

# Andere Methoden zur Klassifikation und Objekterkennung

Heike Zierau

05. Juni 2007

1. Einführung
2. Prototypmethoden
  - ▶ K-means Clustering
  - ▶ Gaussian Mixture
  - ▶ Gaussian Mixture vs. K-means Clustering
3. nächste-Nachbarn Methoden
  - ▶ k-nächste-Nachbarn
  - ▶ Asymptotische Eigenschaften
  - ▶ Anwendung 1-nächster-Nachbar Methode
  - ▶ Adaptive nächste-Nachbarn Methoden
  - ▶ Rechenaufwand

## 1. Einführung

## 2. Prototypmethoden

- ▶ K-means Clustering
- ▶ Gaussian Mixture
- ▶ Gaussian Mixture vs. K-means Clustering

## 3. nächste-Nachbarn Methoden

- ▶ k-nächste-Nachbarn
- ▶ Asymptotische Eigenschaften
- ▶ Anwendung 1-nächster-Nachbar Methode
- ▶ Adaptive nächste-Nachbarn Methoden
- ▶ Rechenaufwand

# 1. Einführung

*Bisher betrachtete Methoden:*

- ▶ Daten an vorgegebenes Modell anpassen

*Eigenschaften der hier vorgestellten Methoden:*

- ▶ einfache, modellfreie Methoden zur Klassifikation und Objekterkennung
- ▶ Klasseneinteilung nicht immer nachvollziehbar
- ▶ als Black-Box - Anwendung sehr effektiv, liefern gute Ergebnisse

# Definitionen

- ▶ *Trainingsdaten*:  $N$  Paare  $(x_1, g_1), \dots, (x_N, g_N)$
- ▶  $x_i$ : *Merkmal*, für  $i \in \{1, \dots, N\}$   
 $g_i$ : *Klassenbezeichnung*, mit  $g_i \in \{1, \dots, K\}$ , für  $i \in \{1, \dots, N\}$
- ▶ *Prototyp*: ein Paar  $(x_k, g_k)$ , wobei normalerweise  $k \notin \{1, \dots, N\}$
- ▶ *“am nächsten”*: euklidischer Abstand im Merkmalsraum bei standardisierten Merkmalen, d.h. Erwartungswert 0 und Varianz 1
- ▶ die Dimension des Merkmalsraums entspricht der Anzahl der Merkmale  $x_i$

## 1. Einführung

## 2. Prototypmethoden

- ▶ K-means Clustering
- ▶ Gaussian Mixture
- ▶ Gaussian Mixture vs. K-means Clustering

## 3. nächste-Nachbarn Methoden

- ▶ k-nächste-Nachbarn
- ▶ Asymptotische Eigenschaften
- ▶ Anwendung 1-nächster-Nachbar Methode
- ▶ Adaptive nächste-Nachbarn Methoden
- ▶ Rechenaufwand

## 2. Prototypmethoden

- ▶ Prototyp-Methoden repräsentieren die Trainingsdaten im Merkmalsraum
- ▶ neue Daten können anhand der Prototypen einfach und schnell klassifiziert werden
- ▶ Entscheidungsgrenzen werden durch Prototypen festgelegt

1. Einführung
2. Prototypmethoden
  - ▶ **K-means Clustering**
  - ▶ Gaussian Mixture
  - ▶ Gaussian Mixture vs. K-means Clustering
3. nächste-Nachbarn Methoden
  - ▶ k-nächste-Nachbarn
  - ▶ Asymptotische Eigenschaften
  - ▶ Anwendung 1-nächster-Nachbar Methode
  - ▶ Adaptive nächste-Nachbarn Methoden
  - ▶ Rechenaufwand

# K-means Clustering

## *Idee:*

- ▶ Klassifikation einer Datenmenge durch Häufungen, d.h.
- ▶ für eine Datenmenge eine bestimmte Anzahl von Häufungszentren/Prototypen definieren

## *Ziel:*

- ▶ iterativ Abstand zwischen Merkmal und Häufungszentrum minimieren

# unmarkierte Datenmenge

## *Iterationsschritte:*

1. gewünschte Anzahl von Startzentren - z.B.  $R$  - zufällig setzen
2. Häufung konstruieren durch Punktemenge, die am nächsten zum Zentrum liegen
3. neues Zentrum der Häufung berechnen

Schritte 2 und 3 bis zur Konvergenz wiederholen.

