die Extraktion von $\mathbf{Q}^*$ eine Variable benötigt, die das Argument für die Maximum-Entscheidung speichert. Das ist erforderlich um zum Zeitpunkt $n = N$ den besten Pfad $\mathbf{Q}^*$ extrahieren zu können. Diese Variable ist $\psi_n(i)$.

$$\psi_n(i) \stackrel{!}{=} \arg\max_{Q_1,\ldots,Q_{n-1}} \left[P(Q_1, \ldots, Q_{n-1}, Q_n = i, \mathbf{x}_1, \ldots, \mathbf{x}_n|\mathbf{\Theta})\right]$$

Die Extraktion des Pfades $\mathbf{Q}^*$ erfolgt durch *backtracking*

$$q_n^* = \psi_{n+1}(q_{n+1}^*) \qquad n = N-1, \ldots, 1.$$

7.2.3 Viterbi Algorithmus

In diesem Abschnitt ist der Viterbi Algorithmus zusammengefasst.

- Initialisierung

$$\delta_1(i) = \pi_i b_{i,\mathbf{x}_1} \quad \forall i = 1, \ldots, N_s \psi_1(i) \qquad\qquad = 0 \quad \forall i = 1, \ldots, N_s$$

- Rekursion

$$\begin{aligned}
\delta_n(j) &= \max_{1\le i\le N_s} \left[\delta_{n-1}(i)a_{ij}\right] b_{j,\mathbf{x}_n} \quad \forall j = 1, \ldots, N_s \quad \forall n = 2, \ldots, N \\
\psi_n(j) &= \arg\max_{1\le i\le N_s} \left[\delta_{n-1}(i)a_{ij}\right] \quad \forall j = 1, \ldots, N_s \quad \forall n = 2, \ldots, N
\end{aligned}$$

- Termination

$$P^*(\mathbf{X}|\mathbf{\Theta}) = \max_{1 \leq j \leq N_s} \delta_N(j)$$
$$q_N^* = \arg \max_{1 \leq j \leq N_s} \delta_N(j)$$

- Backtracking Der beste Pfad $\mathbf{Q}^*$ mündet zum Zeitpunkt N in q_N^*. Durch Backtracking kann $q_{N-1}^*, \ldots, q_1^*$ extrahiert werden, d.h.

$$q_n^* = \psi_{n+1}(q_{n+1}^*), \qquad n = N-1, N-2, \ldots, 1$$

Ein Beispiel für den Viterbi-Algorithmus ist in den Folien zu finden [Rank and Pernkopf, 2004].

8 Graphische Modelle

Die Grafischen Modelle werden in den Folien der Vorlesung [Pernkopf, 2013] genauer beschrieben. Im Tutorial [Pernkopf et al., 2013] in den Kapiteln 3, 4.1, 4.2, 5.1 (Bayesian networks), 5.2 (Markov networks), 5.4 (Markov chain), 7 (Inference), 7.1 (Exact inference), und 8.1 (HMMs, evaluation problem) können weiterführende Informationen gefunden werden.

9 Lineare Transformationen

9.1 Dimensionsreduktion

Durch Dimensionsreduktion wird es möglich, Daten eines höher dimensionalen Raumes in einen Raum mit weniger Dimensionen abzubilden. Das ist in der Regel mit Verlust von Information verbunden (verlustbehaftete Datenkompression). Zusätzlich können die statistischen Eigenschaften der Daten nach der Transformation auch günstiger sein, z.B. de-korreliert.
Es sei gegeben $\{\mathbf{x}_n\}_{n=1}^N, \mathbf{x}_n \in \mathbb{R}^D$, wobei D die Dimension des ursprünglichen Raumes und M die Dimension nach der Transformation ist.

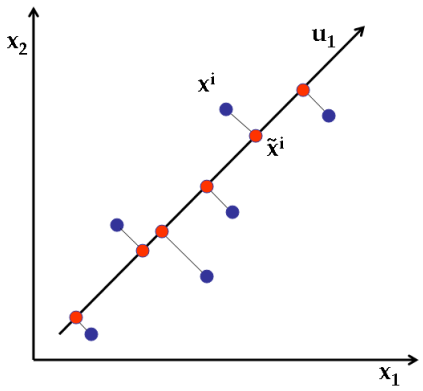

Abbildung 17: Dimensionsreduktion [Learning, 2013]

