Zusammenfassung - Mustererkennung und Kontextanalyse

Andreas Ruscheinski, Marc Meier

24. Januar 2016

Korrektheit und Vollständigkeit der Informationen sind nicht gewährleistet. Macht euch eigene Notizen oder ergänzt/korrigiert meine Ausführungen!

Inhaltsverzeichnis

1 Überblick Klassifikation

1.1 Einführendes Beispiel

- Ziel: Bestimmung von Fischen auf der Basis von Kamerainformationen
- Verwendung der Kamera zum Merkmale (Features) bestimmen des aktuellen Fisches: Länge, Helligkeit und Breite
- Annahme: Die Modelle (Beschreibung) der Fische unterscheiden sich
- Klassifikation: Finde zu gegebenen Merkmalen das am besten passende Modell (Welche Beschreibung der Fische passt am besten zu dem aktuellen Fisch)

1.2 Entscheidungsgrenzen

Komplexe Entscheidungsgrenze ⇒ Mehr Parameter werden für die Bestimmung benötigt ⇒ Weniger Trainingsdaten stehen zur Bestimmung des einzelnen Parameters zur Verfügung ⇒ Parameter wird ungenauer bestimmt und ist anfälliger gegen Schwankungen der Trainingsdaten ⇒. Mehr Features ⇒ höhere Dimensionalität des Merkmalsraums ⇒ größere inhärente Komplexität der gegebenen Form von Entscheidungsgrenzen ⇒ schlechtere Bestimmung der Parameter

1.3 Mustererkennungssysteme

1) Sensing: Erfassung der Umwelt mittels Sensoren, z.B: Kamera, Bewegungssensoren, Mikrophon, RFID-Lesegerät, Problem: Eigenschaften und Begrenzungen (Bandbreite, Auflösung, Empfindlichkeit, Verzerrung, Rauschen, Latenz, ...) des Sensors beeinflussen Problemschwierigkeit

2) Segmentation: Identifikation der einzelnen Musterinstanzen (Identifikation der für unser Problem relevanten Daten), z.B: Fisch auf dem Fließband

3) Merkmalsberechnung: Bestimmung von Features, gesucht sind dabei Features, welche eine Diskrimierungsfähigkeit haben (Zwischen zwei gleichen Klassen ähnlicher Wert, zwischen zwei verschiedenen Klassen großer Werteunterschied) und Invariant gegenüber Signaltransformationen (Rotation, Translation, Skalierung, perspektivische Verzerrung) sind; Problem: Wie kann ich aus einer großen Auswahl an Merkmalen die am besten geeigneten finden?

4) Klassifikation: Annahme: Modelle (Grundlegende Eigenschaften) der Klassen (verschiedene Fische) unterscheiden sich; Ziel: Zuweisung von Probleminstanzen aufgrund deren Merkmale zu der am besten entsprechenden Klasse

5) Nachbereitung: Entscheidung auf Basis Klassifikation, Problem: Wie können wir Kontextinformationen nutzen? Können wir verschiedene Klassifikatoren zusammen nutzen?

1.4 Entwurf von Mustererkennungssystemem

1) Daten sammlen: Wie kann man wissen, wann eine Menge von Daten ausreichend groß und repräsentativ ist, um das Klassifikationssystem zu trainieren und zu testen?

2) Merkmale bestimmen: Gesucht: Einfach zu extrahierende Merkmale mit hoher Diskriminativität, Invariant gegenüber irrelevanten Transformationen, unempfindlich gegenüber Rauschen, Problem: Wie kann ich A-priori-Wissen nutzen?

3) Modell auswählen: Wie erkennt man, wann ein Modell sich in der Klassifikation signifikant von einem anderen Modell – oder vom wahren Modell – unterscheidet? Wie erkennt man, dass man eine Klasse von Modellen zugunsten eines anderen Ansatzes ablehnen sollte? Versuch-und-Irrtum oder gibt es systematische Methoden?

4) Klassifikator trainieren: Verwende gesammelte Daten, um Parameter des Klassifikators zu bestimmen

5) Klassifikator evaluieren: Wie bewertet man die Leistung? Wie verhindert man Overfitting/Underfitting?

Für weitere Informationen sind folgende Referenzen zu konsultieren: [1, S. 3-16]

2 Grundlagen Signalverarbeitung

2.1 Digitalisierung

- Sensordaten liefern kontinuierliche Daten in der Regel Spannungswerte
- das Signal wird gesampelt d.h. zeitlich diskretisiert indem in regelmäßigen ein Wert gemessen wirds
- gesampelte Signal wird in der Amplituide diskretisiert (Analog-Digital-Converter) d.h. liefert anschließend ein digitalen Wert
- ADC setzt Bits entsprechend des Eingangssignals (1/2 Spannung, 1/4 Spannung)
- Problem: LSB im ADC wird nicht gesetzt wenn Wert nur gering Schwankt d.h. es gehen Informationen verloren
- Lösung: Dithering (Addierung von Noise so wird Grenzwert manchmal überschritten und Erwartungswert nährt sich realen Wsert an)
- Wie ist die Abtastrate zu setzen um den korrekten Wert aus den Originalsignalen zu erhalten?
- Antwort: Abtasttheorem „Ein Signal nur dann kann korrekt abgetastet werden, wenn es keine Frequenzanteile enthält die oberhalb der halben Abstastrate liegen.“ $f_{max} \leq \frac{1}{2} f_{sample}$

2.2 Lineare Systeme

- f ist linear gdw: $f(c * (a + b)) = c * (f(a) + f(b))$
- Homogenität: $f(c * a) = c * f(a)$ d.h. eine Veränderung des Input-Signales hat die selbe Auswirkung auf das Output-Signal
- Additivität: $f(a + b) = f(a) + f(b)$ d.h wenn das Input-Signal aus meheren Signalen zusammen gesetzt ist können wir diese einzelnt betrachten und addieren
- Translationsinvarianz: $f(n) = n \rightarrow f(n+s) = f(n)+s$ d.h. das Output-Signal ist invariant zum zeitlichen Verschub
- Kommutativität: $f(h(a))) = h(f(a))$ d.h. in einem zusammengesetzen System können die Bestandteile umgedreht werden
- ein lineares System ist vollständig durch seine Impulsantwort charakterisiert
- Impulsantwort erhalten wird durch Eingabe eines Signals welcher genau an einer Stelle den Wert 1 hat (Delta-Funktion), die Antwort beschreibt das gesamte Verhalten des Systems
- in einem linearen System können Signale zerlegt werden und analysiert werden ohne das Ergebnis zu beeinflussen $f(x) = f(x_1 + x_2 + x_3) = f(x_1) + f(x_2) + f(x_3)$
- ein Input-Signal kann also durch mehere Delta-Funktionen beschrieben werden, jede Delta-Funktion ausgewertet werden und die Ergebnisse wieder addiert werden ohne dass das Ergebnis verfälscht wird
- Ist die Eingabe ein sinusförmiges Signal dann ist auch die Ausgabe ein sinusförmiges Signal mit der gleichen Frequenz

2.3 Faltung

- $x \star h$ (x gefaltet an hs)
- zerlege $x[n]$ in δ-Impulse, Erzeuge für jeden δ-Impuls eine entsprechend verschobene und skalierte Kopie der Impulsantwort, Erzeuge Augangssignal durch summierten dieser Impulsantworten
- Faltung ist kommutativ: $x[n] \star h[n] = h[n] \star x[n]$
- Berechnung der Faltung: $y[i] = \sum_{j=1}^{M} h[j] * x[i - j]$
- Eigenschaften:
  - Kommutativität: $a[n] \star b[n] = b[n] \star a[n]$
  - Assoziativität: $(a[n] \star b[n]) \star c[n] = a[n] \star (b[n] \star c[n])$
  - Distributivität: $(a[n] \star b[n]) + (a[n] \star c[n]) = a[n] \star (b[n] + c[n])$
  - Transferenz: $f(x[n] \star h[n]) = f(x[n]) \star h[n]$
  - Zentraler Grenzwertsatz: $x \star x \star \cdots \star x \rightarrow N$

2.4 Korrelation

- Korrelation = Faltung mit der gespiegelten Impulsantwort (Target)
- $y[n] = x[n] \star t[-n]$ misst die Ähnlichkeit des Signals mit dem Target an allen Stellen des Signals
- Kreuzkorrelation = Ergebnis der Korrelation zweier Signale
- Autokorrelation = Korrelation eines Signals mit sich selbst
- Korrelation ist die optimale Methode zur Detektion eines bekannten Signals in Zufallsrauschen

2.5 Diskrete Fourier-Transformation

- jede peripdosche Funktion lässt sich als Summe von Sinus und Kosinusfunktionen darstellen
- Vorraussetzungen (Dirichlet-Bedingungen)
  - Anzahl der Unstetigkeiten innerhalb einer Periode ist endlich
  - Anzahl der Maxima und Minima innerhalb einer Periode ist endlich
  - Funktion ist in jeder Periode integrierbar (d.h. die Fläche unter dem Betrag der Fnktion ist in jeder Periode endlichs)
- die Basisfunktionen der Fouriertransformation sind orthogonal, bilden also die Basis eines Vektorraums
- Idee: jedes Signal kann als Summe von phasenverschobenen Cosinus-Wellen repräsentiert werden: $s[i] = \sum_{k=0}^{N/2} M_k cos(2\pi * k * i/N + \phi_k), i = 0, \ldots, N-1$
- jedes phasenverschobene Cosinus-Funktion $Mcos(x+\phi)$ kann durch Summe von Cosinus und Sinusfunktion ohne Phasenverschiebung repräsentiert werden: $M * cos(x + \phi) = Acos(x) + Bsin(x)$ wobei $A = M * cos\phi, B = M * sin\phi$
- Signal mit N Werten ein N-dimensionaler Vektore
- N-dimensionaler Vektorraum benötigt man eine Basis von N orthogonalen Vektor (Skalarprodukt zweier Basisvektoren muss verschwinden)
- Diskrete Cosinus und Sinusfunktionen $c_k[i] = cos(2\pi * k * i/N)$ und $s_m[i] = sin(2 * \pi * m * i/N)$ sind orthogonal
- die $N + 2$ diskreten Sinus und Kosinusfunktionen $c_k, s_k; k = 0, \ldots, N/2$ bilden N-dimensionale Basis da s_0 und $s_{N/2}$ sind Nullvektoren und nicht in der Basis
- Zeitbreich beschreibt das Input-Signal
- Frequenzbereich beinhaltet die Real(Kosinus) und Imaginärteile(Sinus) des Inputsignals
- Basisfunktionen:
  - Kosnius: $c_k[i] = cos(2 * \pi * k * i/N)$
  - Sinus: $s_k[i] = sin(2 * \pi * k * i/N)$
  - k ist die Wellenanzahl, gibt an wie viel Zyklen innerhalb der Signallänge enthalten sind
  - c_k ist das Signal der Cosinusfunktionen die mit der Amplitude $Re[k]$ in der Fourierzerlegung auftreten,
  - s_k ... $Im[k]$
- $c_0 = Re[0]$ ist der Gleichstromanteil
- s_0 und $s_{N/2}$ sind null
- Idee: Berechnung der Ähnlichkeit zweier Signale (Korreleation) $\sum_{i=0}^{N-1} x[i] * y[i]$
- Anwendung der Idee auf unsere $c_k[i]$ bzw. $s_k[i]$
- $ReX[k] = \sum_{i=0}^{N-1} x[i] * cos(2 * \pi * k * i/N)$ analog für $ImX[k]$ nur halt mit Sinus; $k = 0, \ldots, N/2$
- es gilt: $A * cos(x) + B * sin(x) = M * cos(x + \phi)$
- Übergang in Polarkoordinaten:
  - Magnitude: $\sqrt{ReX[k]^2 + ImX[k]^2}$
  - Phase: $arctan(\frac{ImX[k]}{ReX[k]})$
  - Vorteil: besser Verständlich für den Menschen um Charakteristiken zu verstehen
- Rekonsuktion des Signals aus ReX, ImX: $x[i] = \sum_{k=0}^{n/2}(Re\hat{X}[k] * cos(2 * \pi * k * i * /N)) + \sum_{k=0}^{n/2}(Im\hat{X}[k] * sin(2 * \pi * k * i * /N))$ mit
  - $Re\hat{X}[k] = \frac{ReX[k]}{N/2}$ für $0 < k < N/2$
  - $Im\hat{X}[k] = \frac{ImX[k]}{N/2}$