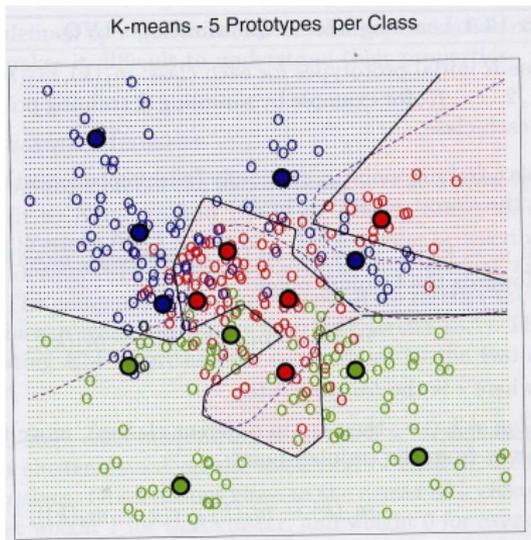
Divergenz theoretisch möglich wenn ein Punkt den gleichen Abstand zu zwei Zentren hat - Zentrum springt hin und her

# markierte Datenmenge

## *Iterationsschritte:*

1. K-means Clustering auf jede der  $K$  Klassen anwenden mit  $R$  Prototypen pro Klasse
2. jedem der  $K \cdot R$  Prototypen eine Klassenbezeichnung  $g_k$  und ein Merkmal  $x_k$  zuordnen
3. neue Daten werden der Klasse des nächsten Prototyps zugeordnet

# Beispiel



Simuliertes Beispiel mit drei Klassen  $g_i \in \{\text{rot, grün, blau}\}$  und  $R = 5$  Prototypen pro Klasse  
gestrichelte Linie ist Bayes'sche Entscheidungsgrenze

# Bewertung

- ▶ keine glatten Entscheidungsgrenzen
- ▶ einfache Möglichkeit Daten zu klassifizieren
- ▶ es treten Falschklassifikationen auf, besonders an den Klassenrändern
- ▶ Ergebnisse hängen von der Wahl und Anzahl der Startzentren ab

1. Einführung
2. Prototypmethoden
  - ▶ K-means Clustering
  - ▶ **Gaussian Mixture**
  - ▶ Gaussian Mixture vs. K-means Clustering
3. nächste-Nachbarn Methoden
  - ▶ k-nächste-Nachbarn
  - ▶ Asymptotische Eigenschaften
  - ▶ Anwendung 1-nächster-Nachbar Methode
  - ▶ Adaptive nächste-Nachbarn Methoden
  - ▶ Rechenaufwand

# Gaussian Mixture

*Idee:*

- ▶ jede Häufung kann durch eine parametrische Verteilung dargestellt werden, z.B. Normalverteilung
- ▶ Datenmenge wurde durch Mischung dieser Verteilungen erzeugt, jede Häufung hat eine andere Dichte
- ▶ wie bei K-means Clustering: Häufungszentren finden

# Modell

- ▶ *Annahme*: es gebe  $K$  Häufungen
- ▶ jede Häufung entstand durch eine Normalverteilung mit den Parametern  $\mu_k, \Sigma_k$
- ▶ Daten sind Vektoren  $X, X \in \mathbb{R}^N$
- ▶ gegeben sind  $n$  konkrete Daten  $x_1, \dots, x_n$