9.2 Projektion von x

Angenommen wir möchten die Daten auf in einen eindimensionalen Raum ($M = 1$) transformieren (siehe Abbildung 17. Nun ist es uns möglich, die Richtung dieses Raumes, in Bezug auf den originalen, D-dimensionalen Raum, mithilfe des D-dimensionalen Vektors $\mathbf{u}_1$ zu bestimmen. Uns interessiert hierbei

lediglich die Richtung des Vektors $\mathbf{u}_1$, mit dessen Hilfe es uns möglich ist jeden Datenpunkt y_n zu berechnen, d.h. die Projektion ist

$$y_n = \mathbf{u}_1^T \mathbf{x}_n. \tag{17}$$

Im $M \leq D$ dimensionalen Raum können wir (17) entsprechend anpassen und erhalten:

$$\mathbf{y}_n = \mathbf{U}^T \mathbf{x}_n \tag{18}$$

$$= \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_M^T \end{pmatrix} \mathbf{x}_n, \tag{19}$$

wobei $\mathbf{y}_n \in \mathbb{R}^M$.

9.2.1 Statistische Eigenschaften der transformierten Daten ($M = 1$)

Um die Transformation zu optimieren benötigen wir die Varianz, sowie den Mittelwert der transformierten Daten. Der Mittelwert berechnet sich durch:

$$\begin{aligned} m_y &= \frac{1}{N} \sum_{n=1}^{N} \mathbf{u}_1^T \mathbf{x}_n \\ &= \mathbf{u}_1^T \mathbf{m}_x, \end{aligned}$$

wobei $\mathbf{m}_x$ der Mittelwert der Daten $\mathbf{x}_n$ ist. Die Varianz berechnet sich wie folgt:

$$\begin{aligned} \sigma_y^2 &= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \mathbf{m}_x)^2 \\ &= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{u}_1^T (\mathbf{x}_n - \mathbf{m}_x))^2 \\ &= \mathbf{u}_1^T \underbrace{\frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{m}_x)(\mathbf{x}_n - \mathbf{m}_x)^T}_{\mathbf{S}} \mathbf{u}_1 \end{aligned}$$

$$= \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1, \tag{20}$$

wobei $\mathbf{S}$ beschreibt die Kovarianzmatrix der Daten $\mathbf{x}_n$ ist.

9.2.2 Principal Component Analysis (PCA) für M=1

Aus Abbildung 18 lässt sich vermuten, dass in Richtung des größten Energieanteils (Varianz) auch die meiste Information enthalten ist. Genau dies ist auch die Annahme auf welcher PCA (=Hauptkomponentenanalyse) beruht. Wir nehmen an, dass die Varianz der Information entspricht. Wir wollen also auf Richtung der maximalen Varianz projizieren um möglichst wenig Information zu verlieren.

$$\begin{aligned} \mathbf{u}_1 &= \arg\max_{\mathbf{u}_1} [\sigma_y^2] \\ &= \arg\max_{\mathbf{u}_1} [\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1] \end{aligned}$$

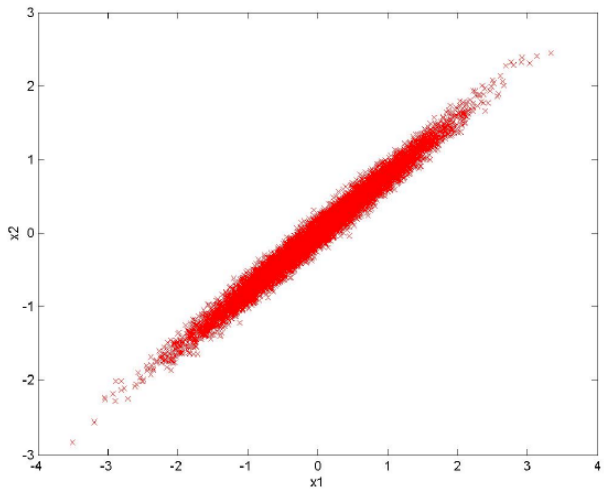

Abbildung 18: Ausgangsdaten für eine PCA [Wohlmayr, 2013]

Um zu vermeiden, dass $\| \mathbf{u}_1 \| \rightarrow \infty$, führen wir eine Normalisierungsbedingung ein:

$$\| \mathbf{u}_1 \| = \mathbf{u}_1^T \mathbf{u}_1 = 1$$

Die entsprechende Nebenbedingung $g(\mathbf{x})$ für Lagrange Optimierung (siehe Anhang A) ergibt sich aus der Normalisierungsbedingung, d.h.