- $Re\hat{X}[0] = \frac{ReX[0]}{N}$
- $Re\hat{X}[N/2] = \frac{ReX[N/2]}{N}$

- DFT ist linear: $F(c * f(t)) = c * F(f(t))$ und $F(f(t) + g(t)) = F(f(t)) + F(g(t))$
- Faltungssatz: $F(x \star y) = F(x) * F(y)$ d.h. Faltung im Zeitbereich entspricht einer Multiplikation im Frequenzbereich
- dual: $F(x * y) = F(x) \star F(y)$

2.6 Anwendung

- Spektralanalyse
- nicht Spektrum des Gesamtsignals von Interesse sondern die Kurzzeitsprektren von Signalausschnitten
- Ansatz: Stanze nacheinander Fenster aus den Signalen und berechne jeweils das Sprektrum für ein solches Fensters, resultierendes Spektrum des Fensters repräsentziert die Frequenzverteilung zu dem Zeitintervall dass das Fenster überdeckt

3 Merkmale

3.1 Statistische Merkmale

3.1.1 Erwartungswert

Der Erwartungswert einer diskreten Zufallsvariable x mit den möglichen Werten x_i und zugehörigen Wahrscheinlichkeiten $P(x_i)$ ist: $E[x] = \sum_{i=1}^{n}(x_i * P(x_i))$.
Der Erwartungswert einer kontinuierlichen Zufallsvariable x mit Wertebreich X und zugehöriger Wahrscheinlichkeitsdichte $p(x)$ ist: $E[x] = \int_X (x * p(x))dx$.
Der Erwartungswert einer Zufallsvariable wird auch als Mittelwert μ bezeichnet.
Für gleichverteilte diskrete Zufallsvariablen x mit Werten $x_1, x_2, \ldots, x_n$ gilt: $P(x_i) = 1/n$ und somit: $E[x] = \sum_{i=1}^{n}(x_i * \frac{1}{n}) = \frac{1}{n} * \sum_{i=1}^{n} x_i$.
Für die Erwartungswerte von Funktionen einer Zufallsvariablen, $f(x)$ gilt: $E[f(x)] = \sum_{i=1}^{n}(f(x_i) * P(x_i))$ bzw. $E[f(x)] = \int_X (f(x) * p(x)dx$.

3.1.2 Momente

Der r-te Moment einer Zufallsvariable x ist $E[x^r]$. Der erste Moment mit $r = 1 : E[x]\mu$ heißt auch Mittelwert.
Der r-te zentrale Moment ist: $\mu_r = E[(x - E[x])^r] = E[(x - \mu)^r]$.
Das zweite zentrale Moment $\mu_2 = E[(x - \mu)^2] = \sigma^2$ heißt auch Varianz; $\sqrt{\sigma^2}$ heißt Standardabweichung σ.
Die Schiefe einer Verteilung ist ein Maß für ihre Asymmetrie: $skew(x) = \frac{\mu_3}{\sigma^3} = \frac{E[(x-\mu)^3]}{\sigma^3}$. Wenn $skew(x) > 0$ linkssteil (rechtsschief) bzw. $skew(x) < 0$ rechtssteil (linksschief).
Die Wölbung einer Verteilung ist ein Maß für ihrere Spitzheit: $kurt(x) = \frac{\mu_4}{\sigma^4}$. Wenn $kurt(x) < 3$ flach bzw. $kurt(x) > 3$ spitz. Falls x normalverteilt gitl: $kurt(x) = 3$.

3.1.3 Emprische Werte

Die vorliegenden Messwerte $x_1, \ldots, x_n$ stellen eine Stichprobe aus der Zufallsvariablen x zugrunde liegenden Verteilung dar. Überlichweise sind die wahren Parameter (μ, σ) dieser Verteilung nicht bekannt. Mann muss daher diese Parameter auf Basis der Stichprobe schätzen.
Der empirische Mittelwert: $\overline{x} = \frac{1}{n} * \sum_{i=1}^{n} x_i$ wobei $E[\overline{x}] = \mu$.
Die empirische Varianz ist: $s^2 = \frac{1}{n-1} * \sum_{i=1}^{n}(x_i - \overline{x})^2$ wobei $E[s^2] = \sigma^2$.

3.1.4 Lageparameter

Lageparameter treffen allgemeine Aussagen über die Position der Verteilung und stellen in gewisser Weise die Lage ihres Schwerpunkts dar. Verschiedene Lagemaße sind dabei unterschiedlich robust, zeigen sich also mehr oder weniger empfindlich gegenüber Ausreißerwerten. Beispiele: Mittelwert, Median, Modus.
Der Median ist der kleinste Wert x_m, bei dem die kumultative Verteiltungsfunktion $F_p(x) = \int_{-\inf}^{x} p(z)dz$ einen Wert von ≥ 0.5 liefert d.h. er teilt die Verteilungsfunktion in zwei (flächenmäßig) gleich große Teile.
Für eine geordnete Stichprobe $x_1 \leq \cdots \leq x_n$ ist der Median der Wert $x_{(n+1)/2}$ (falls n ungerade) bzw. der Wert $1/2 * (x_{n/2} + x_{n/2+1})$. Daraus folgt dass der Median als Lagemaß wesentlich unempfindlicher ist gegenüber Ausreißer als der Mittelwert.

Der Modus einer Verteilung ist der Wert mit der größten Wahrscheinlichkeit. Gibt es nur einen Modus nur einen Modalwert nennt man die Verteilung unimodal, sonst besitzt Verteilung mehre Modalwerte und nennt sie desshhalb bimodal.
Es gilt:

- links-steile, rechts-schiefe Verteilung: Modus $<$ Median $<$ Mittelwert
- rechts-steile, links-schiefe Verteilung: Modus $>$ Median $>$ Mittelwert

3.1.5 Streuungsmaße

Streuungsmaße beschreiben die Breite bzw. die Ausdehnung einer Verteilung und somit die Abweichung vom Schwerpunkt. Sie sind somit ein Maß für die Variabilität der Daten. Beispiel: Varianz, Quatile, Interquartilsabstand.

- Quartile: drei Quartile teilen die geordnete Datenmenge in vier gleich große Segmente, wobei das zweite Quartil gleichzeitig den Median darstellt
- Interquartilabstand bezeichnet den Abstand zwischen den 1. und den 3. Quartil und umfasst also die Hälfte der Daten und ist analog zum Median, robuster gegen Ausreißer als die Varianz-

3.1.6 Standardisierung

- Problem: Normalverteilung $N(\mu, \sigma^2)$ hat nicht zwangsläufig eine Fläche von 1 unter der Kurve
- Lösung: Transformation mit $z = \frac{x-\mu}{sigm}$ dadurch Normalverteilung $N(0,1)$ als Ergebnis
- Mahalanobis-Abstand: $r = \frac{|x-\mu|}{\sigma}$ (Z-Score) (Musst die Distanz zwischen x und μ in Einheiten der Standardabweichung)
- wenn $z = \frac{x-\mu}{\sigma}$ dann gilt:
  - $skew(x) = \frac{E[(x-\mu)^3]}{\sigma^3} = E[(\frac{x-\mu}{\sigma})^3] = E[z^3]$
  - $kurt(x) = \frac{E[(x-\mu)^4]}{\sigma^4} = E[(\frac{x-\mu}{\sigma})^4] = E[z^4]$
  - d.h. Schiefe und Wölbung von x sind das 3 bzw. 4 Moment der standardisierten Verteilung x
- für multivarianten Fall mit Σ die Kozarianzmatrix gilt: $z = \Sigma^{-1/2} * (x - \mu)$ d.h. aus $N(\mu, \Sigma)$ wird $N(0, I)$

3.1.7 Korrelation

- x,y Zufallsvariablen und μ_x, μ_y die Erwartungswerte und σ_x, σ_y die Standardabweichung
- Korreleationkoeffizient p_{xy} ist ein Maß für die lineare Abhängigkeit
- $p_{xy} = \frac{E[(x-\mu_x)*(y-\mu-y)]}{\sigma_x * \sigma_y}$
- wenn $|p_{xy}| = 1$ denn lineare Abhängigkeit zwischen x und y
- wenn x und y unabhängig $\rightarrow p_{xy} = 0$ anders rum nicht
- emprische Korreleation r_{xy} zweier Stichproben $x = x_i, y = y_i$ misst die Abhängigkeit zweier Stichproben: $r_{xy} = \frac{x' * y'}{||x'|| * ||y'||}$ mit $x'_i = x_i - \overline{x}, y'_i = y_i - \overline{y}$

3.1.8 Einsatz statistischer Merkmale

- problemunabhängig d.h. können immer eingesetzt werden da kein Vorwissen über die Problemstruktur

3.2 Merkmalstypen

Niveau	Operationen	Lageparameter	Beispiel
nominal	$\{=, \neq\}$	Modus	Geschlecht
ordinal	$+\{>, <\}$	+Median	Bundesligatabelle
intervall	$+\{-, +\}$	+Erwartungswert	Geburtsjahr
ratio	$+\{/, *\}$	+geom. Mittel	Wohnfläche

- Intervall und Ratio Saklen sind metrisch d.h. Abstandsbegriff ist sinnvoll

- Addition auf Intervallsaklen für Abstände sinnvoll
- Ratioskalen haben Nullpunkt
- Nominal- und Ordinalskalen sind immer diskret

3.3 Merkmale für Zeitreihen

3.3.1 Summarische Merkmale

- Gegeben ein Signal $x = x_1, \ldots, x_n$
- Zero Crossing Rate (im Ortsbereich): $zcr(x) = \frac{1}{n-1} \sum_{i=2}^{n} \mathfrak{I}\{x_i * x_{i-1} < 0\}$ mit $\mathfrak{I}\{A\} = 1$ falls A wahr
- Energie (im Orts- und im Frequenzbereich): $en(x) = \sum_{i=1}^{n} x_i^2$
- Entropie (im Orts- und im Frequenzbereich); $ent(x) = -\sum_{i=1}^{n} x_i^* * log(x_i^*)$ wobei $x_i^* = \frac{x_i}{\sum_{j=1}^{n} x_j}$
- Schwerpunkt (spectral centroid)(Frequenzbereich): $sc(x) = (\sum_{k=1}^{K} f_k * |x_k|^2)/(\sum_{k=1}^{k} |x_k|^2)$
- Bandbreite (bandwidth)(Frequenzbereich) $bw(x) = (\sum_{k=1}^{k} (f_k - sc)^2 * |x_k|^2)/(\sum_{k=1}^{k} |x_k|^2)$

3.3.2 Autokorrelation, Grundfrequenz

- Signal $x = x_1, \ldots, x_n$
- Autokorrelation: $R_x(\tau) = (x \star x)(\tau) = \sum_{i}^{n} x_i * x_{i-\tau}$
- τ = Verzögerungsparameter (Lag)
- üblich Autokorrelation: $ACF_x(\tau) = \frac{R_x(\tau)}{R_x(0)}$
- Verwendung: Bestimmung der Grundfrequenz $->$ das zweite Maximum der ACF liefert die $1/f_0$ Periodendauer
- d.h. Tau als Parameter $\rightarrow$ durchprobieren bis zweite Maximum gefunden, Begründung: $\tau = 0$ ist immer das erste Maxima
- Maximum des Fourierspektrums nicht geeignet da im niedriegen Frequenzbereich nur eine grobe Auflösung

3.3.3 Phasendifferenz

- zwei Sinus Signale: $x_i = sin(w * i), y_i = sin(w * i + \phi)$ d.h. gleiche Frequenz aber Phasenverschiebung
- $\phi = arccos(r_{xy})$

