Data Mining

Alexander Hinneburg

SS 2009

Inhaltsverzeichnis

1 Lehr- und Lernmethoden

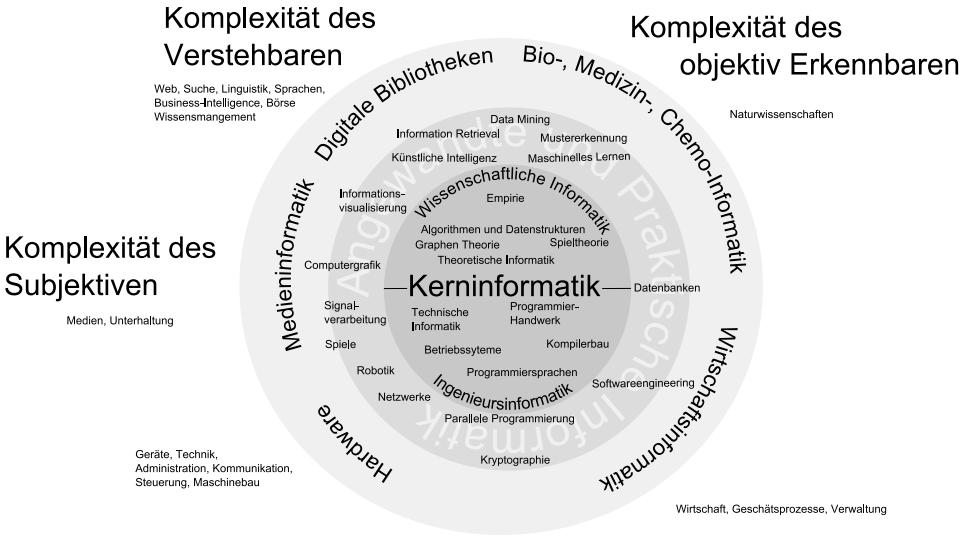
1.1 Verortung des Gebiets, Fernziele

Data Mining, was ist das?

- Motivation ist das Wichtigste beim Lernen
- Fragen zur Motivation
  - Warum soll ich mich mit Data Mining beschäftigen?
  - Kann ich Data Mining mit Gewinn nebenbei hören?
  - Ist Data Mining nur eine Modeerscheinung?
  - Brauche ich die ganze Mathematik für das eigentliche Data Mining?
  - Muss ich hier viel programmieren?

Einordnung von Data Mining

- Welt der Informatik



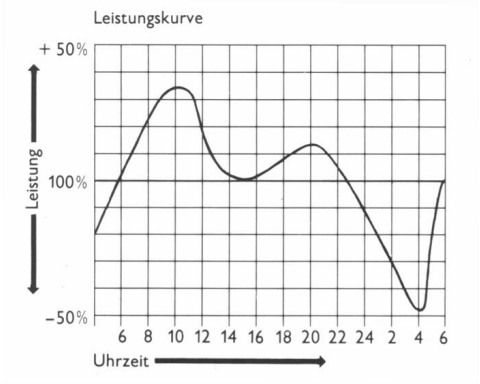
Das Problem Mathematik

- Data Mining und Maschinelles Lernen importiert Erkenntnisse aus Mathematik/Statistik
  - Stoff aus den 70-ern des letzten Jahrhunderts
  - heute in großen Maßstab anwendbar
- Gegen die Krankheit der Modewörter und Abkürzungen hilft nur Mathematik
- Mathematik ist ein Wettbewerbsvorteil
- Gut ausgebildete Absolventen werden gebraucht,
  Sie sollen diese Menschen sein.
- Gestaltung der Vorlesung
  - Weniger Stoff dafür lieber gründlich, dass Sie es verstehen
  - Aufspaltung der Übung in Besprechungsteil und Praxisteil

1.2 Gestaltung der Vorlesung

Unterrichts- und Lernorganisation 1/2

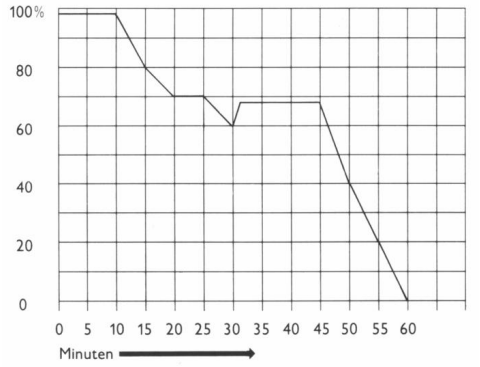
- Leistungsverhalten über den Tag, REFA-Normkurve[1]



- Allgemeine Aussagen
  - Der Leistungshöhepunkt liegt am Vormittag.
  - Erneutes Zwischenhoch am frühen Abend.
- Folgerung
  - Bemühen Sie sich um 8 Uhr zur Vorlesung
  - Wiederholen Sie am frühen Abend die Vorlesung

Unterrichts- und Lernorganisation 2/2

- Leistungswerte der Konzentration im Verlauf von 60 Minuten:



- Folgerung
  - Nach 45 Minuten Pause machen

Zeitliche Aufteilung

Besprechung 8:15 – 9:00 Uhr, Besprechung der Übungen, Wiederholung

10 Minuten Pause

Vorlesung I 9:10 – 9:55 Uhr

10 Minuten Pause

[1] `http://www.gm.fh-koeln.de/~bundschu/dokumente/Referate/358/`

Vorlesung II 10:05 – 10:50 Uhr

10 Minuten Pause

Praxis 11:00 – 11:45 Uhr, Bearbeiten von Beispielen

Aufbereitung des Lernstoffs

- Gesagt ist nicht gehört
- Gehört ist nicht verstanden
- Verstanden ist nicht behalten
- Behalten ist nicht gekonnt
- Gekonnt ist nicht angewendet
- Angewendet ist nicht beibehalten

Konrad Lorenz

Wir behalten

- 10% von dem, was wir lesen
- 20% von dem, was wir hören
- 30% von dem, was wir sehen
- 50% von dem, was wir hören und sehen
  - Bilder und Skizzen machen
- 70% von dem, was man selbst sagt
  - Fragen stellen, Übungen vorrechnen, Stoff wiederholen
- 90% von dem, was man selbst tut
  - Übungen machen, Zusammenfassungen erarbeiten

Quelle: Roland Spinola, Weiterbildung 4/1998

Aufbereitung des Lernstoffs

Je mehr Wahrnehmungskanäle angesprochen werden, desto höher ist die Behaltensquote.

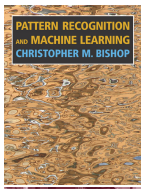
Zur Arbeit mit dem Skript

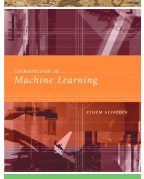
- Es wird ein Skript gegeben
- Viele wichtige Sachen sind nicht im Skript enthalten, weil
  - Formeln an der Tafel entwickelt werden
  - Argumente besprochen werden
- Für Sie ist es wichtig von der Tafel und Diskussion mitzuschreiben
- Mitschrieb-Wiki ist Ihr Beitrag zum Skript

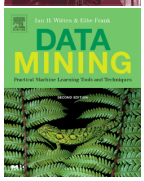
Nehmen Sie das Skript nicht wörtlich
Nachdenken, Nachlesen, Nachfragen

Bücher und Material

- Christopher M. Bishop: Pattern Recognition and Machine Learning. (Viele Abbildungen sind aus dem Buch)
- Ethem Alpaydin: Introduction to Machine Learning (auch in Deutsch).
- Ian H. Witten, Eibe Frank: Data Mining: Practical Machine Learning Tools and Techniques (Second Edition).
- David Heckerman: A Tutorial on Learning with Bayesian Networks `http://research.microsoft.com/en-us/um/people/heckerman/`







Organisation der Vorlesung 1/2

- Vorlesung und Übung finden Mi. 8:15-11:45, Raum 1.27 statt.
- Der Stoff aus Vorlesung und Übung ist prüfungsrelevant.
- Die Vorlesung hat 15 Wochen und ist in drei Teile gegliedert
  - Teil 1 geht von der ersten bis zur 4. Woche
  - Teil 2 geht von der 6. bis zur 9. Woche
  - Teil 3 geht von der 11. bis zur 14. Woche
- In der 5., 10. und 15. Woche werden die Klausuren zur Vorlesungszeit (jeweils 90 min) geschrieben.

Organisation der Vorlesung 2/2

- Es gibt keine Voraussetzungen, um an den Klausuren teilnehmen zu können. Es wird empfohlen die Übungen zu machen.
- Für die Wirtschaftsinformatiker zählen die besten beiden Klausuren von dreien mit jeweils 50 Fachpunkten. Bekanntgabe der Ergebnisse sind jeweils 2 Wochen nach der Klausur.
- Für WI-Inf ist das eine studienbegleitende Prüfung mit 5 LP für Vorlesung und Übung für mindestens 50 Fachpunkte (insgesamt) erbracht werden müssen.

Organisation der Übung

- Die Übungsblätter werden immer am Mittwoch zur Übungszeit ins Netz gestellt.
- Die Übungen sind eine Woche später bis Mittwoch 8.00 Uhr elektronisch mittels Subversion (SVN) abzugeben.
- Übungsgruppen von zwei-drei Personen sind zulässig.
- Zum Vorstellen der Übungsaufgaben muss eine kleine Präsentation in PDF vorbereitet werden.