# Modell

- ▶ Dichte der Häufung  $k$ :

$$f_k(x) = \phi(x; \mu_k, \Sigma_k)$$

- ▶ a priori Wahrscheinlichkeit von  $k$  ist  $\alpha_k$ , wobei  $\sum_{k=1}^K \alpha_k = 1$
- ▶ Dichte der Mischung

$$f(x) = \sum_{k=1}^K \alpha_k f_k(x)$$

# Iterationsschritte des EM-Algorithmus

1. *Initialisierung*
2. *Schätzschritt bei Iteration  $p$* : Jeder Beobachtung einer Klasse eine Gewichtung zuordnen, bzw. a posteriori Wahrscheinlichkeiten berechnen

$$p_{i,k} = \frac{\alpha_k^{(p)} \phi(x_i; \mu_k^{(p)}, \Sigma_k^{(p)})}{\sum_{k=1}^K \alpha_k^{(p)} \phi(x_i; \mu_k^{(p)}, \Sigma_k^{(p)})}, i \in \{1, \dots, n\}, k \in \{1, \dots, K\}$$

3. *Maximierungsschritt*: a priori Wahrscheinlichkeiten, Erwartungswert und Kovarianzmatrix aktualisieren

$$\alpha_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k}}{n}$$

$$\mu_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k} x_i}{\sum_{i=1}^n p_{i,k}}$$

$$\Sigma_k^{(p+1)} = \frac{\sum_{i=1}^n p_{i,k} (x_i - \mu_k^{(p+1)})(x_i - \mu_k^{(p+1)})^t}{\sum_{i=1}^n p_{i,k}}$$

4. Schritt 2 und 3 bis zur Konvergenz wiederholen

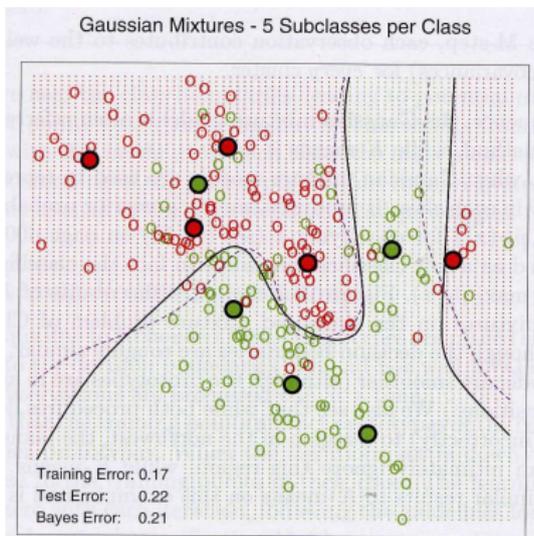
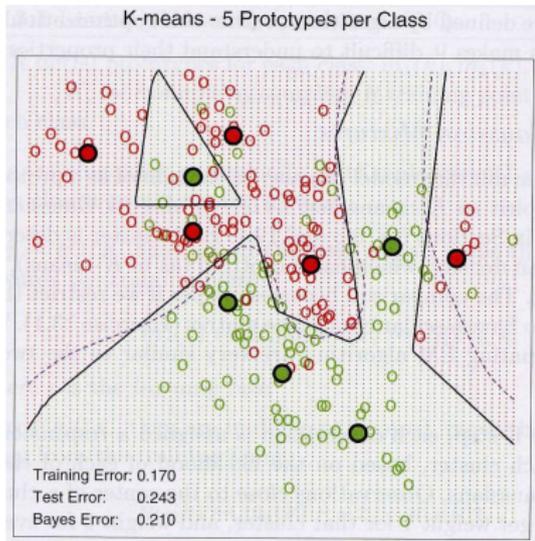
# Bewertung

- ▶ Gaussian Mixture “weich”, K-means “hart”
- ▶ Gaussian Mixture repräsentiert die Merkmalsdichte jeder Klasse
- ▶ glatte a posteriori Wahrscheinlichkeiten  
 $\hat{p}(x) = (\hat{p}_1(x), \dots, \hat{p}_K(x))^T$  zur Klassifikation von  $x$
- ▶ Klassifikationsregel  $\hat{G}(x) = \operatorname{argmax}_k \hat{p}_k(x)$ , wobei  $\hat{G}(x)$  die geschätzte Klasse für  $x$  bezeichnet

1. Einführung
2. Prototypmethoden
  - ▶ K-means Clustering
  - ▶ Gaussian Mixture
  - ▶ **Gaussian Mixture vs. K-means Clustering**
3. nächste-Nachbarn Methoden
  - ▶ k-nächste-Nachbarn
  - ▶ Asymptotische Eigenschaften
  - ▶ Anwendung 1-nächster-Nachbar Methode
  - ▶ Adaptive nächste-Nachbarn Methoden
  - ▶ Rechenaufwand

# Gaussian Mixture vs. K-means-Clustering

## Vergleich:



Entscheidungsgrenzen sehr ähnlich, aber Gaussian Mixture glatter.  
Gaussian Mixture ignoriert Region links oben, K-means nicht.