3.4 Merkmale für kinetische Systeme ????

4 Bayessche Entscheidungstheorie

4.1 Einfürhung

- Natur hat einen Zustand ω aus einer Wertemenge $\Omega = \{\omega_1, \omega_2, \ldots\}$
- Beispiel ω: Fisch ist Wolfsbarsch (ω_1) oder Lachs(ω_2)
- ω ist meistens unbekannt d.h. eine Zufallsvariable
- Wenn es gleich viele Lachse und Wolfsbarsche gibt, ist es gleichwahrscheinlich die eine oder andere Art zu tippen
- a priori Wahrscheinlichkeit = Vorwissen
- P("Barsch") + P("Lachs) = 1
- wenn wir nur die a priori Wahrscheinlichkeitne kennen ist die Wahl mit der größten Wahrscheinlichkeit immer die beste
- Typischerweise haben wir mehr Informationen d.h. zusätzlichen Wissen für unsere Entscheidung

- Beispiel: die Helligkeit (Merkmal $x \in \mathbb{R}$) hängt von der Fischart ω ab d.h. wir haben klassenabhängige Wahrscheinlichkeiten
- $p(x|\omega)$ ist die klassenabhängige Wahrscheinlichkeit des Merkmals x
- uns interessiert $P(\omega_i|x)$ also die Wahrscheinlichkeit dass die Welt den Zustand ω_i hat nachdem wir x beobachtet haben
- Verwendung: Satz von Bayes: $P(\omega_i|x) = \frac{p(x|\omega_i)*P(\omega_i)}{p(x)}$ mit $p(x) = \sum_{i=1}^{2} p(x|\omega_i)P(\omega_i)$ (für unseren Fall von zwei Fischen)
  - $P(\omega_i|x)$: Die Wahrscheinlichkeit des wahren Zustands, nachdem wir den Beweis haben (Posteriori)
  - $p(x|\omega_i$: Die Passfähigkeit des Beweises x zum möglichen wahren Zustand ω (Likelihood)
  - $P(\omega_i)$: Die Wahrscheinlichkeit des wahren Zustands bevor wir den Beweis gesehen haben
- Nun Entscheidung für ω mit der größten Posteriori $P(\omega_i|x)$

4.2 Bayes'sche Entscheidungstheorie

4.2.1 Definitionen

- Merkmalsvektor x als Element eines d-dimensionalen Euklidischen Merkmalsraums R^d d.h. $x \in R^d$
- $\Omega = \{\omega_i, \omega_2, \ldots, \omega_c\}$ eine endliche Menge von möglichen Zuständen (Klassen, Kategorien)
- $A = \{\alpha_1, \ldots, \alpha_a\}$ eine endliche Menge von möglichen Aktionen
- Eine Kostenfunktion $\Lambda(\alpha_i|\omega_j)$ die angiebt welche Kosten die Aktion α_i verursacht wenn der Zustand ω_j ist

4.2.2 Bayes'sche Entscheidungstheorie - formal

- $p(x|\omega_j)$ und $P(\omega_j)$ sind wie vorher die klassenbedingte Wahrscheinlichkeit von x gegeben ω_j bzw. die a-priori Wahrscheinlichkeit von ω_j
- Es gilt analog für die a-posteriori-Wahrscheinlicht: $p(\omega_j|x) = \frac{p(x|\omega_j)*P(\omega_j)}{p(x)}$ wobei $p(x) = \sum_{i=1}^{c} p(x|\omega_i)P(\omega_i)$
- gegeben ein Merkmalsvektor x. Nun soll eine Aktion α_i durchzuführen. Welche kosten entstehen dabei?
- Wenn Zustand ω_j ist sind die Kosten von α_i nach Def: $\Lambda(\alpha_i|\omega_j)$
- Zustand ω_j ist jedoch nicht gegeben, wir kennen nur $P(\omega_j|x)$. Können aber den Erwartungswert der Kosten d.h. das bedingte Risiko $R(\alpha_i|x)$ bestimmen: $R(\alpha_i|x) = E_{\omega|x}[\Lambda(\alpha_i|\omega)] = \sum_{j=1}^{c} \Lambda(\alpha_i|\omega_j)P(\omega_j|x)$
- Bayessche Entscheidungsregel besagt nun: wenn x gegeben, wähle Aktion α_i für die das bedingte Risiko $R(\alpha_i|x)$ minimal ist
- sei $\alpha(x)$ eine Entscheidungsregel welche für ein gegebenes x eine Aktion α_i auswählt
- dann ist das Gesamtrisiko unter dieser Entscheidungsregel: $R_\alpha = \int R(\alpha(x)|x)p(x)dx$
- R_α wird minimal wenn man für jeden Punkt x das Minimum wählt d.h. die Aktion α_i für die $R(\alpha_i|x)$ minimal ist
- das minimale unvermeidbare Risiko R^* heißt Bayes-Risiko

4.2.3 Anwendung

- ???

4.2.4 Klassifikation und Diskriminanten

- Klassifikatoren können durch Diskriminantenfunktionen repräsentiert erden
- Klassifikator für die Klassen $\omega_1, \ldots, \omega_c$ besteht aus einer Menge von Diskriminantenfunktionen $g_i(x), i \in \{1, \ldots, c\}$
- dieser Klassfikator weisst x einer Klasse ω_i gdw. $g_i(x) > g_j(x), \forall j \neq i$
- Bayes-Klassifikator für den allgemeinen Fall mit Risiko kann wie folgt definiert werden: $g_i(x) = -R(\alpha_i|x)$
- für die Klassifikation mit minimalen Fehler vereinfacht sich dies zu: $g_i(x) = P(\omega_i|x) \propto p(x|\omega_i)P(\omega_i)$
- für eine gegebene Menge von Diskriminanten $g_i(x)$ liefert eine Transformation mit einer monoton wachsenden Funktion $f()$ eine äquivalente Menge von Diskriminanten $f(g_i(x))$
- dadurch alternative Formulierungen:
  - $g_i(x) = P(\omega_i|x) = \frac{p(x|\omega_i) * P(\omega_i)}{p(x)}$
  - $g_i(x) = p(x|\omega_i) * P(\omega_i)$
  - $g_i(x) = ln(p(x|\omega_i)) * ln(P(\omega_i))$
- Ein Klassifikator (Entscheidungsregel) für c Klassen zerlegt den Merkmalsraum $\mathbb{R}^d$ in maximal c Entscheidungsregionen $R_i, i \in \{1, \ldots, c\}$
- Falls für einen Merkmalsvektor x gilt $g_i(x) > g_j(x), \forall j \neq i$ dann liegt x in der Entscheidungsregion R_i und x wird als ω_i klassifiziert
- verschiedene Entscheidungsregionen werden durch Entscheidungsgrenzen voneinander getrennt
- Entscheidungsgrenzne sind die Bereiche in denen die beiden größten Diskriminanten denselben Wert annehmen

4.2.5 Normalverteilung

- Formel (univariant): $p(x) = N(x; \mu, \sigma^2) = \frac{1}{\sqrt{2*\pi}\sigma} \exp[-\frac{1}{2}(\frac{x-\mu}{\sigma})^2]$
- Mittelwert (erster Moment): μ, Varianz (zweite zentrale Moment): σ^2
- Der zentrale Grenzwertsatz: Die Summe von n unabhängigen Zufallsvariablen kovergiert gegen Normalverteilung mit $n \to \infty$. Da viele natürliche Vorgänge einer großen Zahl unabhängiger Störfaktoren unterliegen, ist die Normalverteilungsdannahme sinnvoll.
- hat hohe Entropie d.h. die Verteilung mit der größten Unsicherheit über die Werte einer Zufallsstichprobe
- Formel (multivariant): $N(x; \mu, \Sigma) = \frac{1}{(2*\pi)^{d/2}|\Sigma|^{1/2}} * \exp[-\frac{1}{2} * (x-\mu)^t \Sigma^{-1}(x-\mu)]$
  - $x = (x_1, \ldots, x_d)^t$ ein d-dimensionaler Spaltenvektor
  - $\mu = (\mu_1, \ldots, \mu_d)^t$ ein d-dimensnaler Mittelwert-Vektor
  - Σ ist eine $d \times d$ Kovarianzmatrix, $|\Sigma|$ die Diskriminante, Σ^{-1} Inverse Matrix
  - $\mu = E[x] = \int x * p(x))dx$ (können Komponentenweise bestimmt werden $\mu_i = E[x_i]$)
  - $\Sigma = E[(x-\mu)(x-\mu)^t]) = \int (x-\mu)(x-\mu)^t p(x)dx$ ($\sigma_{ij} = E[(x_i - \mu_i)(x_j - \mu_j)]$)
- für Kovarianzmatrix Σ gilt:
  - σ_{ii} sind die Varianzen der entsprechenden x_i also σ_i^2
  - σ_{ij} sind Kovarianzen von x_i und x_j, Wenn x_i und x_j unabhängig dann gilt $\sigma_{ij} = 0$
  - falls alle nicht-diagonalelemente Null sind, ist die multivariante Verteilung einfach das Produkt der d univarianten Verteilungen der Komponenten von x
  - die Kovarianzmatrix ist symmetrisch und positiv (semi-) definit (M positiv definit: $x^t Mx > 0 \forall x \neq 0$)
- Stichproben aus multivarianter Gaußverteilungen bilden typischerweise eine Häufung deren Mittelpunk von μ und deren Form von Σ definiert wird
- Orte gleicher Wahrscheinlichkeitsdichte liegen auf Hyperellipsoiden für die $(x-\mu)^t \Sigma^{-1}(x-\mu)$ konstant ist

- die Hauptachsen der Hyperellipsoide sind die Eigenvektoren von Σ, die Eigenwerte geben die Länge dieser Achsen an
- $r^2 = (x-\mu)^t\Sigma^{-1}(x-\mu)$ ist die quadrierte Mahalanobis-Distanz; Konturen konstanter Dichte sind Hpyerellipsoide konstanter Mahalanobis-Distanz zu μ