Arbeitsaufwand und Fallen

- Nicht zu viele Vorlesungen, 20 SWS sind OK.
- Vorlesungen werden zum Ende hin schwerer.
- Vergleich: Brettspiel Keltis

2 Data Mining Einführung

Data Mining Einführung 1/2

- Ziele und Motivation
  - Entdeckung von unbekanntem, nützlichem, interessantem und verstehbarem Wissen, Hypothesen, Fakten
  - Daten wurden oft nicht für Data Mining gesammelt
  - Datensammlungen wachsen ständig

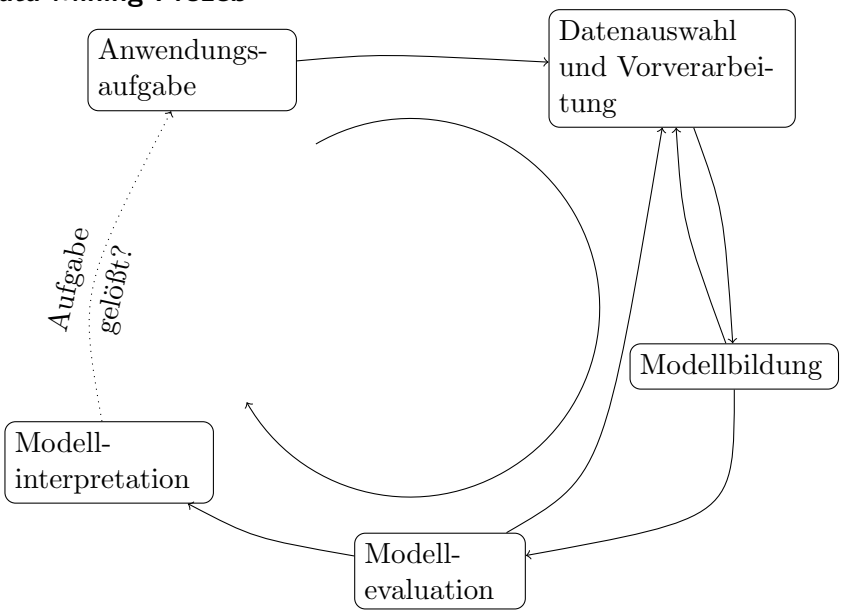
    Turning data grave yards into gold mines.
- Geschichte
  - Beginn 1993 mit Datenbank-Workshops
  - Seit 1995 eigene Konferenzen, ACM SIGKDD, IEEE ICDM, SIAM SDM, European ECML/PKDD, PA-KDD
  - Seit 1999 eigene Gesellschaften ACM SIG-KDD, GI-AG KDML
  - Seit 2004 teilweise Konvergenz mit Maschinellem Lernen und Information Retrieval

Data Mining Einführung 2/2

- Möglichkeiten und Unmöglichkeiten
  - Ziel: Modell der Wirklichkeit
  - Arten von Modellen
    - Entity-Relationship (ER) Modell, Relationales Schema, Objektorierentiertes (OO) Modell
    - Hidden Markov-Modell, Gaussisches Mischmodell
  - Flaschenhals-Methode
    - Trennung von relevanten Informationen vom Rauschen
    - Kompression: Probabilistische Modelle, Kodierungstheorie

2.1 Data Mining Prozeß

Data Mining Prozeß



Typen von Anwendungsaufgaben 1/3

- Beschreiben und Reduzieren
  - Was steckt in den Daten?
  - Beispiele
    - Kundensegementierung
    - Kleidenkonfektionsgrößen
    - Themen in Dokumentsammlungen

Typen von Anwendungsaufgaben 2/3

- Klassifizieren
  - Gegeben Beispiele, lerne Methode Objekte in Klassen/Kategorien einzuordnen
  - Beispiele
    - Treue Kunden / Wechselkunden
    - Spam / normale Emails
    - Autos
- Regression
  - Gegeben Beispiele, lerne Methode einem Objekt einen numerischen, geordneten Wert zuzuweisen
  - Beispiele
    - Noten geben, Prüfungen bewerten
    - Bewertungen im Web

Typen von Anwendungsaufgaben 3/3

- Vorhersage
  - Gegeben eine Zeitreihe, setze die Reihe sinnvoll fort
  - Beispiele
    - Wettervorhersage
    - Anzahl den Anwesenden in der Vorlesung beim nächsten Termin
    - Wichtigkeit eines Themas in den Veröffentlichungen im nächsten Jahr
- Zusammenhänge/Beziehungen/Kausalitäten
  - Lerne aus den Daten: Regeln, Netzwerke, Themen/Konzepte
  - Beispiele
    - Kunden, die dieses Buch kauften, haben auch jenes gekauft.

Datenauswahl und Vorverarbeitung

- Daten müssen repräsentativ sein
- Daten sollen kein unnötiges, leicht entfernbares Rauschen enthalten
- Daten müssen informativ sein
- Daten müssen schlank sein
- Hilfsmittel
  - Datenbanken und Data Warehouses
  - Normalisierungsstandards, Reduktion der Variabilität
  - Einfache Analysen und Wichtungsschemata
  - Definition von beschreibenden Attributen (Feature-Extraction)

Modellbildung

- Wahl der Modellklasse, Aufbau der Pipeline
- Einstellen und Tunen der Parameter
- Wahl der Trainingsdaten
- Wahl der Trainingsmethoden
- Wahl der Initialisierung des Trainings