## Zu Beispiel

- ▶ Bei sehr geringer Merkmalsdichte kann Gaussian Mixture Region ignorieren:
- ▶  $p_k(x)$ ,  $k = \text{grün}$  wird überdeckt von den Wahrscheinlichkeiten für Klasse rot
- ▶ K-means-Clustering: Es tritt grüne Häufung auf  $\Rightarrow$  Prototyp setzen  $\Rightarrow$  Entscheidungsgrenze festlegen

## 1. Einführung

## 2. Prototypmethoden

- ▶ K-means Clustering
- ▶ Gaussian Mixture
- ▶ Gaussian Mixture vs. K-means Clustering

## 3. nächste-Nachbarn Methoden

- ▶ k-nächste-Nachbarn
- ▶ Asymptotische Eigenschaften
- ▶ Anwendung 1-nächster-Nachbar Methode
- ▶ Adaptive nächste-Nachbarn Methoden
- ▶ Rechenaufwand

### 3. nächste-Nachbarn Methode

*Idee:*

- ▶ Datenmenge klassifizieren in Abhängigkeit der nächsten Nachbarn
- ▶ Klassifizierer sind erinnerungsbasiert und benötigen kein anzupassendes Modell

## 1. Einführung

## 2. Prototypmethoden

- ▶ K-means Clustering
- ▶ Gaussian Mixture
- ▶ Gaussian Mixture vs. K-means Clustering

## 3. nächste-Nachbarn Methoden

- ▶ **k-nächste-Nachbarn**
- ▶ Asymptotische Eigenschaften
- ▶ Anwendung 1-nächster-Nachbar Methode
- ▶ Adaptive nächste-Nachbarn Methoden
- ▶ Rechenaufwand

# k-nächste-Nachbarn Methode

*Vorgehensweise:*

1. noch nicht klassifizierter Punkt  $x_0$  gegeben
2. finde  $k$  Trainingspunkte  $x_r$ ,  $r = 1, \dots, k$ , mit kleinstem euklidischen Abstand zu  $x_0$
3. Klassifikation durch Mehrheitswahl der  $k$  Nachbarn, d.h.  $x_0$  der Klasse zuordnen, in der die Mehrheit der  $k$  Nachbarn von  $x_0$  enthalten sind.

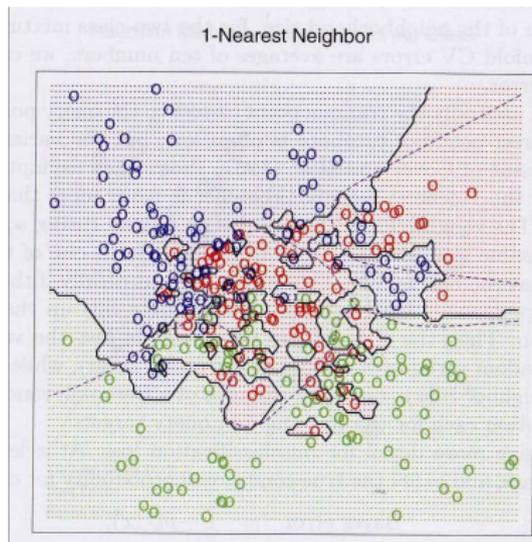
# Eigenschaften

- ▶ k-nächste-Nachbarn Methode erfolgreich bei vielen Klassifikationsproblemen, z.B. handgeschriebene Ziffern, Satellitenbilder, EKG-Bilder
- ▶ liefert gute Ergebnisse bei vielen möglichen Prototypen pro Klasse und unregelmäßigen Entscheidungsschranken.
- ▶ Beziehung zwischen k-nächste-Nachbarn und Prototyp-Methoden: bei 1-nächster-Nachbarn-Klassifikation ist jeder Trainingspunkt ein Prototyp.