4.3 Diskriminanten für die Normalverteilung

- vorher gezeigt dass Diskriminante $g_i(x) = ln(p(x|\omega_i)) * ln(P(\omega_i))$ Klassifikation mit minimaler Fehlerrate liefert
- wenn $p(x|\omega_i) = N(\mu_i, \Sigma_i)$ einsetzen in $g_i(x)$: $g_i(x) = -\frac{1}{2}(x-\mu_i)\Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}ln(2\pi) - \frac{1}{2}ln(|\Sigma_i|) + ln(P(\omega_i)) \rightarrow$ Entfernung von $-\frac{d}{2}ln(2\pi)$ da konstant und nicht von der Klasse abhängt
- Also erhalten wir: $g_i(x) : g_i(x) = -\frac{1}{2}(x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i) - \frac{1}{2}ln(|\Sigma_i|) + ln(P(\omega_i))$
- Entscheidungsgrenzen:
  - $\Sigma_i = \sigma^2 I$ d.h. Alle Merkmale unabhängig, gleiche Varianzen, auf der Hauptdiagonalen stehen die gleichen Werte; Die Klassen bilden kugelförmige Häufungen gleicher Größe um ihre jeweiligen Mittelpunkte.
    - es gilt: $\Sigma_i^{-1} = (1/\sigma^2)I$ und $|\Sigma_i| = \sigma^{2d}$
    - einsetzen in Formel: $g_i(x) = -\frac{1}{2\sigma^2}(x-\mu_i)^t(x-\mu_i) - \frac{1}{2}ln(\sigma^{2d}) + ln(P(\omega_i))$
    - Konstanten entfernen: $g_i(x) = -\frac{1}{2\sigma^2}(x-\mu_i)^t(x-\mu_i) + ln(P(\omega_i))$
    - Euklidische Distanz: $(x-\mu_i)^t(x-\mu_i) = ||x-\mu_i||^2$
    - Also: $g_i(x) = -\frac{||x-\mu_i||^2}{2\sigma^2} + ln(P(\omega_i))$
    - Expandieren: $g_i(x) = -\frac{1}{2\sigma^2}(x^tx - 2\mu_i^tx + \mu_i^t\mu) + ln(P(\omega_i))$
    - Also erhalten wir eine lineare Diskriminante: $g_i(x) = w_i^t * x + w_{i0}$ (lineare Maschine, w_{i0} ist Schwellwert)
    - d.h. die Entscheidungsgrenzen zwischen den Regionen werden durch die lineare Gleichung: $g_i(x) = g_j(x)$ definiert (Hyperebenen)
    - die Hyperebene ist orthogonal zum Vektor w, der Verbindungsline zwischen den beiden Mittelwertvektoren
    - Line wird in x_0 geschnitten falls die a-priori-Wahrscheinlichkeiten gleich sind, liegt x_0 zwischen μ_i und μ_j
  - $\Sigma_i = \Sigma$ d.h. alle Klassen haben gleiche Kovarianzmatrix d.h.Die Klassen bilden hyperellipsoide Häufungen gleicher Größe, Form und Orientierungen um ihre jeweiligen Mittelpunkte.
    - wieder einsetzen in g_i: $g_i(x) = -\frac{1}{2}(x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i) - \frac{1}{2}ln(|\Sigma_i|) + ln(P(\omega_i))$
    - Konstanten entfernen: $g_i(x) = -\frac{1}{2}(x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i) + ln(P(\omega_i))$
    - Ausmultiplizieren und Symmetrie von Σ liefert wieder lineare Maschine
    - nicht mehr orthogonal zur Verbindungslinie der Mittelwerte, bei gleicher a-priori-Wahrscheinlicht x_0 immer noch auf halben Wege zwischen μ_i und μ_j
  - $\Sigma_i = beliebig$
    - einsetzen und ausmultiplizieren ergibt Quadratische-Diskriminante
    - beliebige Normalverteilungen führen zu Entscheidungsgrenzen welche allgemeine Hyperquadriken sind
    - Bereits kleine Anzahl von Klassen können die Entscheidungsgrenzen komplexe Formen annehmen lassen

4.4 Empirischer Fall

- im Anwendungsfall eher unrealisitisch das man alle Parameter kennt d.h. muss diese auf Basis von Traningsdaten schätzen
- wenn man Normalverteilung annehmen möchte:
  - $\hat{P}(\omega_i) = \frac{n_i}{n}$
  - $\hat{\mu}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij}$
  - $\hat{\Sigma} = \frac{1}{n-c}\sum_{i=1}^{c}\sum_{j=1}^{n_i}(x_{ij}-\hat{\mu}_i)(x_{ij}-\hat{\mu}_i)^t$

  - mit n_i Anzahl der Traningsdatensätze für die Klassei, n Anzahl aller Traningsdatensätze und x_{ij} Traningsdatensatz Nummer j für Klasse i
  - dies nennt man Lineare Diskriminazanalyse
- für Fall 3:
  - $\hat{P}(\omega_i) = \frac{n_i}{n}$
  - $\hat{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$
  - $\hat{\Sigma_i} = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (x_{ij} - \hat{\mu}_i)(x_{ij} - \hat{\mu}_i)^t$
  - dies nennt man Quadratische Diskriminazanalyse
- Bias/Variance Tradeoff = ???

4.5 Diskrete Merkmale

- Integrale durch Summen ersetzen, sonst alles wie bisher

5 Parameterschätzung

5.1 Einführung

- Ziel: Konstruieren eines Klassifikators
- Benötigen: a-priori-Verteilung $P(\omega_i)$ und klassenbedingten Verteilungen $p(x|\omega_i)$
- Problem: Verteilungen sind unbekannt
- haben Anzahl an Stichproben $D = (x_1, \ldots, x_n)^t$ mit dazugehörigen Klassen $t = (\omega_{i1}, \ldots, \omega_{\omega_{in}})^t$ mit D ist eine $n \times d$ Matrix, d ist die Anzahl der Dimensionen des Merkmalsraums. Jede Zeile von D ist eine Beobachtung x_i
- n_i Anazahl der Trainingsdaten der Klasse ω_i und es gilt $n = \sum_{i=1}^{c} n_i$
- $P(\omega_i)$ können wir schwätzen durch z.b. Zahlen der Häufigkeiten: $\hat{P}(\omega_i) = \frac{n_i}{n}$
- $p(x|\omega_i)$ ist das Problem wir müssen eine Funktion $f(x)$ schätzen d.h. den Funktionswert an unendlich vielen Stellen $x \in R^d$, Problem: Raumvolumen wächst exponentiell $\rightarrow$ bei einer festen Anzahl von Trainingsdaten sind x_i immer weiter im Raum versteht d.h. die leeren Bereiche welche $f(x)$ raten muss werden immer größer
- Lösung: Annahme das $f(x)$ keine beliebige Form hat sondern aus einer Funktionsfamilie $f(x|\theta)$ stammt, aus der die gescuhte Funktion durch einen Parametervektor θ mit niedriger Dimensionaltiääöt bestemmt werden wird
- Beispiel: $f(x|\theta) = N(x|\mu, \Sigma)$ mit $\theta = (\mu, \Sigma)$ d.h. wir müssen nicht mehr unendlich viele Werte aus R schätzen sondern nur noch wenige Parameter
- Aufgabe: Auf Basis der Annahme der parametrischen Form $p(x|\theta_i)$ der Verteilungsfunktion $p(x|\omega_i)$ bestimme den Parametervektor θ_i mit Hilfe der Traningsdaten D^i, t^i
- Lösungen:
  - Maximum Likelihood (Punktschätzung)
  - Bayes'sch (Bestimmung der a-posteriori Verteilung von θ)
- Also: Wir haben Trainingsdaten $D = x_1, \ldots, x_n$ und wir nehmen an, dass x_k aus einer parametrischen Verteilung $p(x_k|\theta)$ stammen. Gesucht ist nun eine Schätzung für die wahren Parameter die am besten mit den Traningsdaten vereinaber ist

5.2 Maximum Likelihood-Schätzer

- Annahme: der Parametervektor θ hat einen wohldefinierten festen Wert der uns unbekannt ist
- Maximum-Likelihood-Schätzer: $\hat{\theta} = argmax_{\theta \in \Theta} p(D|\theta)$ also derjenige Wert $\hat{\theta}$ der die Likelihood der Traningsdaten maximiert, Anwendung Satz von Bayes: $p(\theta|D) = \frac{p(D|\theta) * p(\theta)}{p(D)}$
- Frage: Wo kommt der her? Man erwartet eigentlich $p(\theta|D)$
- Antwort: eigentlich interessiert uns $p(\theta|D)$ aber uns interessiert nicht die gesamte Verteilung sondern nur der Wert $argmax_\theta p(\theta|D)$
- Annahme: Alle Werte von θ sind gleich wahrscheinlich $p(\theta) = c$
- also $argmax_\theta p(\theta|D) = argmax_\theta \frac{p(D|\theta) * p(\theta)}{p(D)} = argmax_\theta \frac{p(D|\theta) * c}{c'} = argmax_\theta p(D|\theta) = \hat{\theta}_{ML}$
- Gegeben:
  - Menge von Daten: $D = x_1, x_2, \ldots, x_n$
  - parametrische Verteilungsfunktion: $p(x|\theta)$
  - p-dimensionalen Parametervektor: $\theta = (\theta_1, \ldots, \theta_p)^t$
- Gesucht: $\hat{\theta}$ der $p(D|\theta)$ maximiert
- Annahme: x_k sind unabhängige Stichproben aus $p(x|\theta)$ d.h. es gilt: $p(D|\theta) = \prod_{k=1}^{n} p(x_k|\theta)$
- Vorgehen:
  1. Bestimme die Log-Likelihood-Funktion: $l(\theta) = log \prod_{k=1}^{n} p(x_k|\theta) = \sum_{k=1}^{n} log(p(x_k|\theta))$
  2. Wende Gradientenoperator $\nabla_\theta = (\frac{\partial}{\partial \theta_1}, \ldots, \frac{\partial}{\partial \theta_p})^t$ auf $l(\theta)$ an d.h. bestimme: $\nabla_\theta l = \sum_{k=1}^{n} \nabla_\theta ln(p(x_k|\theta))$
  3. Löse die Gleichung: $\nabla_\theta l = 0$
  4. Lösung der Gleicung ist $\hat{\theta}$
- Bias: Der Erwartungswert der Schätzwerte entspricht nicht den realen Parametern. ML-Schätzer sind jedoch asymptotisch erwartungstreu
- Maximum-Likelihood ist sehr einfach anzuwenden
- Wenn die a-posteriori Verteilung unimodal ist, dann liefert ML Ergebnisse die ähnlich gut sind wie Bayessche Schätzung, Ml kovergiert mit $n \to \infty$ gegen die wahren Parameter

5.3 Bayessche Parameterschätzung

- Aufgabe: Bestimme $p(x|D)$ anhand einer Stichprobe D die aus einer parametrischen Verteilung $p(x|\theta)$ gezogen wurde. Dabei habe $p(x|\theta)$ eine bekannte funktionale Form, nur θ ist nicht festgelegt
- Wir stellen uns θ als Zufallsvariable vor die eine a-priori-Verteilung (vorher θ unbekannt und fest jetzt unbekannt und aus einer Verteilung) $p(\theta)$ hat aus der wir eine a-posteriori-Verteilung $p(\theta|D)$ berechnen können
- es ergibt sich folgende a-posteriori Verteilung für die Daten: $p(x|D) = \int p(x|\theta) p(\theta|D) \partial\theta$
- Vergleich zu ML-Ansatz: $p(x|D) = p(x|\hat{\theta}_{ML})$ mit $\hat{\theta}_{ML} = argmax_\theta p(D|\theta)$
- Ziel: Bestimmung der Parameterverteilung $p(\theta|D)$
- Bayessche Satz: $p(\theta|D) = \frac{p(D|\theta) * p(\theta)}{\int p(D|\theta) p(\theta) \partial\theta}$ mit $p(D|\theta)$ ist Likelihood
- Annahme: Unabhängigkeitsannahme für D d.h. die Merkmale hängen nicht voneinander ab es gilt also: $p(D|\theta) = \prod_{k=1}^{n} p(x_k|\theta)$

6 Dimensionsreduktion

6.1 Einführung

- wenn Merkmale unabhängig sind dann lassen sich theoretisch Klassifizierer mit sehr hohrer Präzision realisieren
- für zwei-Klassen-Problem $p(x|\omega_i) = N(\mu_i, \Sigma)$ der Bayes-Fehler: $P(e) = \frac{1}{\sqrt{2*\pi}} \int_{r/2}^{\infty} e^{-u^2/2} du$ mit r^2 ist die quadrierte Mahalanobis-Distanz
- mit wachsendem r sinkt der Fehler, bei unabhängigen Merkmale ist $\Sigma = diag(\sigma_1^2, \ldots, \sigma_d^2)$
- man erhält: $r^2 = \sum_{j=1}^{d} \frac{(\mu_1^{(j)} - \mu_2^{(j)})^2}{\sigma_j^2}$ d.h. je größer d (Anzahl der Dimensionen / Merkmale) desto besser ist die Klassifikation (theoretisch, praktisch eher nicht so)
- im Bayesschen Framework, wenn wir die Formen der Verteilungen kennen können wir die Anzahl der Features beliebig erhöhren und der Ansatz sorgt von selbst dass diese für irrelevante Features entsprechend gering gewichtet werden
- Probleme:
  - in hohen Dimensionen sind bereit geringe Unterschiede zwischen wahrer und angenommener Verteilung sehr groß
  - in hohen Dimensionen ist die Anzahl der Traningsdaten nicht mehr ausreichend um die erforderliche Anzahl der Parametern ausreichend zuverlässig zu schätzen
- Lösung:
  - Hauptkomponentenanalyse (PCA): Projektion in einem Raum mit reduzierter Dimensionalität die die Daten am besten repräsentiert
  - Fisher-Diskriminante: Projektion in einem Raum mit reduzierter Dimensionalität in der die Daten unterschiedlicher Klassen am besten getrennt sind