$$g(\mathbf{x}) = 1 - \mathbf{u}_1^T \mathbf{u}_1 = 0$$

Wir lösen mithilfe der Lagrange Optimierung und erhalten:

$$\mathcal{L}(\mathbf{u}_1, \lambda_1) = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$

$$\frac{\partial \mathcal{L}(\mathbf{u}_1, \lambda_1)}{\partial \mathbf{u}_1} = 2\mathbf{S}\mathbf{u}_1 - \lambda_1 2\mathbf{u}_1 \stackrel{!}{=} 0 \tag{21}$$

Aus (27) folgt

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \tag{22}$$

Dies sagt aus, dass $\mathbf{u}_1$ ein Eigenvektor von $\mathbf{S}$ zum Eigenwert λ_1 sein muss. Wenn wir nun mit $\mathbf{u}_1^T$ multiplizieren und uns $\mathbf{u}_1^T \mathbf{u}_1 = 1$ zu nutze machen erhalten wir

$$\lambda_1 = \mathbf{u}_1^T S \mathbf{u}_1 = \sigma_y^2 \tag{23}$$

was der transformierten Varianz entspricht. Daraus erkennen wir, dass die Varianz ihr Maximum erreicht, wenn wir $\mathbf{u}_1$ gleich dem Eigenvektor mit dem größten Eigenwert λ_1 setzen. $\mathbf{u}_1$ wird oft auch als erste Hauptkomponente (principal component) bezeichnet.

9.3 PCA für $M > 1$

Die PCA wird hauptsächlich für die Vorverarbeitung von Daten verwendet. Wie in (18) dargestellt, können wir auf M Dimensionen projizieren, d.h.

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_M \end{pmatrix}}_{\mathbf{y}_n} = \underbrace{\begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_M^T \end{pmatrix}}_{\mathbf{U}^T} \cdot \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_D \end{pmatrix}}_{\mathbf{x}_n}. \tag{24}$$

Dazu sind mehrere Transformationsvektoren $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M]$ notwendig. Diese Vektoren sind die Eigenvektoren der Kovarianzmatrix $\mathbf{S}$. Diese erhält man durch lösen des Eigenwertproblems $\mathbf{SU} = \mathbf{UL}$, wobei die Diagonalmatrix $\mathbf{L}$ die Eigenwerte der Kovarianzmatrix enthält, d.h.

$$\mathbf{L} = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_0 \end{pmatrix}. \tag{25}$$

Wir erinnern uns dass die Eigenwerte der Varianzen entsprechen (siehe (23)). Die Kovarianzmatrix der transformierten Daten $\mathbf{y}_n$ ist eine Diagonalmatrix, d.h. die transformierten Daten sind dekorreliert (y_i und y_j, $i \neq j$ sind paarweise dekorreliert.)

Wir modifizieren die PCA und führen die Projektion wie folgt durch:

$$\mathbf{y}_n = \mathbf{L}^{-\frac{1}{2}} \mathbf{U}^T (\mathbf{x}_n - \mathbf{m}_x)$$

Diese Transformation wird in der Literatur oft als *whitening* bezeichnet. Wir sehen, dass die transformierten Daten mittelwertfrei sind und eine Einheitsmatrix als Kovarianzmatrix aufweisen, d.h.:

- $\mathbf{m}_y = \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n = 0$
- $\mathbf{S}_y = \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n \mathbf{y}_n^T$
  $= \mathbf{L}^{-\frac{1}{2}} \mathbf{U}^T \underbrace{\frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{m}_x)(\mathbf{x}_n - \mathbf{m}_x)^T}_{\mathbf{S}} \mathbf{U} \mathbf{L}^{-\frac{1}{2}}$
  $= \mathbf{L}^{-\frac{1}{2}} \underbrace{\mathbf{U}^T \mathbf{S} \mathbf{U}}_{\mathbf{L}} \mathbf{L}^{-\frac{1}{2}} = \mathbf{I}$ (Einheitsmatrix)

9.4 LDA - Linear Discriminant Analysis (overview)

Das projizieren von Daten in eine Dimension M ($M < D$) führt unweigerlich zu Datenverlust. Es könnte z.B. sein, dass gut separierte Klassen im Originalraum nach der Projektion mittels PCA in einem D-dimensionalen Raum stark überlappend sind (Beispiel Tafel). PCA berücksichtigt keine Klassenlabels bei der Bestimmung der Transformationsmatrix.