# Bias und Varianz

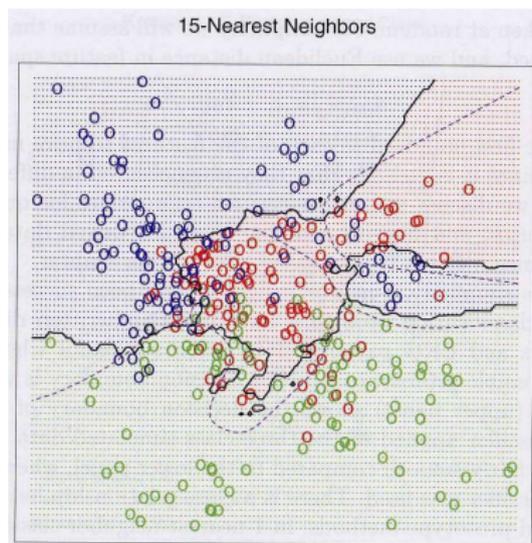
- ▶ *Bias*: ein Trainingspunkt wird falsch klassifiziert
- ▶ *Varianz*: ein Testpunkt wird falsch klassifiziert

# 1 nächster Nachbar



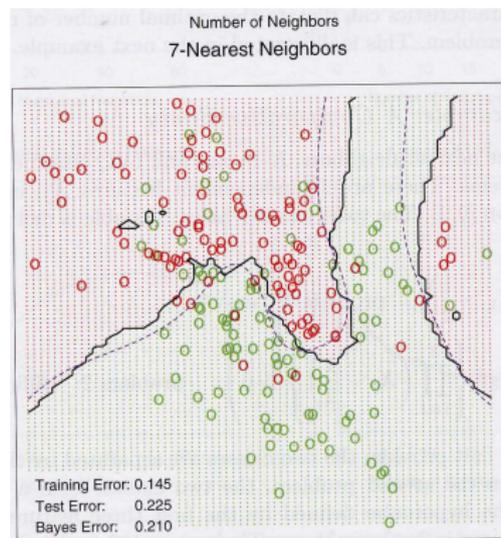
Überklassifikation, aber keine Falschklassifikationen der Trainingsdaten: kleiner Bias, große Varianz

## 15 nächste Nachbarn



Häufig Falschklassifikationen der Trainingsdaten: großer Bias, dafür kleine Varianz

# 7 nächste Nachbarn



Scheint optimal zur Minimierung des Testfehlers

## 1. Einführung

## 2. Prototypmethoden

- ▶ K-means Clustering
- ▶ Gaussian Mixture
- ▶ Gaussian Mixture vs. K-means Clustering

## 3. nächste-Nachbarn Methoden

- ▶ k-nächste-Nachbarn
- ▶ **Asymptotische Eigenschaften**
- ▶ Anwendung 1-nächster-Nachbar Methode
- ▶ Adaptive nächste-Nachbarn Methoden
- ▶ Rechenaufwand

# Asymptotische Eigenschaften

- ▶ Bias der 1-nächste-Nachbarn Methode klein
- ▶ Klasseneinteilung genau auf diese Trainingsdaten abgestimmt
- ▶ Falschklassifikation eines Testpunktes sehr wahrscheinlich, d.h. große Varianz

# Ergebnis von Cover und Hart

asymptotisch gilt:

$$1 - \frac{\text{Fehlerrate}}{\text{naechster - Nachbarn Klassifikator}} \leq 2 \cdot \text{Bayessche Fehlerrate}$$

# Bewertung

Sei  $p_k(x)$  die Wahrscheinlichkeit, dass  $x$  in der Klasse  $k$  liegt.

Sei  $k^*$  die dominante Klasse der Nachbarn von  $x$ , d.h.