6.2 Hauptkomponentenanalyse

- Aufgabe: reduziere die Anzahl der Dimensionen auf $q < d$
- Ansatz: finde einen q-dimensionalen Teilvektorraum E^q mit Basis: $E = (e_1, \ldots, e_q)^t$ so dass die Projektion der x_k in den Raum E^q möglichst gut repräsentiert sind d.h einen minimalen quadratischen Fehler verursacht
- Kriterium zu minimieren: $J(E) = \sum_{k=1}^{n} (E^t(Ex_k)) - x_k)^2$
- E ist eine $q \times d$ Matrix die Vektoren von einem d-dimensionalen Vektorraum E^d in einen q-dimensionalen Untervektorraum E^q abbildet
- $x'_k = Ex_k$ die Projekt einen d-dimensionalen Vektors x_k in E^q
- $x''_k = E^t x'_k$ ist die Rücktransformation des reduzierten Vektors in den ursprünglichen Vektorraum, in dem der Fehler (Distanz zwischen x''_k und x_k) gemessen
- q kann nicht größer als n sein da n Datenvektoren spannen höchstens n-dimensionalen Vektorraum auf
- Vorgehen:
  1. Berechne empirische Kovarianzmatrix $\hat{\Sigma}$
  2. Berechne die Eigenwerte λ_i und Eigenvektoren e_i von $\hat{\Sigma}$
  3. wähle $q < d$ Eigenvektoren zu den größten Eigenwerten als dimensionsreduzierten Merkmalsraum mit Transformationsmatrix mit $E = (e_1, \ldots, e_q)^t$
  4. Die neuen q-dimensionalen Merkmalsvektoren x' ergeben sich jetzt durch die Transformation $x' = Ex$

6.3 Diskriminantenanalyse

- Aufgabe: Suche eine Transformation die das Verhältnis der Varianz zwischen den Klassen zur Varianz innerhalb der Klasse maximiert
- Die Transformation erzeugt Räume mit maximal $c-1$ Dimensionen da es nur c Klassen gibt, können c Klassenmittelpunkte einen Raum mit maximal $c-1$ Dimensionen aufspannen
- wir definieren:
  - Streuung innerhalb der Klassen: $S_W = \sum_{i=1}^{c} \sum_{k=1}^{n_i} (x_{ik} - \mu_i)(x_{ik} - \mu_i)^t$
  - Streuung zwischen den Klassen i: $S_B = \sum_{i=1}^{c} n_i(\mu_i - \mu)(\mu_i - \mu)^t$
  - mit x_{ik} ist Datensatz k zu Klasse i, n_i ist die Anzahl der Datensätze für Klasse i, μ ist der Mittelwert überalle Klassen
- Die Determinante einer Streuungsmatrix ist ein Maß für das Raumvolumen das die zugehörigen Daten benaspruchen
- Streungsverhältnis: $\frac{|S_B|}{|S_W|}$
- Gesucht: Transformation W in einen $q \leq c-1$ dimensionalen Raum E^q so dass das Streungsverhältnis in E^q maximal wird, W ist eine $q \times d$ Matrix
- mit W gegeben: $\hat{S}_B = \sum_{i=1}^{c} n_i(W\mu_i - W - \mu)(W\mu_i - W - \mu)^t = \cdots = W * S_B * W^t$ analog: $\hat{S}_W = W * S_W * W^t$
- Aufgabe: minimiere $J(W) = \frac{|W * S_B * W^t|}{|W * S_W * W^t|}$
- Lösung: Eigenwertproblem (ausgelasssen) lösen

6.4 Nichtlineare Diskiminaten

- Gegeben: Feature Space welcher nicht linear trennbar ist
- Aufgabe: Finde einen Vektor $\phi = (\phi_1, \ldots, \phi_f)$ von f (nichtlinearen) Transformationen $\phi_i(x)$ und betrachte lineare Diskriminante in f-dimensionalen Raum E^f

7 Nicht-parametrische Methoden

7.1 Einführung

- Bisher:
  - vollständigen Wissen über die Verteilung: Bayessche Entscheidungsregel
  - oder: wir kennen zumindestens die funktionale Form der Verteilung und können die Parameter der Verteilung aus den Trainingsdaten schätzen
- Nun: Verteilungsfunktionen sind nicht bekannt
- Problemstellung:
  - Gesucht: unbekannte Verteilung $p(x)$
  - Gegeben: Menge von Stichproben $x_i \sim p(x)$ aus dieser Verteilung
  - Idee: Dichte der x_i in der Nähe von x verwenden um damit auf die Wahrscheinlichkeit $p(x)$ zu schließen $\rightarrow$ je dichter die x_i in einer Region liegen desto wahrscheinlicher dürfte es ja sein, Werte aus dieser Region zu erhalten vorrausgesetzt $p(x)$ ist nicht bösartig zerklüftet
- Gegeben eine Region $R \subset R^n$ im Merkmalsraum und eine Verteilungsfunktion $p(x)$
- Wahrscheinlichkeit P dass ein $x \in R^n$ in R liegt dann: $P := Pr(x \in R) = \int_R p(x')dx'$
- nun seien n Stichproben $x_1, \ldots, x_n$ mit $x_i \sim p(x)$ gegeben
- Wahrscheinlichkeit dass genau k mit $k < n$ dieser Stichproben in R liegen: $P_k = \binom{n}{k} P^k (1-P)^{n-k}$ (binomialverteilt)
- Erwartungswert: $E[k] = \sum_k k * k_k = n * P$

- mit $E[k]$ lässt sich nun P umgekehrt schätzen: $P \approx \frac{k}{n}$
- Mit der Annahme dass $p(x)$ ist kontinuierlich und einer Umgebung von x nicht allzu stark schwanklt: Wenn V das Volumen der Region R ist dann Gilt: $k/n \approx P = \int_R p(x')dx' \approx \int_R p(x)dx' = p(x)\int_R 1dx' = p(x) = V$
- damit ergibt sich: $p(x) \approx \frac{k/n}{V}$
- Bedeutung: wir können $p(x)$ Abschätzen mit Hilfe einer Stichprobe der Größe n, wenn wir die Anzahl k der Samples in einer Region R um x in Relation zur Gestamtzahl zur Größe der Region V setzen
- zu zeigen: der Schätzwert kann die gesuchte Wahrscheinlichkeit $p(x)$ beliebig genau approximieren wenn man n nur groß genug wählt d.h.: $p(x) = \lim_{n\to\infty} \frac{k_n/n}{V_n}$ wenn man k_n und V_n geeignet definiert
- damit $p(x) = \lim_{n\to\infty} \frac{k_n/n}{V_n}$ gilt müssen folgende Bedingungen gelten:
  - $\lim_{n\to\infty} V_n = 0$ d.h. der Fehler die Durchschnittsbilding im Raumvolumen verschwindet
  - $\lim_{n\to\infty} = \infty$ d.h. die möglicherweise irrationale Zahl P kann durch die Anzahl der Samples in R beliebig genau approximiert werden
  - $\lim_{n\to\infty} = 0$ mit schrumpfenden Raumvolumen sinkt auch die relative Anzahl von Treffern im Volumen
- Zwei Ansätze:
  - Parzen-Window-Ansatz (Kern-Dichte-Schätzer): Volumen V_n in Abhängigkeit von n festlegen und dann zeigen dass Konvergenz mit $n \to \infty$ erreicht wird
  - k-nearest-Neighbour: Anzahl der Treffen in Abhängigkeit von n festlegen und hierfür Konvergenz zeigen

7.2 Parzen Windows

- Idee: um den Punkt x, dessen Wk wir bestimmen wollen, eine Testregion mit vorgegebene Volumen V_n legen und dann zählen, welcher Anteil k_n der Sichtprobe in dieses Volumen fällt
- das Verhältnis k_n/n ist dann eine Schätzung für $p(x)$: Schätzung mit n Samples: $p_n(x)$
- Annahme: Testregion R_n für n Stichproben ist Hyper-Cube mit Kantenlänge: h_n dann gilt: $V_n = h_n^d$ d.h. Das Volumen ist die d-te Potenz der Kantenlänge (d = Dimonensionenzahl)
- Vorgehen:
  - Fensterfunktion: $\phi(u) = \begin{cases} 1 & |u_j| \leq 1/2; i = 1\ldots,d \\ 0 & sonst \end{cases}$ (stanzt einem im Ursprung zentrierten Hyperwürfel der Kantenlänge 1 aus)
  - für Hyperwürfel mit kantenlänge h_n zentriert um x gilt: $\phi(\frac{x-x_i}{h_n}) = \begin{cases} 1 & \text{falls} x_i \text{im Hyperwuerfel um x liegt} \\ 0 & sonst \end{cases}$
  - x_i die Punkte welche sich im Hypercube befinden
  - Anzahl der Stichproben im Fenster: $k_n = \sum_{i=1}^n \phi(\frac{x-x_i}{h_n})$
  - es folgt aus $p_n(x) = \frac{k_n/n}{V_n}$ und k_n: $p_n(x) = 1/n * \sum_{i=1}^n \frac{1}{V_n}\phi(\frac{x-x_i}{h_n})$
  - $p_n(x)$ ist der Durchschnitt von n Funktionen die von x und den x_i abhängen
- Erweiterungen:
  - $\phi(\frac{x-x_i}{h_n})$ kann unterschiedlich interpretiert werden: 1) Funktion von x_i die ihren Wert in Anhängigkeit davon liefert ob x_i im Fenster um x liegt (k_n = Anzahl der x_i im Fenster um x) ODER 2) Funktion von x die ihren Wert in Abhängigkeit davon liefert ob x im Fenster um das jeweilige x_i liegt (k_n = Anzahl der Fenster um die x_i in denen x liegt)
  - Andere Fenster
    - Hyperwürfel mit 1/0 Zählung erzeugt Diskontinuität d.h scharfe Kanten
    - besser glatte Fenster: Gewicht des Samples sinkt kontinuierlich mit der Entfernung
    - Betrachtung von ϕ als Kern (Kernel) K: für K gilt: $\forall u : K(u) \geq 0; \int K(u)du = 1; \forall u : K(u) = K(-u)$

  - ∗ Epanechnikov-Kern minimiert den erwarteten integrierten quadratischen Fehler zwischen geschätzter und wahrer Verteilung: $K_{ep}(u) = \begin{cases} \frac{d+2}{2*c_d}(1 - ||u||^2) & \text{falls} ||u||^2 < 1 \\ 0 & \text{sonst} \end{cases}$

- Betrachtung als Faltung:
  - $f_n(x') = \sum_{i=1}^{n} \delta(x' = x)$ d.h. eine Stichprobe wird durch δ-Funktionen repräsentiert
  - für einen Kern definieren wir: $K_h(x, x') = \frac{1}{n*h} K(\frac{x-x'}{h})$
  - wir erhalten: $p_n(x) = K_h(x, x') \star f_n(x') = K_h(x, x') \star \sum_{i=1}^{n} \delta(x' = x) = \cdots = \frac{1}{n*h} \sum_{i=1}^{n} K(\frac{x-x_i}{h})$
  - d.h. Kern-Dichte-Schätzung ist die Filterung des Signals an einem Filter mit der Impulsantwort K_h

- Flexibilität:
  - Parzen-Window Methode kann bei jeder Art von Verteilung angewendet werden
  - Anzahl von Dimensionen Grenzen gesetzt: die Anzahl der erforderlichen Samples wächst exponentiell mit d d.h. $d > 3$ meist nicht mehr sinnvoll

