10.5 Maximum-Likelihood Klassifikation (I)

Idee

- Für die Klassifikation sind wir interessiert an den bedingten Wahrscheinlichkeiten $p(C_i(x,y)|D(x,y))$.
- Wenn man diese bedingten Wahrscheinlichkeiten kennt, dann ordnet man einem Pixel (x,y) mit Grauwertvektor D(x,y) die Klasse C_j mit dem maximalen Wert für diese Wahrscheinlichkeit zu:

  Bayes'sche Entscheidungsregel

$$(1)\ \forall i \neq j [p(C_j(x,y)|D(x,y)) > p(C_i(x,y)|D(x,y))] \Rightarrow C_j(x,y)$$

Maximum Likelihood Entscheidungsregel

- Die obigen bedingten Wahrscheinlichkeiten sind meist a priori unbekannt.
- Die umgekehrten bedingten Wahrscheinlichkeiten $p(D(x,y)|C_i(x,y))$. lassen sich jedoch aus den Trainingsdaten schätzen.

10.5 Maximum-Likelihood Klassifikation (II)

Maximum Likelihood Entscheidungsregel (Fortsetzung)

- Die gesuchten Wahrscheinlichkeiten lassen sich daraus mit Hilfe des *Bayes'schen Theorems* folgendermassen berechnen:

$$(2)\ p\,(C_i(x,y)|D\,(x,y)) = p\,(D\,(x,y)|C_i(x,y)|C_i(x,y) \cdot p\,(C_i(x,y))/p\,(D\,(x,y))$$

  wobei $p(C_i(x,y))$ die unbedingte Wahrscheinlichkeit bzw. die Häufigkeit der Klasse C_i im Bild ist und $p(D(x,y))$ die Häufigkeit des Grauwertvektors $D(x,y)$.
- Einsetzen von (2) in (1) liefert die *Maximum Likelihood Entscheidungsregel:*

$$(3)\ \forall i \neq j\ [p(D(x,y)|C_j(x,y)) \cdot p(C_j(x,y)) > p(D(x,y)|C_i/x,y)) \cdot p\,(C_i\,(x,y))] \Rightarrow C_j(x,y)$$

- Die $p(D(x,y))$, die schwer zu bestimmen sind, kürzen sich dabei heraus.
- (3) lässt sich mit der Definition

$$g_{i(D(x,y))} = \ln(p(D(x,y)|C_j(x,y))) + \ln(p(C_j(x,y)))$$ so umschreiben:

$$(4)\ \forall i \neq j [g_{j(D(x,y))} > g_{i(D(x,y))}] \Rightarrow C_j(x,y)$$

10.5 Normalverteilungen (I)

Schätzung der bedingten Wahrscheinlichkeiten

- Frage: Wie kann man aus den beobachteten *p(D(x,y)|*C_i*(x,y))* die bedingten Wahrscheinlichkeiten für alle *D(x,y)* und für alle C_i bestimmen?
- Methode: nehme an, dass die bedingten Wahrscheinlichkeiten *p(D(x,y)|*C_i*(x,y)) normalverteilt* sind.

10.5 Normalverteilungen (II)

Eindimensionaler Fall

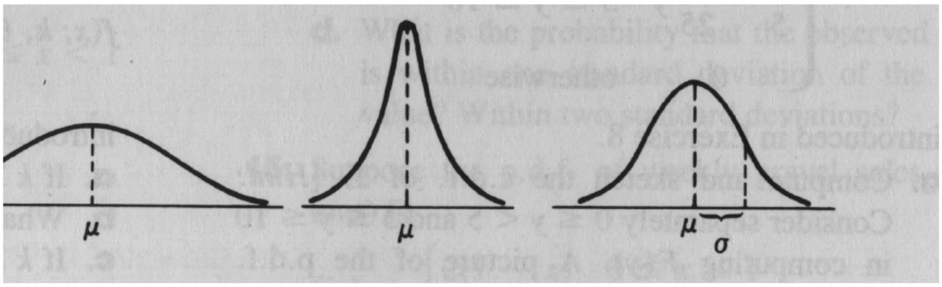
- Eindimensionaler Fall: m = 1, d.h. nur ein Grauwert pro Pixel. Dann gilt

$$p(D(x,y)|C_i(x,y)) = \frac{1}{\sqrt{2\pi}\sigma_i} \cdot e^{-(D(x,y)-m_i)^2/2\sigma_i^2}$$