$p_{k^*}(x) \geq p_k(x), \forall k = 1, \dots, K$ , mit  $k \neq k^*$

Dann gilt asymptotisch:

$$\text{Bayes Fehler} = 1 - p_{k^*}(x)$$

$$1 - \text{naechster - Nachbar - Fehler} = \sum_{k=1}^K p_k(x)(1 - p_k(x))$$

desweiteren gilt asymptotisch:

$$1 - p_{k^*}(x) \leq \sum_{k=1}^K p_k(x)(1 - p_k(x)) \leq 2(1 - p_{k^*}(x))$$

- ▶ die Darstellung der Fehlerraten wurde im Paper von Cover und Hart von 1967 bewiesen

# Beweis

(i)

$$\begin{aligned}\sum_{k=1}^K p_k(x)(1 - p_k(x)) &= \sum_{k=1}^K p_k(x) - \sum_{k=1}^K p_k(x)^2 \\ &\geq 1 - p_{k^*}(x) \sum_{k=1}^K p_k(x) \\ &= 1 - p_{k^*}(x)\end{aligned}$$

# Beweis

(ii)

$$\begin{aligned}\sum_{k=1}^K p_k(x)(1 - p_k(x)) &= p_{k^*}(x)(1 - p_{k^*}(x)) + \sum_{k \neq k^*} p_k(x)(1 - p_k(x)) \\ &\leq (1 - p_{k^*}(x)) + (1 - p_{k^*}(x)) - \sum_{k \neq k^*} p_k(x)^2 \\ &\leq 2(1 - p_{k^*}(x))\end{aligned}$$

1. Einführung
2. Prototypmethoden
  - ▶ K-means Clustering
  - ▶ Gaussian Mixture
  - ▶ Gaussian Mixture vs. K-means Clustering
3. nächste-Nachbarn Methoden
  - ▶ k-nächste-Nachbarn
  - ▶ Asymptotische Eigenschaften
  - ▶ **Anwendung 1-nächster-Nachbar Methode**
  - ▶ Adaptive nächste-Nachbarn Methoden
  - ▶ Rechenaufwand

# Anwendung 1-nächste-Nachbarn Methode



- ▶ bei handgeschriebenen Ziffern häufig kleine Veränderungen, z.B. kleine Rotationen - kein Problem für menschliches Auge
- ▶ Graustufenwerte der Pixel eines rotierten und nichtrotierten Bildes unterscheiden sich stark

# Aufbau

- ▶ Merkmalsraum mit 256 Dimensionen (16x16 Pixelbilder)
- ▶ eine Dimension entspricht einem Pixel, also einem Merkmal
- ▶ ein Pixel kann Graustufenwerte aus  $\{1, \dots, 1024\}$  annehmen
- ▶ ein Punkt im Merkmalsraum ist ein 256-dimensionaler Vektor und repräsentiert die Ziffer 3
- ▶ zwei Punkte gehören zu einer Klasse, wenn sich ihre Bilder nur durch eine Rotation unterscheiden

# Kurvenvergleich

## *Idee:*

- ▶ Graustufenwerte der Pixel verändern sich stetig bei der Rotation - glatte Kurve im Merkmalsraum
- ▶ bei Rotation um  $360^\circ$  liegen originale und rotierte Ziffer auf einer Kurve

## *Problem:*

- ▶ hoher Rechenaufwand
- ▶ Unterscheidung von „6“ und „9“

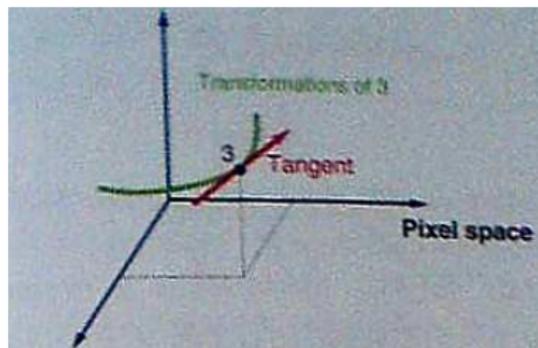
# invariante Metrik

- ▶ durch Rotation verändert sich der euklidische Abstand zwischen den Bildern
- ▶ kleinsten Abstand zwischen zwei Kurven bestimmen ungenau  
⇒ der euklidische Abstand im  $\mathbb{R}^{256}$  kann sehr groß sein
- ▶ die Rotationskurve wird als invariante Metrik bezeichnet
- ▶ Metrik  $d$  invariant, falls  $d(x, y) = d(x + a, y + a)$  für alle  $x, y, a$  im Merkmalsraum