- Klassifikation:
  - Bestimme $p(x|w_i)$ mit Hilfe der Parzen-Window-Methode
  - Ordne einen neuen Punkt diejenigen Klasse zu, die die maximale a-posteriori-Wahrscheinlichkeit hat
  - Seien $x_{ik}, k = 1, \ldots, n_i$ die Samples der Klasse i
  - klassenbedingte Wahrscheinlichkeit $\hat{p}(x|w_i) = \frac{1}{n_i * h_i} \sum_{k=1}^{n_i} K(\frac{x-x_{ik}}{h_i})$
  - a-priori-Wahrscheinlichkeiten: $\hat{p}(w_i) = \frac{n_i}{n}$
  - Klassifikation durch: $argmax_{w_i} \hat{p}(x|w_i)\hat{p}(w_i)$
  - Kombination mit Naiven-Bayes-Klassifikator: Annahme dass die einzelnen Merkmale voneinander unabhängig sind: $p(x|w_j) = \prod_{i=1}^{d} p(x_i|w_j)$
  - hierbei sind $p(x_i|w_j)$ eindimensionale Verteilungen die sich dann auch sinnvoll mit Hilfe des Parzen-Window-Ansatzes approximieren lassen

7.3 k Nearest Neighbours

- Dichteschätzung:
  - Ansatz: $p(x) = \frac{k}{n*V}$
  - Kern-Dichte-Schätzer: Volumenfest bestimmt durch h
  - k Nearest Neighbours: k wird festgelegt und Volumen wird so lange vergrößert bis k Samples gefunden wurden
  - man kann zeigen dass wenn man k_n langsamer wachsen lässt als n, die erzeugten Volumen V_n gegen 0 konvergieren: $p_n(x) = \frac{k_n}{n*V_n} \approx p(x)$ mit $\lim_{n\to\infty} p_n(x) = x$

- $p_n(x) = \frac{k_n}{n*V_n} \approx p(x)$ in der Praxis
  - sei $n = 1$. Es gibt also genau 1 Sample x_1 und wir haben $k_n = \sqrt{n} = 1$
  - gegeben sei ein Testpunkt x
  - Das kleinste Volumen k_1, des kleinstenm Hyperwürfels mit Zentrum x, in der wir gerade noch x_1 finden? $2|x - x_1|$
  - daraus ergibt sich eine sehr schlechte Schätzung: $p_1(x) = \frac{1}{2|x-x_1|}$

- Verhalten für k-NN Dichteschätzung: für ein n kann die k-NN Dichteschätzung ziemlich stachelig sein

- Klassifikation
  - n Samples x_i aus c Klassen
  - suchen k nächsten Samples in der Umgebung eines Punktes x
  - k_j Anzahl der Samples aus k welche zur Klasse w_j gehören ($\sum_{j=1}^{c} k_j = k$)
  - mit $\hat{p}(x, w_j) = \frac{k_j}{n/V}$ erhalten wir: $\hat{P}(w_j|x) = \frac{p(x,w_j)}{p(x)} = \frac{k_j}{k}$ alsop der Stimmanteil für w_j relativ zur Gesamtanzahl der Stimmen k
  - Klassifikation nach: $argmax_{w_j} \hat{P}(w_j|x)$
  - Um einen Punklt x zu klassifizieren: Bestimme die k nächstliegenden Trainingspunkte x_i und weise x diejenige Klasse zu, die unter diesen k Punkten am häufigsten Auftritt

8 Support Vector Machines

8.1 Einführung

- Geg: Zwei Klassen in einem Merkmalsraum mit nichtlinearer Trennbarkeit
- Lösung: Transformation in einen höherdimensionalen Merkmalsraum
- Allg. die Transformation ist unbekannt welche eine lineare Trennbarkeit erreicht
- Schrotflinten-Verfahren: Wähle eine geeignete Menge von Basisfunktion und probiere rum :D
- Wie findet man einen geeigneten Satz von Transformation $\phi : R^d \rightarrow R^f$
- Wie findet man eine für die transformierte Punktemenge eine geeignete lineare Entscheidungsgrenzen im R^f?
- Wie sorgt man dafür dass die Klassifikationkomplexität...?

8.2 Entscheidungsgrenzen

- gegeben eine Transformation $\phi : R^d \rightarrow R^f$ und Trainingsdaten und Klassenzugehörigkeit
- gesucht lineares Modell: $y(x) = w^t\phi(x) + b$ so dass die Ebene $y(x) = 0$ die beiden Klassen trennt
- Gesucht wird als für gegebene x_i eine möglichst gute Trenngrenze
- Wir suchen diejenige Ebene $y(x) = 0$ die den Abstand zu den jeweils am nächsten liegenden Punkten beider Klassen maximiert
- Diejenige Punkte die der Ebene $y(x) = 0$ am nächsten liegen stützen die Ebene ab $\rightarrow$ Stützvektoren
- Mathematisch: Gesucht $y(x) = w^t\phi(x) + b = 0$ die den minimalen Abstand der Punkte zu ihr maximiert also de Lösung für: $argmax_{w,b}(\frac{1}{||w||}min_i(t_i(w^t\phi(x_i) + b)))$
- Problem: schwieriges Optimierungsproblem
- Lsg: Umwandlung in ein einfacheres
- Beobachtung: Wenn wir von gegebenen w,b zu skalierten Werten k*w und k*b übergehen ändert sich das Ergebnis nicht d.h $\frac{t_i(w^t\phi(x_i)+b)}{||w||} = \frac{t_i((k*w)^t\phi(x_i)+k*b)}{||k*w||}$
- für gegebenes w,b finden wir also immer ein k so dass ür die skalierte Lösung $\frac{t_i((k*w)^t\phi(x_i)+k*b)}{||k*w||} = \frac{t_i(w'^t\phi(x_i)+b')}{||w'||} = \frac{1}{||w'||}$
- Ansatz: suche einfach k so dass gilt:$t_i(w'^t\phi(x_i) + b') = t_i((k * w)^t\phi(x_i) + k * b) = 1$ gilt: wir benennen w' und b' um in w und b
- für die der optimalen Entscheidungsgrenze am nächsten liegen gilt: $t_i(w^t * \phi(x_i) + b) = 1$
- Für alle Punkte fordern wir also: $t_i(w^t*\phi(x_i)+b) \geq 1$ (kanonische Repräsentation der Entscheidungsgrenze)
- Punkte mit Wert = 1 sind aktive Punkte
- Vereinfachtes Optimierungsproblem: $argmax_{w,b}(\frac{1}{||w||}min_i(t_i(w^t\phi(x_i) + b))) = argmax_{w,b}\frac{1}{||w||}$ und die kanonische Repräsentation erfüllt
- Statt $\frac{1}{||w||}$ zu maximieren können wir auch $||w||^2$ minimieren d.h. die suchen $argmin_{w,b}\frac{1}{2}||w||^2$ + Nebenbedingungen
- Problem: quadratische Programmierung d.h. Konvexe Problemstellung besitzen ein globales Minimum

8.3 Lagrange-Multiplikatoren

- gegeben Funktion $f(x)$ die wir optimieren wollen und eine Bedingung $g(x) = 0$ die erfüllt werden muss
- $\nabla_x f(x) + \lambda \nabla_x g(x) = 0$. Man bezeichnet λ als Lagrange-Multiplikator
- erweiterte Funktion: $L(x, \lambda) = f(x) + \lambda g(x)$
- Optimum muss gelten: $\nabla_x L(x, \lambda) = 0$ sowie $\nabla_\lambda L(x, \lambda) = g(x) = 0$
- Wie mit Ungleichungen?
  1. Optimum liegt innerhalb des Constraintbereichs also $g(x_B) > 0$ und Gradient von g spielt keine Rolle also $\lambda = 0$
  2. Optimum liegt auf dem Constraint also $g(x_A) = 0$ und Gradient ist antiparallel zum Gradient von f d.h. $\nabla f(x) = -\lambda \nabla g(x)$ für $\lambda > 0$

8.4 Bestimmung der Entscheidungsgrenze

- Die Formulierung mit Lagrange-Multiplikatoren $a_i \geq 0$ für die Nebenbedingung ergibt sich zu: $L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i(t_i(w^t \phi(x_i) + b) - 1)$
- Minuszeichen weil wir in Bezug auf w und b minimieren
- Ableitungen:
  - $w = \sum_{i=1}^{n} a_i t_i \phi(x_i)$ (verbraucht den 1/2 Faktor)
  - $0 = \sum a_i * t_i$
- Ableitung einsetzen um w und ?? zu eliminieren
- neuen Problem: $\hat{L}(a) = \sum a_i - \frac{1}{2} \sum_i \sum_j a_i a_j t_i t_j k(x_i, x_j)$ mit $a_i \geq 0$ und $\sum_i a_i t_i = 0$
- Die Kerne $k(x_i, x_j)$ sind hierbei definiert durch $k(x, x') = \phi(x)^t \phi(x')$
- Wenn wir das duale Optimierungsproblem gelöst haem erhalten für die a_i über die sich dann auch b bestimmen lässt
- Klassifikation über: $y(x) = w^t \phi(x) + b$
- Wenn wir das gefundene w einsetzen mit $w = \sum_{i=1}^{n} a_i t_i \phi(x_i)$ ergibt sich: $y(x) = \sum_{i=1}^{n} a_i t_i \phi(x_i)^t \phi(x_i) + b = \sum_{i=1}^{n} a_i t_i k(x_i, x) + b$
- Optimierungsproblem ist definiert durch:
  - Funktion: $\bar{L}(a) = \sum_{i=1}^{n} a_i - 1/2 \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j t_i t_j k(x_i, x_j)$
  - Krush Kuhn-Tucker-Bedingungen:
  - $a_i \geq 0$
  - $t_i y(x_i) - 1 \geq 0$
  - $a_i\{t_i y(x_i) - 1\} = 0$
- für den Punkt x_i gilt also entweder $a_i = 0$ oder $t_i y(x_i) = 1$ d.h. entweder der Punkt spielt in der Klassifikation keine Rolle ODER er hat den minimalen Abstand zur Entscheidungsebene (Stützvektor)
- wenn wir einen Wert a haben dann können wir b ermitteln
- für einen Stützvektor x_j gilt:
  - $t_j y(x_j) = 1$
  - also $t_j(\sum_{i \in S}^{n} a_i t_i k(x_i, x_j) + b) = 1$
- S die Menge der Indizes der Stützvektoren (a_i der anderen Vektoren sind ja ohehin 0)
- numerisch stabilere Lösung: Durchschnitt der b Werte über alle Stützvektoren

8.5 Kerne und Dimensionen

- Was bedeutet $\hat{L}(a) = \sum_{i=1}^{n} a_i - 1/2 \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j t_i t_j k(x_i, x_j)$?
- Ein Kern misst die Ähnlichkeit zweier Vektoren
- um $\hat{L}$ zu maximieren, müssen wir ähnliche x_i, x_j finden die unterschiedliche Klassen angehören $t_i t_j = -1$ und diese mit großen a_i, a_j wichten
- Diese erhöhen den Wert von $\hat{L}$ da die Summe mit negativen Vorzeichen in die Maximierung eingeht
- $w = \sum_{i=1}^{n} a_i t_i \phi(x_i)$ bedeutet dann
- die Summe kann in zwei Teile zerlegt werden: die Summe der Klassen $t_i = 1$ und die Summe der Klasse $t_i = -1$: $w = \sum_{i \in t_i = 1} a_i \phi(x_i) - \sum_{j \in t_i = -1} a_j \phi(x_j)$
- w ist die Differenz des gewichteten Durchschnitts beider Klassen, bestimmt für die interessanten Vektoren
- Quadratische Optimierung mit M Variablen haben eine Komplexität von $O(M^3)$
- in der dualen Formulierung haben wir n Variablen die a_i. Also haben wir so viele Variablen wie Datenpunkte
- ursprünglich hatten wir f Variablen (die Komponenten von w), so viele wie Dimensionen in R^f
- wenn wir eine feste kleine Zahl f von Basisfunktionen haben bringt der Übergang in die duale Formulierung keinen großen Gewinn
- erhebliche Vorteile wenn die Merkmalsräume betrachten deren Dimensionalität die Anzahl der Datenpunkte übersteigt (Extrem: unendlich viel Dimensionen)
- Wie können wir mit unendlich vielen Dimensionen rechnen?
- $\hat{L}(a) = \sum_{i=1}^{n} a_i - 1/2 \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j t_i t_j k(x_i, x_j)$ und $y(x) = \sum_{i=1}^{n} a_i t_i k(x_i, x) + b$ ... wir können R^f als Skalarproddukt zweier transformierter Vektoren repräsentieren durch den Kern: $k(x, x') = \phi(x)^t * \phi(x')$
- d.h. wir können das Skalarprodukt berechnen ohne das Bild der Transformation $\phi(x)$ in R^f selbst bestimmen zu müssen
- Beispiele:
  - Polynome mit Grad f: $k(x, x') = (1 + x^t x')^f$
  - Radiale Basisfunktion (gaußsche Basis): $k(x, x') = exp(-||x - x'||^2 / 2\sigma^2))$
  - Logistischer Sigmoid: $k(x, x') = tanh(k_1 * x^t x' + k_2)$
- Die Skalarprodukte $x^t x'$ und Abstände $||x - x'||$ werden jeweils im ursprünglichen d-dimensionalen Merkmalsraum R^d berechnet