Modellevaluation

- Schätzung des Modellfehler
  - Passt das Modell überhaupt auf die Daten?
- Konfidenzintervalle des Modellfehlers
- Vergleich mit Grundwahrheit (Goldstandard)
- Systematische Methoden zur effektiven Ausnutzung der Daten
  - Kreuz-Validierung
  - Leave-One-Out
  - Bootstrap
  - Permutationstests
- Test gegen Null-Hypothese
  - Rolle des Advocatus Diaboli

Modellinterpretation

- Semantische Deutung des Modells
- Plausibilitätsvergleich der gelernten Ergebnisse mit Hintergrundwissen
- Analyse von Fehlschlägen
- Visualisierung, Verdichten von Informationen

Ethische Fragen

- Werden durch die Ergebnisse Rechte verletzt
  - Persönlichkeitsrechte
  - Urheber- und Datenschutzrechte
  - Vertrauliche Informationen
- Privacy Preserving Data Mining
  - Definition neuer Begriffe
  - Echte Beiträge in der Methodik
- Soziale Implikationen
- Missbrauchsszenarien

2.2 Beispiel: Polynom-Kurvenanpassung

Probleme beim Data Mining

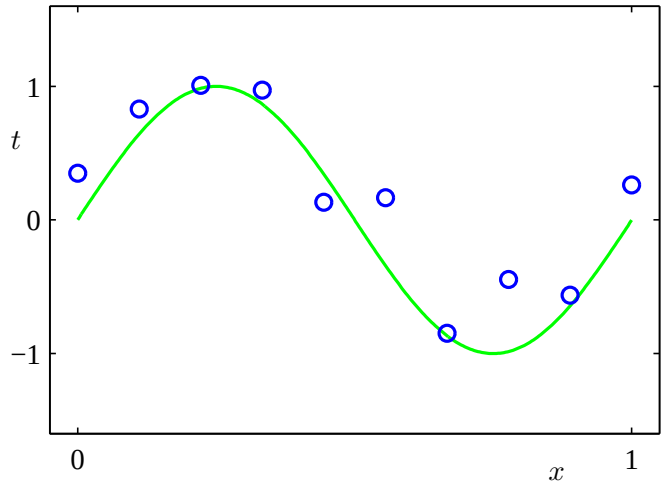
- Wie sehen Data Mining Modelle aus?
- Worin besteht das Lernen?
- Was sind die Schwierigkeiten bei der Wahl der Parameter?
- Was ist Over-fitting?
- Einfluß der Modellkomplexität
- Einfluß der Datenmenge
- Regulierung der Komplexität von Modellen beim Lernen
- Beispiel: Polynom-Kurvenanpassung
  - Keine großen theoretischen Voraussetzungen
  - Viele Probleme lassen sich anschaulich erklären
  - Leider keine grundlegende Theorie dahinter

Beispielproblem: Polynom-Kurvenanpassung

- Problemstellung:
  - Gegeben numerische Eingabe x, ordne eine numerische Ausgabe y zu.
  - Beispieldaten:
    - N Beobachtungen $\vec{x} = (x_1, \ldots, x_N)^T$ (geschrieben als Spaltenvektor)
    - mit zugehörigen Ausgabewerten $\vec{t} = (t_1, \ldots, t_N)^T$.
  - Problemtyp:
- Synthetische Daten für Lernspiel
  - $x_n, n = 1, \ldots, N$ gleichverteilt in $[0, 1]$.
  - $\vec{t}$ berechnet durch $\sin(2\pi x)$ plus Gaußverteiltes Rauschen
- Ziel
  - Modell: neuen Eingaben $\hat{x}$ Ausgaben $\hat{t}$ zuordnen.

Synthetische Daten für Lernspiel

- $N = 10$ Ein- und Ausgaben
- Daten sind blaue Kreise
- Grüne Kurve $\sin(2\pi x)$



Modellklasse

- Modellklasse der Polynome vom Grad M
- Polynomfunktion der Form

$$y(x, \vec{w}) = w_0 + w_1 x + w_2 x^2 + \dots w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1}$$

- M ist Ordnung des Polynoms
- Koeffizienten $\vec{w} = (w_0, w_1, \dots w_M)^T$
- Polynomfunktion $y(x, \vec{w})$ ist eine nichtlineare Funktion bezüglich x, aber eine lineare Funktion bezüglich der einzelnen Koeffizienten w_j.

Fehlerfunktion

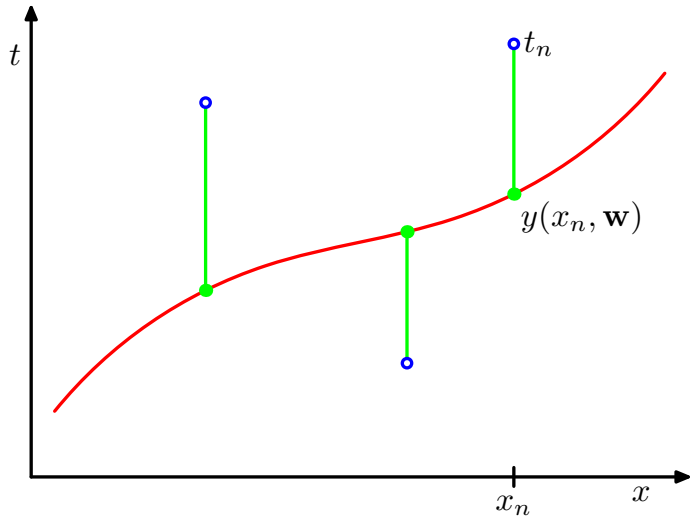
- Anpassen der Parameter des Modelles, die Koeffizienten $\vec{w}$ an Trainingsdaten
- Optimierungsproblem: minimiere Fehlerfunktion