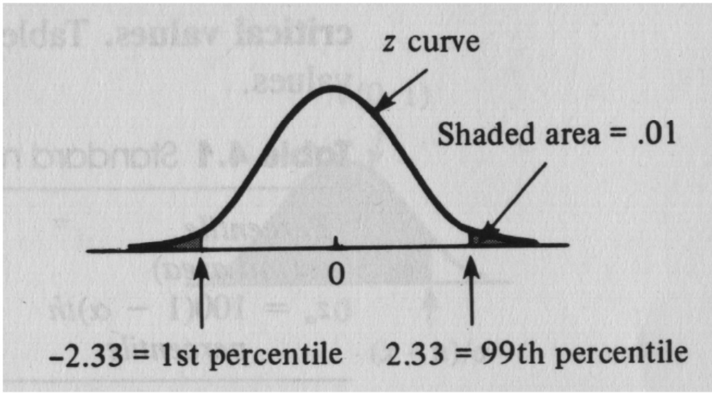
  mit m_i als Mittelwert und σ_i als Standardabweichung von *D(x,y)* für die Klasse C_i
- Annahme: es ist eine hinreichend große Menge von D(x,y)-Werten gegeben.
- Dann wird m_i geschätzt als Mittelwert der *D(x,y)*, σ_i^2 als Durchschnitt der Quadrate der Differenz der D(x,y) zu m_i.

10.5 Normalverteilungen (III)

Beispiel



μ = m: Mittelwert, σ: Standard-Abweichung



10.5 Normalverteilungen (IV)

Mehrdimensionaler Fall

- m > 1, d.h. mehrere Grauwerte pro Pixel.
- Dann ist *mi ein m-dimensionaler Vektor und σi 2 eine mxm Matrix Σi .*
- Die *Kovarianz-Matrix Σi für die Klasse Ci beschreibt die Korrelation zwischen den verschiedenen Spektralbereichen und ist definiert als:*

$$\Sigma^{jk}{}_i = E[((D_j(x,y) - m_{ij}) \cdot (D_k(x,y) - m_{ik}))]$$

- Sie wird geschätzt als:

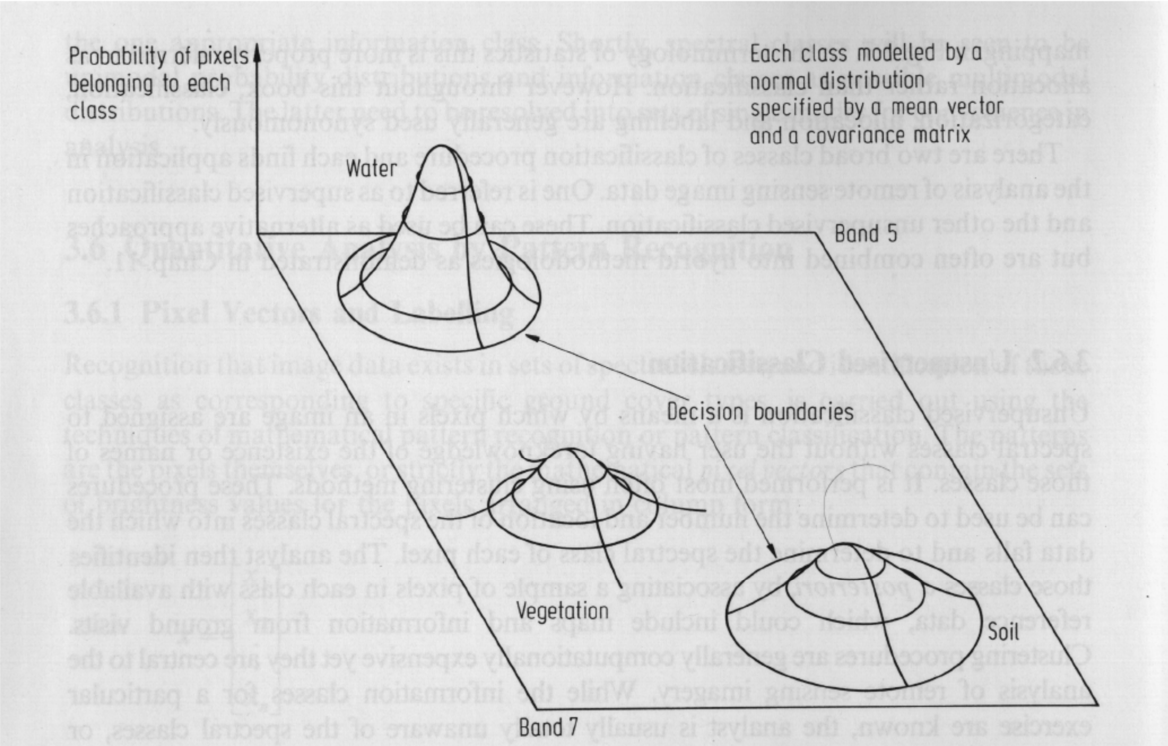
$$\Sigma^{jk}{}_i = \frac{1}{card(C_i) - 1} \cdot \sum_{(x,y) \in C_i} [((D_j(x,y) - m_{ij}) \cdot (D_k(x,y) - m_{ik}))]$$