8.6 Überlappende Entscheidungsgrenzen

- Bisher: Annahme dass Punkte perfekt linear separierbar in R^f sind
- Jetzt: Führe zusätzliche Schlupfvariablen $v_i \geq 0$ die das Eindringen von x_i in die verbotene Zone beschreiben
- Ziel: Minimierung einer modifizierten Zielfunktion: $C \sum_{i=1}^{n} v_i + 1/2 * ||w||^2$; $C > 0$ steuert den Kompromiss zwischen der Strafe durch die Schlupfvariablen und der Komplexität der Entscheidungsfläche
- $v_i = 0$ für alle x_i ausserhalb des Randbereiches bzw. auf dem Rand
- $v_i < 1$ für Punkte hinnerhalb des Randbereiches aber auf der richtigen Seite
- $v_i > 1$ falls Punkt auf der falschen Seite
- falls $v_i > 0$ gilt also $v_i = |t_i - y(x_i)$
- Anpassung der Bedingung $t_i y(x_i) \geq 1$ durch Bedingung: $t_i y(x_i) \geq 1 - v_i$
- Die Summe der v_i ist eine Strafe für die Verletzung des Randbereiches. je größer ein v_i desto größer die Verletzung

- in der Zielfunktion muss jetzt nicht nur $||w||^2$ minimiert werden sondern auch die Strafe
- Ziel: min $C \sum_{i=1}^{n} v_i + 1/2||w||^2$
- da für jeden falschen klassifizierten Punkt gilt $v_i > 1$ stellt $\sum_{i=1}^{n} v_i$ eine Obergrenze für die Anzahl der falsch klassifizierten Punkte da
- C ermöglicht es den Kompromiss zwischen Verletzung des Randbreichs un Komplexitäöt der Entscheidungsgrentze zu wählen
- je größer C desto größer wird die Bedeutung v_i desto größer wird tendenziell $||w||$ um v_i klein zu halten
- je größer $||w||$ desto enger umkurvt die Entscheidungsgrenze die Stützvektoren
- bester Wert für C wird sinnvollerweise wieder durch Kreuzvalidierung bestimmt
- Parameter für SVM:
  - C
  - Form des Kerns
  - Parameter des Kerns
- Überanpassung erkennt man an einem schmalen Randbereich (großes w) und einer großen Anzahl der Stützvektoren
- groß ist problemabhängig

9 Nichtmetrische Methoden: Bäume

9.1 Einführung

- Bisher: Klassifikationsferfahren brauchten eine Metrik
- Problem: Nominalskalen haben keine Metrik (Abstandsbegriff)
- Häufig liegen Daten zu Objekten als Eigenschaftslisten vor; Eigenschafte haben keinen Abstandsbegriff
- Lösung: Klassifikation durch Entscheidungsbaum
- Abstieg von Wurzel in die Blätter; jeder Knoten überprüft ein Merkmal
- Vorteil: sehr leicht interpretierbar d.h. Grund für Klassifikation direkt klar da die Merkmalsausprägungen im Muster die Konjunktion der Eigenschaftsbelegungen entlang des Pfades zu Klassifikation erfüllt

9.2 Aufbau von Bäumen

- Betrachte $A \subset D$ der Testdaten
- Wenn alle Instanzen in A derselben Klasse w_A angehören ist die Teilmenge rein, dann kann ein Blattknoten angelegt werden der der Klasse w_A zugeordnet wird
- Wenn die Instanzen in A unterschiedlichen Klassen angehören muss eine Entscheidungs getroffen werden
  - Trotzdem Blattknoten anlegen und Klassifikation gemäß der Mehrheit
  - inneren Knoten anlegen und A in mehere Teilmengen A_j aufspalten; Rekursives Verfahren auf A_j
- Anzahl der Verzweigung
  - Sollte man eine Eigenschaft mit n Ausprägungen mit meiner n-Wege Verzweigung teilen?
  - im allgemeinen werden dadurch die Daten aber zu stark fragmentiert wodurch eine sinnvolle Zerlegung behindert wird
  - eine n-Wege-Zerlegung kann durch mehere binäre Zerlegungen repräsentiert werden deshalb wird dieser Ansatz oft bevorzugt
- Auswahl der Eigenschaft
  - Eigenschaft für die Zerlegung ob diese eine Kombination aus meheren Merkmalen (polytetisch) oder nur ein Merkmal(monothetisch) beinhalte
  - monothetische Eigenschaften zerlegen den Merkmalsraum in Regionen deren Grenzen Hypereben sind die Orthogonal zu den jeweils gewählten Merkmalsdimensionen liegen

9.3 Reinheit von Knotens

- Grundlegendes Prinzip der Merkmalsauswahl ist möglichst einfache Bäume zu erzeugen (flach und wenige Knotens) (vergleiche Ockhams Rasiermesser)
- um dies zu erreichen wählen wir in jedem Knoten N diejenige Eigenschaft T, welche die Reinheit der Kinderknoten maximiert
- Unreinheit eines Knoten definiert durch die Durchmischtheit der dem Knoten zu geordneten Daten in Bezug auf ihre Klassenzugehörigkeit → einfacher definiert als die Reinheits
- die Unreinheit eines Knoten N bezeichnen wir als $i(N)$
- für Reinheitsmaße $i(N)$ soll gelten:
  - $i(N) = 0$ falls deer Knoten rein ist
  - $i(N)$ um so größer je stärker der Knoten durchmischt
- ein Ansatz für Unreinheit: Informationen die wir brauchen um zu einer Instanz $x \in N$ die zugehörige Klasse zu bestimmen
- Wie viel Information brauchen wir falls N ein gemischter Knoten ist in dem die einzelnen Klassen i mit relativen Häufigkeiten p_i enthalten sind und man uns sagt dass $x \in N$ zu Klasse j gehört? $log_2 \frac{1}{p_j} = -log_2(p_j)$ Bits
- Insgesamt ergibt damit der Erwartungswert der Information (Informationsentropie): $i(N) = -\sum_{i=1}^{c} p_i log_2 p_i$
- Basis für die Auswahl einer Zerlegung ist dann die Suche nach einer Eigenschaft T, die zwei Kindknoten N_L, N_R erzeugt die die Unreinheit soweit wie möglich minimiert
- für Informationsentropie: eine Zerlegung die so wenig wie möglich zusätzliche Information erforderlich macht um die Klasse einer Instanz zu bestimmen die im Teilbaum von N liegt (ein Test T der möglichst viel Information liefert)
- Annahme: haben Zerlegung bei der ein Anteil von P_L Instanzen im linken Kindknoten N_L landet und ein Anteil von $(1 - P_L)$ Instanzen im rechten Kindknoten N_R
- Renheitsgewinn: $\delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$
- Maximierung des Reinheitsgewinn durch die sinnvolle Wahl der Eigenschaft T mit deren Hilfe die Zerlegung durchgeführt wird (Wenn Informationsentropie als Maß der Unreinheit repräsentiert $\delta i(N)$ den Gewinn an Informationen in Bits
- weitere Reinheitsmaße:
  - $i_{gini}(N) = \sum_{i=1}^{c} p_i(1 - p_i)$: Gini-Index ist der erwartete Fehler wenn man einer zufälligen Instanz aus N ein zufälligesLabel aus N zuweist. Gini-Index als Maß der Varianz der Zufallsstichprobe in Ns
  - $i_{bayes}(N) = 1 - max_i p_i$: Ist der Bayes-Fehler für die Klassifikation einer Instanz auf BAsis der Stichprobe iN
- unterschiedliche Reinheitsmaße für die Optimierung unterschiedlicher Kriterien
  - Informationentropie: Mittlere Information in Bits die erforderlich ist um eine Instanz im Knoten zu klassifizieren
  - Gini-Index: Varianz der Stichprobe in n (bzgl der Verteilungsfunktion der Klassenlabel)
  - Bayes: Fehler bei Klassifikation gemäß der Bayes-Regels
- Differenzierbarkeit von Gini und Entropie hilfreich wenn das Maximum über einen kontinuierlichen Parameterrraum bestimmt werden muss
- Gini und Entropie reagieren sensitiver auf Änderungen der Klassenverteilungen in den Knoten in Folge eines Splits

9.4 Eigenschaften und Attribute

- Split einer Instanzmenge A wird durch eine Eigenscahft T(x) definiert aus der sich dann die Teilmengen ergeben
- bevorzugt binäre Splits mit boolschen Funktion T(x)
- für die Teilmenge des Splits: $A_L = \{x \in A|T(x)\}, A_R = \{x \in A|\neg T(x)\}$
- binäre Splits: Vorteil eindimensionales Optimierungsproblem, n-äre Splits müssen höherdimensionale Optimierungsprobleme lösens
- wie kann T sinnvoll definiert werden → hängt von den Skalenniveau ab
- metrische Attribute: x_i sucht man typischerweise eine Konstante s so dass ein Test die Form $x_i < s$ annimmt
- Idee: optimierende Funktion $\delta i(N)$ als Funktion von s dar und kann dann numerische Optimierungsverfahren verwenden um s zu bestimmen das den maximalen Gewinn liefert
- Idee geht auch für Kombination von realwertigen Attributen → Split beschreibt eine Hypereben im Teilraum
- man kann auch die nach Merkmal sortierte Liste der Instanzen betrachten um Schnittpunkte s der Form $x_l < s < x_u$ zu bestimmen → wahl durch den Mittelwert, gewichteten Durchschnitt $(1 - P)x_l + Px_u$ wobei P die Wahrscheinlichkeit dafür angibt dass ein Muster im linken Bereich liegt
- ordinale Attribute: Durchmusterung einer ach Attributwert sortierten Liste von Instanzen anwendbar
- nominale Attribute: jede Zerlegung der Menge der Attributwerte in zwei Teilmengen als Schnittpunkt betrachtet → b Attributewerte sind 2^{b-1} mgl Zerlegung (erheblicher Aufwand)
- man könnte ordinale Attribute direkt für die Zerlegung von A in b Teilmegen (b-Wege-Splitss) nutzen: Reinheitsgewinn: $\Delta i(N) = i(N) - \sum_{k=1}^{b} P_k i(N_k)$s mit P_k der Anteil der Instanzen ist, bei denen das Attribut den k-ten Attributwert
- Nachteil: Splits mit großem b bevorzugt auch wenn sie keine sinnvolle Struktur in den Daten repräsentieren (2 Klassen aber 3 Wege
- dehalb muss der Reinheitsgewinn für den Split relativ zur Menge an Informationen betrachtet werden die schon on der Struktur der Aufteilung selbst enthalten $\Delta_b i(N) = \frac{\Delta i(N)}{-\sum_{k=1}^{b} P_k log_2 P_k}$
- Split sind nur lokale Operationen und erzeugen nur ein lokales Optimum ohne sich für die globalen Auswirkungen zu berücksichtigen → Baum entspricht nicht dem globalen Optimum