$$E(\vec{w}) = \frac{1}{2} \sum_{n=1}^{N} [y(x_n, \vec{w}) - t_n]^2 \tag{2}$$

- Nichtnegative Größe
- Null, wenn Polynom alle Trainingspunkte berührt
- Alternative Fehlerfunktionen?
- Wie kann man ein Optimierungsproblem lösen?

Geometrische Interpretation der Fehlerfunktion

- $E(\vec{w})$ ist Summe der quadrierten grünen Längeneinheiten



Ideen zur Lösung des Optimierungsproblems

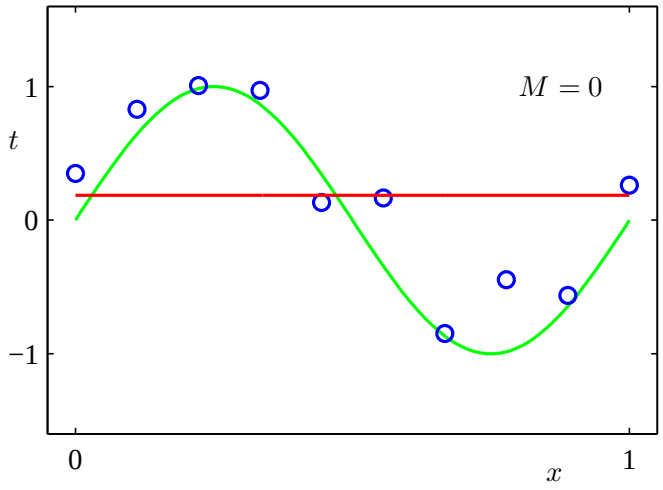
- Fehlerfunktion ist quadratisch in Koeffizienten w_j
  $\Rightarrow$ Abbleitungen nach w_j sind linear in w_j.
- Abbleitung Null setzen $\Rightarrow$ Lösung eines Gleichungssystems
- Eindeutige Lösung $\vec{w}^*$
- Polynom $y(x, \vec{w}^*)$ gibt die zugehörige Funktion (Modell)

Modell-Auswahl

- Offene Frage
  - Wie wird M gewählt?
  - Beliebige Werte für $M = 0, 1, \ldots$ sind möglich
- Erster Ansatz
  - Probiere Werte $M = 0, 1, 3, 9$