# Tangente

- ▶ bei handgeschriebenen Ziffern treten im Normalfall nur kleine Rotationen auf
- ▶ zum Vergleich zweier Bilder kleine Rotationen durchführen
- ▶ Tangente im Originalbild an Kurve legen
- ▶ Tangente approximiert invariante Kurve

# Tangente



- ▶ die Rotation beschreibt glatte Kurve
- ▶ Tangente im Punkt  $x$  an die Kurve legen

# Vorgehensweise

1. an Originalbild kleine Rotation durchführen
2. Tangente im Bild an Rotationskurve legen
3. „ähnlichste“ Tangente aus Tangenten der Trainingsmenge finden - z.B. gleiche Richtung, gleicher Winkel
4. das Bild mit der „ähnlichsten“ Tangente gehört zur gleichen Klasse

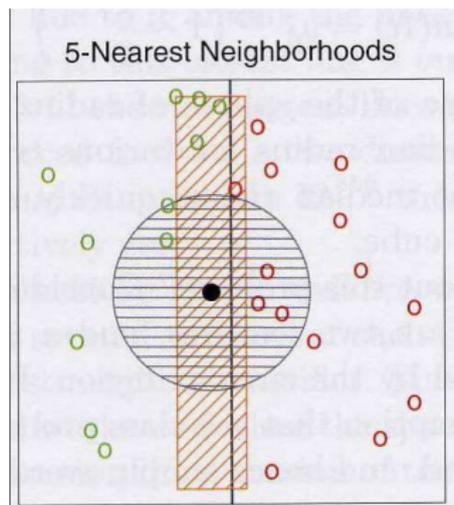
# Fehlerraten

- ▶ sehr kleine Fehlerraten, vergleichbar mit dem menschlichen Auge.
- ▶ Für ein Problem mit 7291 Trainingsbildern und 2007 Testzahlen ergaben sich folgende Fehlerraten:

Methode	Fehler
neuronales Netzwerk	0,049
1-Nächster-Nachbar/euklidischer Abstand	0,055
1-Nächster-Nachbar/Tangentenabstand	0.026

1. Einführung
2. Prototypmethoden
  - ▶ K-means Clustering
  - ▶ Gaussian Mixture
  - ▶ Gaussian Mixture vs. K-means Clustering
3. nächste-Nachbarn Methoden
  - ▶ k-nächste-Nachbarn
  - ▶ Asymptotische Eigenschaften
  - ▶ Anwendung 1-nächster-Nachbar Methode
  - ▶ **Adaptive nächste-Nachbarn Methoden**
  - ▶ Rechenaufwand

# Adaptive nächste-Nachbarn Methoden



Nachbarschaft eines Randpunktes

# Problem

- ▶ zu klassifizierender Punkt  $x_0$  liegt in grüner Klasse, jedoch drei der fünf Nachbarn sind rot  
⇒ nach Mehrheitswahl unter den Nachbarn gehört  $x_0$  zur Klasse rot
- ▶ allgemein: k-nächste-Nachbarn Methode unpraktisch bei begrenztem Trainingsdatenumfang und höherdimensionalen Merkmalsräumen  
⇒ häufige Falschklassifikationen

# Mahalanobis-Distanz

## *Lösung*

- ▶ in höherdimensionalen Merkmalsräumen: Punkte als Realisierungen von Zufallsvektoren auffassen
- ▶ Abstand zweier Punkte bestimmen durch Mahalanobis-Distanz

$$d(\underline{x}, \underline{y}) = \sqrt{(\underline{x} - \underline{y})^T \Sigma^{-1} (\underline{x} - \underline{y})}$$

- ▶ sie ergibt sich annäherungsweise durch logarithmieren der Dichte der multivariaten Normalverteilung mit  $\underline{y}, \Sigma$