Wie wir bereits wissen projizieren wir in der Form:

$$y = \mathbf{u}^T \mathbf{x} \tag{26}$$

Um viel Information von den einzelnen Klassen nach der Transformation zu erhalten, können wir $\mathbf{u}$ so wählen, dass die Distanz zwischen den beiden Klassenzentren, d.h. den Mittelwerten, nach der Transformation maximiert wird.

Als Beispiel nehmen wir an, dass wir 2 Klassen trennen müssten. Folgende gelabelte (Klassenzugehörigkeit bekannt) Daten sind uns hierfür bekannt:

$$\mathbf{X} = \{\underbrace{\mathbf{x}_1, \ldots, \mathbf{x}_{N_1}}_{T_1}; \underbrace{\mathbf{x}_{N_1+1}, \ldots, \mathbf{x}_N}_{T_2}\}$$

Die Mittelwertvektoren der beiden Klassen sind gegeben durch:

$$\mathbf{m}_{x,1} = \frac{1}{|T_1|} \sum_{\mathbf{x}_n \in T_1} \mathbf{x}_n \qquad \text{und} \qquad \mathbf{m}_{x,2} = \frac{1}{|T_2|} \sum_{\mathbf{x}_n \in T_2} \mathbf{x}_n$$

Nach der Transformation sind die Mittelwerte der projezierten Daten

$$m_{y,1} = \mathbf{u}^T \mathbf{m}_{x,1} \qquad \text{und} \qquad m_{y,2} = \mathbf{u}^T \mathbf{m}_{x,2}$$

Annahme: Die einfachste Lösung wäre es, $\mathbf{u}$ so zu wählen, dass die Distanz zwischen den Mittelwerten der transformierten Daten am größten ist. Dafür verwenden wir folgenden Ansatz:

$$\begin{aligned} \mathbf{u} &= \arg\max_{\mathbf{u}} (|m_{y,2} - m_{y,1}|) \\ &= \arg\max_{\mathbf{u}} \mathbf{u}^T \left(|\mathbf{m}_{x,2} - \mathbf{m}_{x,1}|\right) \\ &\text{subject to: } \mathbf{u}^T \mathbf{u} = 1 \end{aligned}$$

Wir verwenden wieder den Ansatz von Lagrange, um die Maximierung durchzuführen, d.h.

$$\mathcal{L}(\mathbf{u}, \lambda) = \mathbf{u}^T \left(|\mathbf{m}_{x,2} - \mathbf{m}_{x,1}|\right) + \lambda(1 - \mathbf{u}^T \mathbf{u}).$$

Die Ableitung wird auf Null gesetzt, d.h.

$$\frac{\partial \mathcal{L}(\mathbf{u}, \lambda)}{\partial \mathbf{u}} = |\mathbf{m}_{x,2} - \mathbf{m}_{x,1}| - \lambda 2\mathbf{u} \stackrel{!}{=} 0 \tag{27}$$

und wir erhalten:

$$\mathbf{u} \propto (\mathbf{m}_{x,2} - \mathbf{m}_{x,1})$$

Das erzielte Ergebnis ist nicht gerade optimal, da es sich nur auf den Mittelwert bezieht. Siehe Abbildung 19 links, es ist deutlich zu erkennen, dass sich die 2 Klassen nach der Projektion überlappen.

Bei der LDA wird dem Problem Abhilfe geleistet, indem wir auch die Varianz der Daten nach der Transformation miteinbeziehen:

$$\sigma_{y,k}^2 = \sum_{n \in T_k} (y_n - m_{y,k})^2 = \mathbf{u}^T \mathbf{S}_k \mathbf{u}.$$

Wir können die Within-Class Kovarianz für die Daten $\mathbf{X}$ leicht bestimmen. Die Within-Class-Kovarianz gibt die Kovarianz innerhalb einer Klasse an, d.h.

$$\sigma_y^2 = \sigma_{y,1}^2 + \sigma_{y,2}^2 = \mathbf{u}^T \mathbf{S}_1 \mathbf{u} + \mathbf{u}^T \mathbf{S}_2 \mathbf{u} = \mathbf{u}^T \mathbf{S}_w \mathbf{u},$$

wobei die Within-Class Kovarianz $\mathbf{S}_w$ gleich

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2 = \sum_{\mathbf{x}_n \in T_1} (\mathbf{x}_n - \mathbf{m}_{x,1})(\mathbf{x}_n - \mathbf{m}_{x,1})^T + \sum_{x_n \in T_2} (\mathbf{x}_n - \mathbf{m}_{x,2})(\mathbf{x}_n - \mathbf{m}_{x,2})^T$$

ist. Als nächstes benötigen wir noch die Between-Class Kovarianzmatrix welche wir mit $\mathbf{S}_b$ bezeichnen werden. Die Between-Class Kovarianz spiegelt die Separation von den projezierten Mittelwerten wieder, d.h.