Ergebnisse 1/4

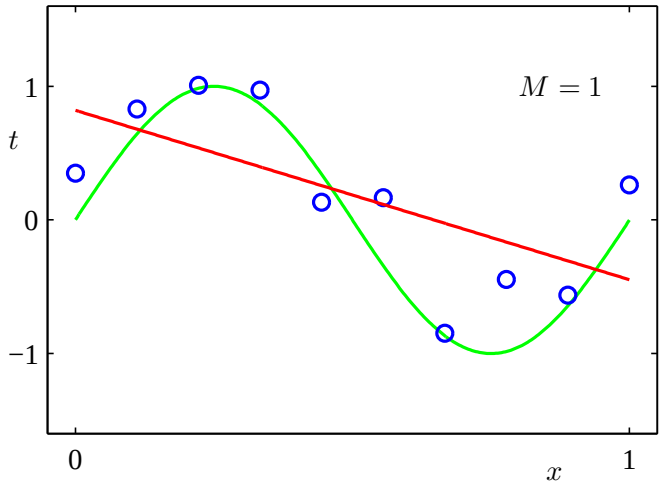
- $M = 0$



- Visueller Eindruck: schlechtes Modell

Ergebnisse 2/4

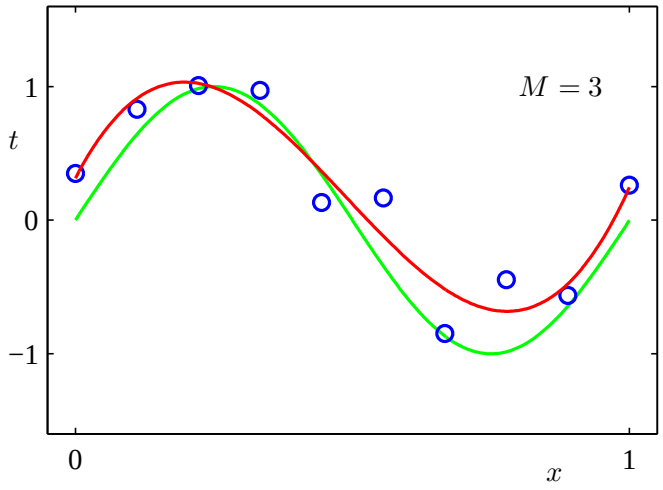
- $M = 1$



- Visueller Eindruck: schlechtes Modell

Ergebnisse 3/4

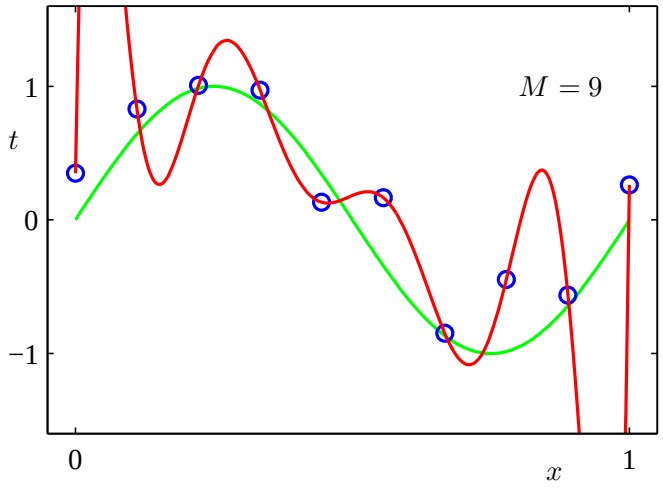
- $M = 3$



- Visueller Eindruck: paßt ganz gut, wenn auch nicht zu 100%

Ergebnisse 4/4

- $M = 9$



- Visueller Eindruck: paßt zu 100%, Polynom sieht seltsam aus
- Over-Fitting

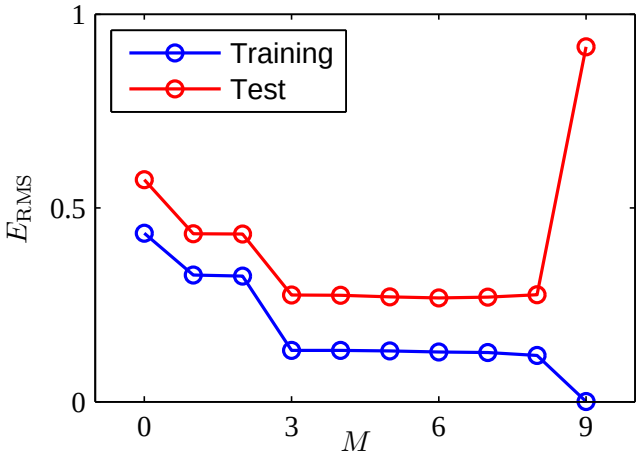
Evaluation des Modells

- Modell zum Zuordnen von Ausgaben zu neuen Eingaben
- Testdaten mit 100 Datenpunkten (gleiche synthetische Erzeugung)
- Evaluation
  - Berechne für jeden Wert von M die Parameter $\vec{w}^*$
  - Berechne Fehlerfunktion $E(\vec{w}^*)$ jeweils für Trainings- und Testdaten
- Normalisierung des Fehlers, Root-Mean-Square Fehler (RMS)

$$E_{RMS} = \sqrt{2E(\vec{w}^*)/N} \tag{3}$$

Trainings- und Testfehler

- RMS für Trainings- und Testdaten



- $3 \leq M \leq 8$ liefert sinnvolle Ergebnisse
- Modell für $M = 9$ verallgemeinert nicht gut

Diskussion 1/2

- Ergebnisse sind paradox
  - Modell $M = 9$ enthält alle anderen Modelle als Spezialfall
  - $M = 9$ sollte mindestens genauso gut abschneiden wie $M = 3$
- Annahme: $\sin(2\pi x)$ ist bestes Modell
  - Taylor-Reihe von

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \text{ für alle } x$$

    enthält alle höheren Potenzen
  - also sollte die Qualität mit steigendem M besser werden

Diskussion 2/2

- Inspektion der Lösungen für verschiedene M

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

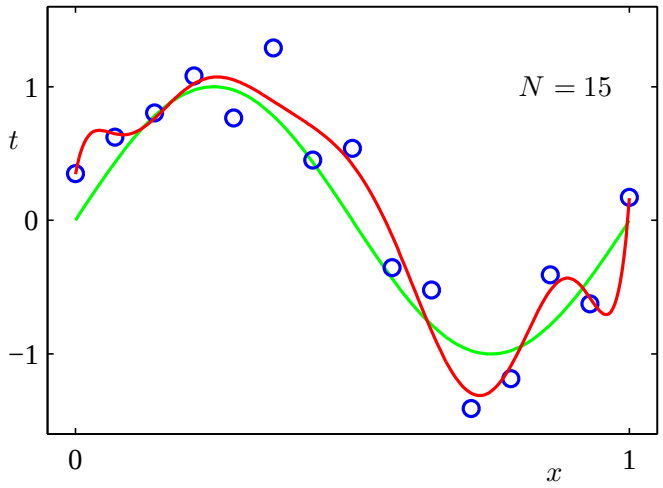
- Koeffizienten haben mit steigendem M größere Skale
- Für $M = 9$ wird das Rauschen mitgelernt
  - Kosten: komplizierte Oszillationen zwischen den Datenpunkten