9.5 Ende des Wachstum

- Wie lang wächst ein Baum?
- Extrem: so lange bis jedes Blatt nur eine Instanz enthälts
- Nachteile:
  - Baum sehr groß
  - Baum hat Trainingsdaten auswendig gelernt → je größer der Baum desto eher enthält Tests die sich auf verrauschte Merkmale beziehen
- Ziel: Performance des Baumes überprüfen mit einem Testdatensatz
- Validierungsverfahren / Lösungens
  - Aufspaltung der Daten für Training und zum Testen, solange Baum neue Ebene generieren bis die Validierung ein Wiederabsinken der Klassifikationsgenauigkeit anzeigt
  - Schwellwert β für den Reinheitsgewinn festlegen → Knoten wird nur gesplittet wernn der Reinheitsgewinn größer als dieser Schwellwert $\Delta i(N) > \beta$ (Vorteil: alle Daten stehen für das Training zur Verfügung)
  - Festlegung dass jedes Blatt mindestens n Instanzen oder einen relativen Anteil von p aller Trainingsdaten enthalten muss

- globales Abbruchkritierum: $\alpha * size + \sum_{leafnodes} i(N)$, size ein Maß für die Größe des Baumes → Baum wächst solange bis die gewichtete Summe aus Komplexität des Baumes und restlicher Unreinheit eine bestimmte Größe erreicht hats, Kompromiss zwischen der Komplexität des Baums und dem restlichen Klassifikationsfehler möglich (Problem: α sinnvoll festzulegen)
- Signifikanztets verwenden um zu prüfen ob ein Split eine statistisch signifikate Verbesserung der Reinheit erzeugt (Chi-Square-Test); Problem: Wachstum kann zu früh beendet werden da nicht vorrausschauend; Lösung: Pruning

9.6 Zurückschneiden

- Horizont-Effekt vermeiden dem umgekehrten Weg wählen: Baum bis maximale Größe wachsen lassen und danach den Baum durch Vereinen benachbarte Knoten auf eine sinnvolle Größe zurückschneiden lassen
- Reduced Error Pruning
  - Zerlegung der Daten in Trainingsdaten und Validierungsdaten
  - Erzeuge maximalen Baum für Traningsdaten
  - für jeden Knoten Prüfe Performance für Validierungsdaten unter Bedingung das der Teilbaum an diesem Knoten gelöscht wird
  - Lösche denjenigen Knoten+Teilbaum für den die Performance maximiert wird
  - Ersetze den Knoten durch ein Blatt mit der Mehrheitsklassifikation des gelöschten Teilbaum
  - Ende wenn Präzision für Validierungsdaten sinkt
- Cost Complexity Pruning
  - Maß für Gesamtkomplexität des Baums als gewichtete Summe von Größe und Restunreinheit,
  - $|T|$ die Anzahl der Blätter eines Baumes T
  - Gesamtkomplexität: $C_\alpha(T) = \alpha|T| + \sum_{m=1}^{|T|} N_m i(m)$ mit N_m die Anzahl der Instanzen im Blatt m ist
  - aus Maximalen Baum wird eine Sequenz immer kleinerer Bäume erzeugt in dem jeweils derjenige innere Knoten kollabiert wird der das Anwachsen von: $\sum_{m=1}^{|T|} N_m i(m)$ minimiert
  - finden von α erfolgt mit Hilfe eines Validierungsansatzen, wähle $\hat{\alpha}$ mit dem das beste Ergebnis für einen separaten Testdatensatz erreicht wird
- Regelansatz (C4.5)
  - Pfad von Wurzel zu Blatt als Regel darstellen: Konjunktion aller Eigenscahften
  - Regeln können unabhängig voneinander durch Löschen von Vorbedingungen beschnitten erden so dass die Präzision verbessert
  - resultierende Regelmenge wird dann in der Reihenfolge der geschätzten Präzision sortiert
  - für unterschiedliche Pfade kann unterschiedlich beschnitten werden ohne eine Reorganisation des Baumes erforderlich ist

9.7 Klassifikation

- reine Blätter erhalten das homogene Klassenlabel der ihrer Instanzen
- unreine Blätter erhalten das Klassenlabel der Mehrheit

9.8 Wahl der Attribute

- Es ist hilfreich wenn die Attribute gut gewählt werden, so dass die Tests im Baum auch senkrecht zu den effektiven Attributdimensionen sind (Vorverarbeitung durch PCA)

9.9 Fehlende Attribute

- Situation: Fragenbögen sind unvollständig
- Lösung 1: Weise den häufigsten Wert alle Instanzen im aktuellen Knoten zu
- Lösung 2: erzeuge virtuelle Punkte, für jeden möglichen attributwert einen, jeder Pu nkt erhält ein Gewicht entsprechend der Häufigkeit des Wertes, Führe für alle virtuellen Punkte die weitere Klassifikation durch; bestimme die endgültige Klasse durch Abstimmung unter den q Punkten
- Während der Konsturktion des Baumes: so tun als wäre der Punkt nicht vorhanden

9.10 Bewertung

- Umgang mit unterschiedlichen Skalenniveaus
- Fehlerbehaftete Daten, fehlende Daten, viele Attribute
- nichtlineare Entscheidungsgrenzen

10 Algorithmenunabhängige Verfahren

10.1 Der beste Klassifikator

- Gegeben Klassen Ω und Merkmalsraum X
- Menge der möglichen Werlten wird durch Funktion $P(\omega|x)$ beschrieben die angibt wie wahrscheinlich es ist einer gegebene Welt dass x der Klasse ω angehört
- Ziel des Klassifikators: diese Funktion (die posteriori Wahrscheinlich) aus einer Menge von Traningsdaten rekonstruieren
- bester Klassifikator ist derjenige Klassifikator der für alle denkbaren Funktionen $P(\omega|x)$ einen geringeren Klassifikationsfehler liefert als jeder andere
- Was müssen wir machen damit wir überhaupt Aussage über die Form von $P(\omega|x)$ machen können?
- benötigen Annahmen über wie wir sich die unbekannte Funktion von bekannten Werten in unbekannten Regionen extrapolieren lässt
- Annahmen bedeutet einfach dass wir unterstellen dass bestimmte Formen der Funktion $P(\omega|x)$ wahrscheinlicher sind als andere
- Wenn $P(\omega|x)$ beliebige Funktion sein kann was folgt?
- falls $P(\omega|x)$ nicht zur Annahme passt kann der Klassifikator damit nicht gut zurechtkommen
- für die jenigen P deren Form gut zur Annahme des Klassifikators passen liefert er Ergebnisse die besser sind als bloßes Raten
- diejenigen P deren Form der Annahme widerspricht liefert er Ergebnisse die schlechter sind als Raten
- Also Das, was ein Klassifikator durch seine Annahme gegenüber bloßem Raten in einem Bereich des Raums der Funktionen P gewinnt, verliert er notwendigerweise in einem anderen Bereich dieses Raums.
- Gemittelt über alle P ist jeder Klassifikator so gut wie raten d.h. jeder Klassifikator ist gleich gut
- No Free Lunch Theorem: „Gemittelt über alle möglichen Funktionen P haben alle Klassifikatoren die gleiche Leistung bei der Klassifikation neuer Daten. “d.h. es gibt keinen besten Klassifikator der immer passt

10.2 Evaluierung von Klassifikatoren

- Test auf Trainingsdaten nicht aussagefähig da Klassifikator kann auswendig lernen
- bei genügend Daten können wir Aufteilen in zwei Mengen: Test- und Evaluations-Daten
- Parameter der Klassifikatoren benötigen wir ebenfalls ein Teil der Daten $\rightarrow$ Validierungsdaten
- Also Aufteilung der Gesamtdaten in 3 Teile: Trainingsdaten (50%), Validierungsdaten (25%), Testdaten(25%)
- Leistungskriterium ist die Fehlerrate bei den Testdaten (Anteil korrekt klassifizierter Instanzen).
- Wichtig: Testdaten nicht für Training oder Validierung verwenden!!!
- Typisch: Traning bis Validierungsleistung wieder schlechter wird, anschließend Leistungsschätzung mit Testdaten
- nach der Auswahl des besten Klassifikators: Training auf allen Daten
- Also: je größer Menge der Trainingsdaten desto besser der Klassifikator, je größer die Menge der Testdaten desto bessere Fehlerschätzung
- Problem: viele Daten
- Lösung: Kreuzvalidierung, Bootstrap

10.3 Schätzung des Fehlers

- Gegeben Fehlerrate von 25% wie sicher ist es dass dies die tatsächliche Fehlerrate ist?
- Annahme: Klassifikator liefert mit Wahrscheinlichkeit p das richtige Ergebnis
- Test mit $N = 1000$ hat also $S = 750$ korrekte Klassifikationen also Erfolgsrate von 75%, wie sicher ist das Ergebniss?
- Betrachtung als Sequenz von 1000 Bernulli-Expiermente die mit Wahrscheinlichkeit von p Erfolg liefern: Mittelwert und Varianz für Bernoulli-Expiment sind p bzw. p(1-p)
- für Folge von N Experimente ist die emprische Erfolgsrate $f = S/N$ eine Zufallsvariable mit dem Erwartungswert p und der Varianz $p(1-p)/N$
- für große N ist f annhährend Normalverteilt mit Mittelwert p
- Frage: Wie gross ist das Intervall in dem p mit einer Sicherheit von z.B 80% liegt?
- Konfidenzintervall für p: $z = \frac{f-p}{p*(1-p)*N}$ dies ist nun $N(0,1)$ verteilt
- man Erhält die Gleichung: $P(-b \leq z \leq b) = 80\%$
- Intervallgrenzen b kann man für die Normalverteilung nachschlagen, insgesamt quadratische Gleichung da p auflösen
- Approximation durch Normalverteilung aber nur für große N (N > 100)

10.4 Kreuzvalidierung

- wenn nicht genügend Daten können auch mehere Tests durchgeführt werden und der Durchschnittsfehler angegeben werden
- k-fache Kreuzvalidierung werden die Daten zufällig in k Teilmengen zerlegt (oft stratifizierte Teilmengen für geringere Varianz d.h. der relative Klassenanteil ist gleich)
- k Läufe $i = 1, \ldots, k$ bei denen Teilmenge i als Testdaten verwendet wird und der Rest als Trainingsdaten
- üblich: k=10
- Leave-one-out Kreuzvalidierung: N-fache Kreuzvalidierung
  - Verwendet nahezu alle Daten als Trainingsdaten
  - Beinhaltet keinen Zufall bei dem Aufbau der Teilmengen (keine Varianz)
  - rechenintensiv
  - garantiert nicht stratifizierte Testdaten
  - Bsp: Feature Zufallszahlen, Klassen Hälfte 1 andere 0ß
  - Bester Klassifikator: Entscheidung für die häufigste Klassen
  - Leave-One-Out hat 100% Fehler da ausgelassenes Datum dafür sorgt das die gegensätzliche Klasse mehrheitlich vorhanden ist

10.5 Bootstrap

- Kreuzvalidierung: Stichprobe ohne zurücklegen
- Bootstrap: Stichprobe mit zurücklegen d.h. Datum kann mehrfach gezogen werden
- Alle nicht in Bootstrap-Probe enthaltene Daten als Trainingsdaten benutzen
- Speziell: 632-Bootstrap
  - Chance: n-Elementige Menge bei n Auswahlvorgängen nicht gewähtlt zu werden: $(1 - \frac{1}{n})^n \approx 0.368$ d.h eine Bootstrap Stichprobe der Größe n enthält eta 63.2% der insgesamt verfügbaren Instanzen
  - Problem: Fehlerschätzung konservativ da bei Kreuzvalidierung mit k=10 90% der Trainingsdaten zur Verfügung stehen
  - realisitische Fehlerschätzung: pessimistische Schätzung mit der optimistischen Schätzung kombinieren: $error = 0.632 * error_{test} + 0.368 * error_{training}$

Literatur

[1] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.