$$(m_{y,2} - m_{y,1})^2 = \left[\mathbf{u}^T(\mathbf{m}_{x,2} - \mathbf{m}_{x,1})\right]^2 = \mathbf{u}^T(\mathbf{m}_{x,2} - \mathbf{m}_{x,1})(\mathbf{m}_{x,2} - \mathbf{m}_{x,1})^T\mathbf{u} = \mathbf{u}^T\mathbf{S}_b\mathbf{u},$$

wobei

$$\mathbf{S}_b = (\mathbf{m}_{x,2} - \mathbf{m}_{x,1})(\mathbf{m}_{x,2} - \mathbf{m}_{x,1})^T.$$

Wir wollen uns nun das Verhältnis zwischen Between-Class und Within-Class Kovarianz ansehen und dieses maximieren, d.h. die Distanz der Mittelwerte der projezierten Daten soll möglichst groß sein (Between-Class Kovarianz maximieren) und die Kovarianz der projezierten Daten soll möglichst kompakt sein (kleine Within-Class Kovarianz). Das führt zu folgendem Kiterium (auch bekannt als Fisher Kriterium)

$$J(\mathbf{u}) = \frac{(m_{y,2} - m_{y,1})^2}{\sigma_{y,1}^2 + \sigma_{y,2}^2} = \frac{\mathbf{u}^T\mathbf{S}_b\mathbf{u}}{\mathbf{u}^T\mathbf{S}_w\mathbf{u}}. \tag{28}$$

Wir leiten $J(\mathbf{u})$ (28) nach $\mathbf{u}$ ab und setzen die Ableitung auf 0. Dies ergibt

$$(\mathbf{u}^T\mathbf{S}_b\mathbf{u})\mathbf{S}_w\mathbf{u} = (\mathbf{u}^T\mathbf{S}_w\mathbf{u})\mathbf{S}_b\mathbf{u}.$$

Die Terme $(\mathbf{u}^T\mathbf{S}_b\mathbf{u})$ und $(\mathbf{u}^T\mathbf{S}_w\mathbf{u})$ sind skalare Größen und wir erhalten ein allgemeines Eigenwertproblem

$$\lambda\mathbf{S}_w\mathbf{u} = \mathbf{S}_b\mathbf{u}. \tag{29}$$

Wir multiplizieren beide Seiten mit $\mathbf{S}_w^{-1}$ und sehen, dass $\mathbf{S}_b\mathbf{u}$ immer in Richtung $(\mathbf{m}_{x,2} - \mathbf{m}_{x,1})$ ist. Wir erhalten schließlich unsere Lösung:

$$\mathbf{u} \propto \mathbf{S}_w^{-1}(\mathbf{m}_{x,2} - \mathbf{m}_{x,1})$$

In Abbildung 19 sehen wir die Projektion auf $\mathbf{u} \propto (\mathbf{m}_{x,2} - \mathbf{m}_{x,1})$ (links). Dabei ist eine starke Überlappung der Klassen nach der Projektion festzustellen. Rechts sehen wir das Ergebnis von LDA, wenn wir die Varianz in die Optimierung miteinbeziehen. Dabei kann eine deutliche Verbesserung erzielt werden. LDA kann auf mehr als zwei Klassen erweitert werden.

10 Anhang A

10.1 Optimierung mittels Lagrange-Multiplikatoren

Die Lagrange Optimierung wenden wir an, wenn wir keine Kurvendiskussion zur Ermittlung von Extremwerten durchführen wollen oder können, z.B. wenn wir eine Nebenbedingung erfüllen müssen, die nicht einfach in die Kurvendiskussion einsetzbar ist. Angenommen wir wollen einen Extremwert x_0 einer Funktion $f(x)$ (Objective function) mit einer Nebenbedingung $g(x)$ (Constraint) finden. Wenn die