Abhängigkeit von der Datenmenge

- Größere Datenmenge, weniger Over-Fitting
- Je mehr Daten, desto komplexere Modelle können gelernt werden
- Heuristik
  - Anzahl der Datenpunkte sollte größer als f·Anzahl der Parameter sein,
  - $f = 5$ bis 10
- Mehr Datenpunkte sind meist teuer in
  - Beschaffung
  - Rechenkapazität

Abhängigkeit von der Datenmenge

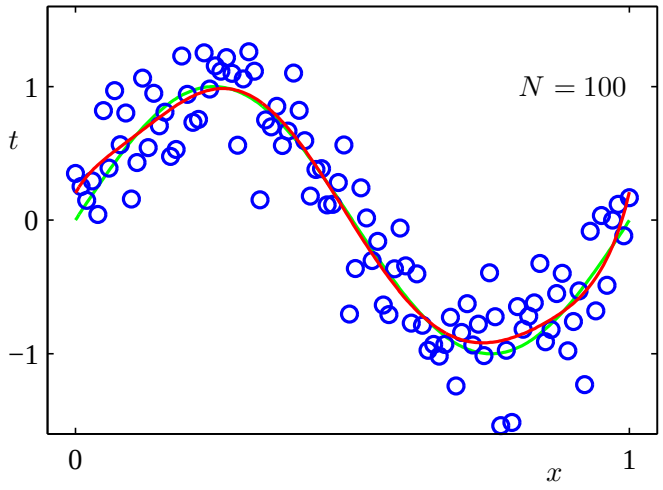
- Abnahme des Over-Fitting-Problems mit größeren Datenmengen



- Minimiere Fehlerfunktion (2) mit $M = 9$

Abhängigkeit von der Datenmenge

- Abnahme des Over-Fitting-Problems mit größeren Datenmengen



- Minimiere Fehlerfunktion (2) mit $M = 9$

Alternativer Umgang mit Overfitting

- Abhängigkeit der Modellkomplexität von Größe der Datenmenge ist unbefriedigend
- Modellkomplexität sollte dem Problem angepaßt sein
- Bisheriger Lernansatz entspricht Maximum-Likelihood-Schätzer
- Bayessche Schätzer vermeiden Overfitting durch Regulierungstechniken

Regulierung von Modellparametern

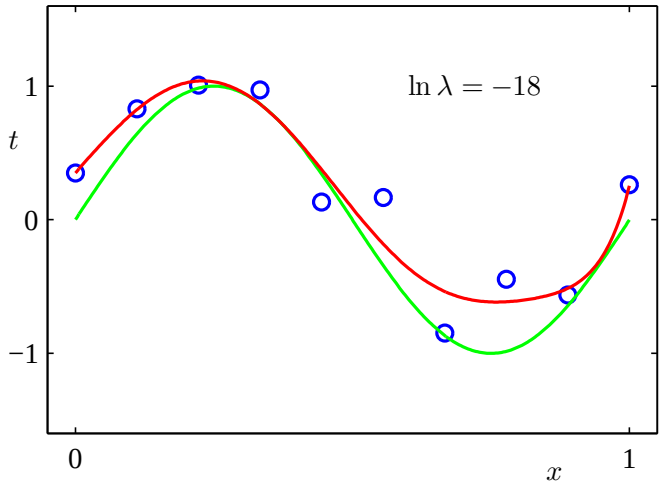
- Ziel
  - Vermeide Lösungen mit großen Absolutwerten (führt zu Oszillationen)
- Idee
  - Einführen eines Strafterms in die Fehlerfunktion
  - Bestraft große Absolutwerte

$$\tilde{E}(\vec{w}) = \frac{1}{2}\sum_{n=1}^{N}[y(x_n, \vec{w}) - t_n]^2 + \frac{\lambda}{2}\|\vec{w}\|^2 \tag{4}$$

- $\|\vec{w}\|^2 = \vec{w}^T\vec{w} = w_0^2 + w_1^2 + \ldots + w_M^2$
- In Abhängigkeit von λ ist der zweite Term groß, wenn die Absolutwerte der Parameter groß sind
  - Lösungen mit Oszillationen bekommen größeren Fehler zugewiesen