# Graphische Darstellung

- ▶ die Mahalanobis-Distanz ist skaleninvariant, d.h.  
 $f(ax_1, ax_2, \dots, ax_n) = c(a)f(x_1, x_2, \dots, x)$
- ▶ sie ist translationsinvariant
- ▶ graphisch: bei gleicher Mahalanobis-Distanz zweier Punkte zum Mittelpunkt entsteht eine Ellipse
- ▶ bei euklidischem Abstand entsteht ein Kreis
- ▶ euklidischer Abstand und Mahalanobis-Distanz sind gleich, wenn  $\Sigma = I$  der Einheitsmatrix

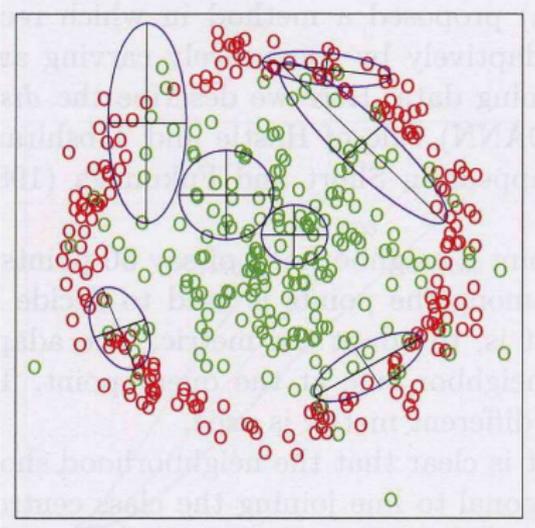
# Anwendung

- ▶ Berechnung der Erwartungswerte  $\mu_1, \mu_2$  und Kovarianzmatrix  $\Sigma$  der beiden Datensätze
- ▶ Berechnung der Mahalanobis-Distanz eines Punktes  $z$  zu den Mittelpunkten der Datensätze
- ▶  $z$  gehört zu der Klasse mit dem kleineren Abstand zum Mittelpunkt

# Beispiel

- ▶ Generiere Zwei-Klassen-Daten
- ▶ Daten in Klasse 1 sind unabhängig standardnormalverteilt mit Nebenbedingung:  
Realisierungen nur auf einen Ring mit Radius  $r \in (a, b)$ ,  $a < b$  vom Mittelwert entfernt
- ▶ Daten in Klasse 2 sind unabhängig standardnormalverteilt ohne Nebenbedingung
- ▶ 250 Trainingsdaten pro Klasse

# Beispiel



Klasse 1 umrundet Klasse 2 fast vollständig

1. Einführung
2. Prototypmethoden
  - ▶ K-means Clustering
  - ▶ Gaussian Mixture
  - ▶ Gaussian Mixture vs. K-means Clustering
3. nächste-Nachbarn Methoden
  - ▶ k-nächste-Nachbarn
  - ▶ Asymptotische Eigenschaften
  - ▶ Anwendung 1-nächster-Nachbar Methode
  - ▶ Adaptive nächste-Nachbarn Methoden
  - ▶ **Bewertung**

# Bewertung

- ▶ einfache Methode
- ▶ keine Vorkenntnis über die Daten erforderlich
- ▶ liefert gute Ergebnisse z.B. beim Erkennen handgeschriebener Ziffern
- ▶ hat hohen Rechenaufwand für Auffinden der Nachbarschaften: bei  $N$  Beobachtungen und  $p$  Merkmalen sind  $Np$  Operationen nötig um Nachbarschaft für jeden Folgenpunkt zu finden

# Literatur

-  T. Hastie, R. Tibshirani, J. Friedman. "The elements of statistical learning." Springer, 2001, Kap. 13
-  <http://en.wikipedia.org>
-  <http://www.springerlink.de>
-  [http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial\\_html/AppletKM.html](http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html)
-  T.M. Cover, P.E. Hart, IEEE Transactions on Information Theory, Vol. IT-13, No. 1, Januar 1967

Vielen Dank für eure Aufmerksamkeit!