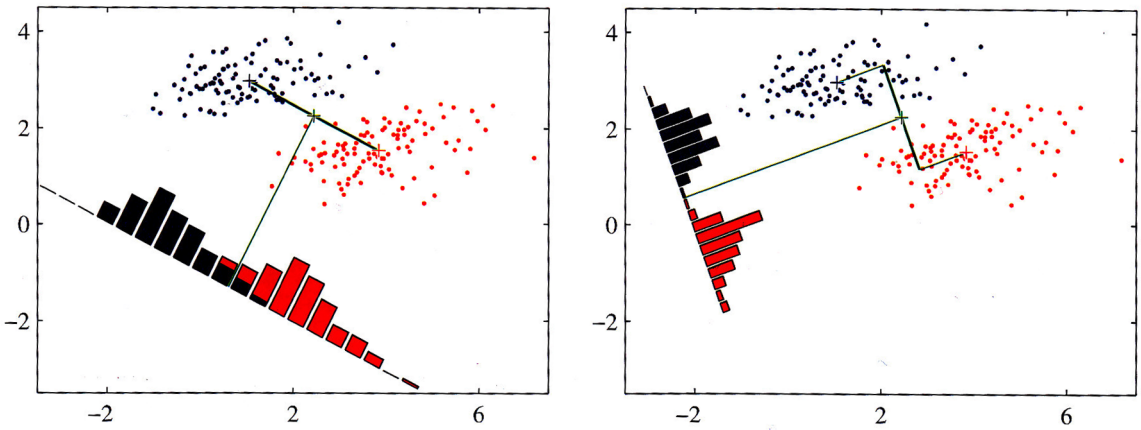

Abbildung 19: Dimensionsreduktion [Bishop, 2007]

Nebenbedingung $g(x)$ in der Form $g(x) = 0$ ausgedrückt werden kann, können wir den Extremwert ermitteln.
Wir schreiben:

$$\begin{gathered} f(x) \\ \text{subject to } g(x) = 0 \end{gathered}$$

Nun können wir die Lagrange Funktion wie folgt anschreiben:

$$\mathcal{L}(x, \lambda) = f(x) + \underbrace{\lambda g(x)}_{=0 \text{ (durch Nebenbedingung)}} \tag{30}$$

λ ist der Lagrange Multiplier. Um nun den Extremwert zu erhalten, müssen wir den Gradient (∇) betrachten. Dies geschieht mittels Ableitung von (30) nach x.

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla_x f(x) + \lambda \nabla_x g(x) =$$

$$\frac{\partial \mathcal{L}(x, \lambda)}{\partial x} = \frac{\partial f(x)}{\partial x} + \lambda \frac{\partial g(x)}{\partial x} = 0 \tag{31}$$

Wenn wir nun (31) nach λ und x lösen erhalten wir eine Menge von kritischen Punkten. Nun gilt es nur noch diese Punkte richtig zu identifizieren, um den von uns gesuchten Extremwert x_0 zu finden [Duda et al., 2001].

Abbildungsverzeichnis

Literatur

[Bishop, 2007] Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing 2011 edition.

[Duda et al., 2001] Duda, R., Hart, P., and Stork, D. (2001). *Pattern classification*. Pattern Classification and Scene Analysis: Pattern Classification. Wiley.

[Köhler, 2005] Köhler, B.-U. (2005). *Konzepte der statistischen Signalverarbeitung*. Springer, Berlin [u.a.].

[Learning, 2013] Learning, C. M. (2013). Dimensionality reduction and principal component analysis. `https://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Lectures.PCA`. Letzter Aufruf am: 10.06.2013.

[Pernkopf, 2013] Pernkopf, F. (2013). Graphical models. `https://www.spsc.tugraz.at/sites/default/files/CI_EW_GMs.pdf`. Letzter Aufruf am: 10.06.2013.

[Pernkopf et al., 2013] Pernkopf, F., Peharz, R., and Tschiatschek, S. (2013). Introduction to probabilistic graphical models. `https://www.spsc.tugraz.at/sites/default/files/PGM.pdf`. Letzter Aufruf am: 10.06.2013.

[Rank and Pernkopf, 2004] Rank, E. and Pernkopf, F. (2004). Hidden markov models. `https://www.spsc.tugraz.at/sites/default/files/Specomm04_3.pdf`. Letzter Aufruf am: 10.06.2013.

[Wikipedia, 2013] Wikipedia (2013). Mixture distribution. `http://en.wikipedia.org/wiki/Mixture_distribution`. Letzter Aufruf am: 06.06.2013.

[Wohlmayr, 2013] Wohlmayr, M. (2013). Blind source separation: Eine einführung. `http://www.spsc.tugraz.at/system/files/bss.pdf`. Letzter Aufruf am: 10.06.2013.