- Es gilt:

$$(5)\ p(D(x,y)|C_i(x,y)) = \frac{1}{\sqrt{(2\pi)^m}\sqrt{|\Sigma_i|}} \cdot e^{\left\{-\frac{1}{2} \cdot ((D(x,y) - m_i)^T \cdot \Sigma_i^{-1} \cdot (D(x,y) - m_i))\right\}}$$

10.5 Normalverteilungen (V)

Beispiel



10.5 Maximum-Likelihood Klassifikation (III)

Endgültige Maximum-Likelihood Entscheidungsregel

- Durch Einsetzen von (5) erhalten wir

$$g_i(D(x,y)) = \ln(p(C_i(x,y))) + \ln(p(D(x,y)|C_i(x,y)))$$

$$g_i(D(x,y)) = \ln(p(C_i(x,y))) - \frac{1}{2} \cdot \ln|\Sigma i| - \frac{1}{2} \cdot ((D(x,y) - m_i)^T \cdot \Sigma i^{-1} \cdot (D(x,y) - m_i))$$

- Falls die $p(C_j(x,y))$ unbekannt sind, wird die Gleichverteilung aller Klassen C_i angenommen. Dann vereinfacht sich die Funktion g_i zu:

$$g_i(D(x,y)) = -\ln|\Sigma i| - ((D(x,y) - m_i)^T \cdot \Sigma i^{-1} \cdot (D(x,y) - m_i))$$

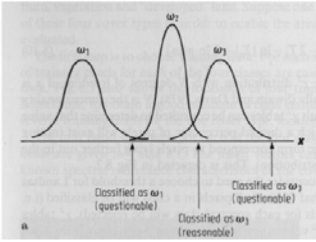
Anzahl der Trainingspixel

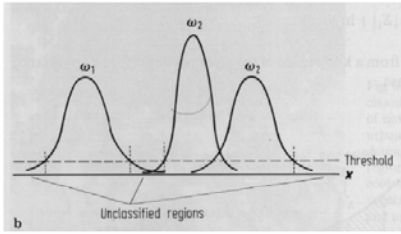
- Es werden *mindestens m + 1* Trainingpixel pro Klasse C_i benötigt, damit die Kovarianz-Matrix nicht singulär wird.
- Experimentelle Untersuchungen zeigen jedoch, dass in praktischen Anwendungen mindestens *10 m* und für gute Ergebnisse etwa *100 m* Trainingspixel pro Klasse nötig sind.

10.5 Maximum-Likelihood Klassifikation (IV)

Problem

- Nach der bisherigen Entscheidungsregel wird jedes Pixel einer Klasse zugeordnet, auch wenn die bedingte Wahrscheinlichkeit für diese Klasse sehr klein ist.
- Dieser Fall kann z.B. auftreten, wenn für eine Klasse nicht genügend Trainingsdaten vorhanden sind. Er führt leicht zu Fehlern in der Klassifikation (a).





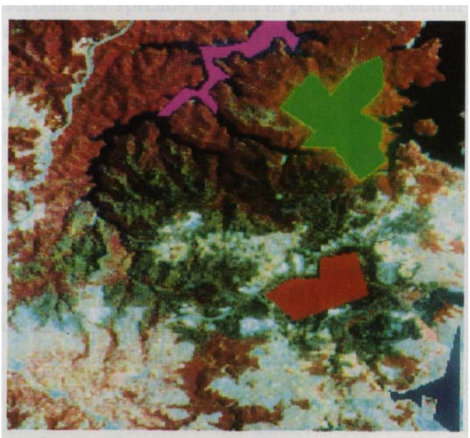
Einführung von Grenzwerten

- Wähle einen *Grenzwert (Threshold)* p_{min}.
- Pixel, deren Maximum-Likelihood-Wert nicht grösser als p_{min} ist, werden keiner Klasse zugeordnet (b).

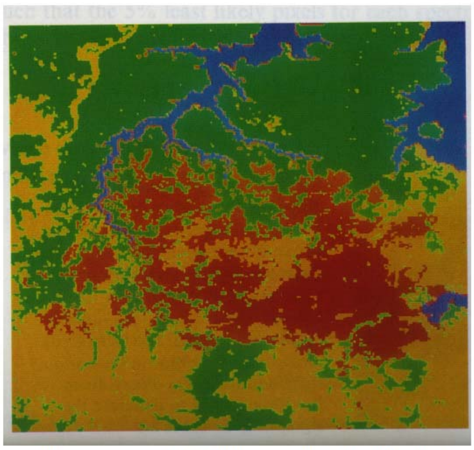
10.5 Maximum-Likelihood Klassifikation (V)

Beispiel

- Landsat Rasterbild 256 x 276 Pixel
- Vier Klassen: water, fire burn, vegetation, urban
- für jede Klasse ein Trainingsfeld (Menge räumlich benachbarter Trainingspixel



Eingangsbild mit Trainingsfeldern



Ausgangsbild (thematische Karte)

10.5 Maximum-Likelihood Klassifikation (VI)

Beispiel (Fortsetzung)

- Mittelwerte und Kovarianz-Matrizen für die vier Klassen

Class	Mean vector	Covariance matrix			
Water	44.27	14.36	9.55	4.49	1.19
	28.82	9.55	10.51	3.71	1.11
	22.77	4.49	3.71	6.95	4.05
	13.89	1.19	1.11	4.05	7.65
Fire burn	42.85	9.38	10.51	12.30	11.00
	35.02	10.51	20.29	22.10	20.62
	35.96	12.30	22.10	32.68	27.78
	29.04	11.00	20.62	27.78	30.23
Vegetation	40.46	5.56	3.91	2.04	1.43
	30.92	3.91	7.46	1.96	0.56
	57.50	2.04	1.96	19.75	19.71
	57.68	1.43	0.56	19.71	29.27
Developed	63.14	43.58	46.42	7.99	−14.86
(urban)	60.44	46.42	60.57	17.38	−9.09
	81.84	7.99	17.38	67.41	67.57
	72.25	−14.86	−9.09	67.57	94.27

- Tabellarisches Ergebnis der Klassifikation

Class	No. of pixels	Area (ha
Water	4830	2137
Fireburn	14182	6274
Vegetation	28853	12765
Developed (urban)	22791	10083

10.6 Nearest-Neighbor Klassifikation (I)

Motivation

- Die Maximum-Likelihood Klassifikation benötigt eine zuverlässige Schätzung der Mittelwerte und der Kovarianz-Matrizen für jede Klasse.
- Bei gleicher Menge von Trainingspixeln können die Mittelwerte zuverlässiger geschätzt werden als die Kovarianz-Matrizen.

  ⇒ entwickle Klassifikator, der nur Mittelwerte benutzt

Idee

- Bestimme die Mittelwerte m_i *für jede Klasse* C_i .
- Ordne jedes Pixel $D(x,y)$ der Klasse C_j mit dem nächstgelegen m_j (*Nearest Neighbor*) zu.

10.6 Nearest-Neighbor Klassifikation (II)

Nearest-Neighbor Entscheidungsregel

- Wir definieren

$$d(D(x,y), m_i) = (D(x,y) - m_i)^T \cdot (D(x,y) - m_i)$$

$$d(D(x,y), m_i) = D(x,y) \cdot D(x,y) - 2 \cdot m_i \cdot D(x,y) + m_i \cdot m_i$$

- Die *Nearest-Neighbor Entscheidungsregel* lautet wie folgt:

$$\forall i \neq j [g_{j(D(x,y))} > g_{i(D(x,y))}] \Rightarrow C_j(x,y)$$

mit

$$g_{i(D(x,y))} = 2 \cdot m_i \cdot D(x,y) - m_i \cdot m_i$$

10.6 Nearest-Neighbor Klassifikation (III)

Vergleich

- Nearest-Neighbor Klassifikation benötigt weniger Trainingspixel
- Die Trennflächen zwischen den *D(x,y)* verschiedener Klassen sind linear (Nearest-Neighbor) bzw. quadratisch (Maximum-Likelihood), so dass die Maximum-Likelihood Klassifikation bei Klassen mit konkaver Form eine wesentlich bessere Klassifikationsgüte liefert.
- Effizienz
  - Maximum-Likelihood Klassifikation:
    - $\ln(p(C_i(x,y))) - \frac{1}{2} \cdot \ln|\Sigma i|$ wird für jede Klasse vorberechnet
    - für jedes Pixel und für jede Klasse bleiben *m² + m Multiplikationen und m² + 2 m + 1* Additionen auszuführen
  - Nearest-Neighbor Klassifikation
    - $2\, m_i$ und $m_i * m_i$ werden für jede Klasse vorberechnet
    - für jedes Pixel und für jede Klasse *m* Multiplikationen und *m* Additionen

⇒ Nearest-Neighbor Klassifikation ist wesentlich effizienter