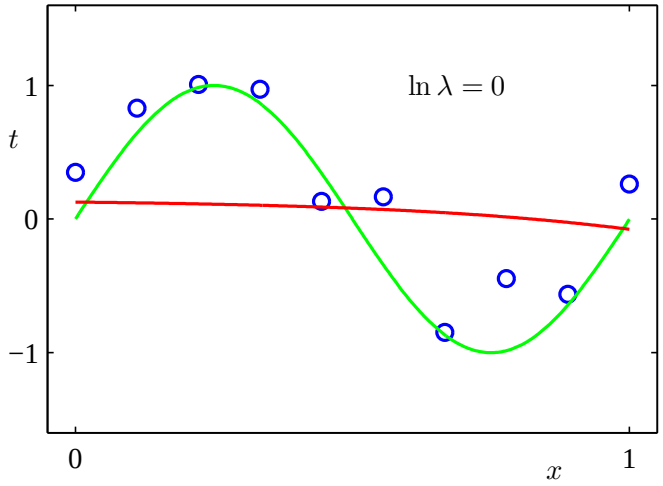
Regulierung, Beispiele

- $M = 9$, $\ln \lambda = -18, \Rightarrow \lambda = 1,523 \cdot 10^{-8}$



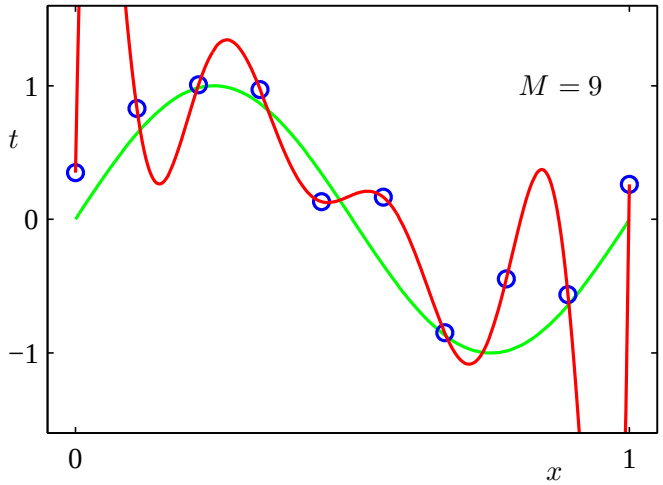
Regulierung, Beispiele

- $M = 9$, $\ln \lambda = 0, \Rightarrow \lambda = 1$



Regulierung, Beispiele

- $M = 9$, $\ln \lambda = -\infty, \Rightarrow \lambda = 0$



- Ist Modell ohne Regulierung

Inspektion der Koeffizienten

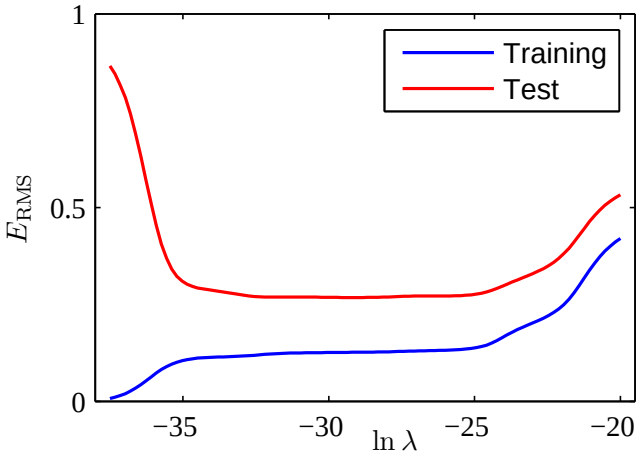
- $M = 9$ und 10 Datenpunkte

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.3	-31.97	-0.05
w_4^*	-231639.30	3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

- Regulierung reduziert Absolutwerte der Parameter
- Parameter λ kontrolliert diesen Effekt

Einfluß der Regulierung auf Fehler

- $M = 9$, 10 Datenpunkte Trainingsdaten



Verfahren zum Lernen des Modells

- Einfache praktische Bestimmung der Modellkomplexität
  - Partitioniere Daten in Trainings-, Validierungs- und Testdaten
  - Nutze Trainingsdaten um Parameter $\vec{w}^*$ zu bestimmen
  - Nutze Validierungsdaten um Modellkomplexität zu bestimmen (M oder λ)
  - Nutze Testdaten um Modellqualität zu bestimmen
- Relativ verschwenderischer Umgang mit Daten, später sparsamere Verfahren
- Bisher alles ad hoc per Intuition eingeführt, später alles auf solider Grundlage von Wahrscheinlichkeittheorie

3 Wahrscheinlichkeitstheorie

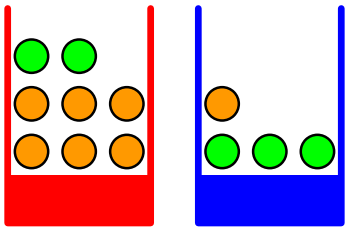
Wahrscheinlichkeitstheorie

- Grundkonzept für Data-Mining Modelle
- Konsistente Theorie zur Quantisierung und Manipulation von Informationen über Unsicherheit
- Kombination mit Entscheidungstheorie
- Enge Verbindung mit Informations- und Kodierungstheorie
- Interpretationen von Wahrscheinlichkeit
  - Häufigkeit
  - Maß für Unsicherheit (Bayessche Wahrscheinlichkeit)
    - Aussagen über nicht wiederholbare Ereignisse bei unvollständigen Informationen

3.1 Wahrscheinlichkeitsregeln

Einfaches Beispiel

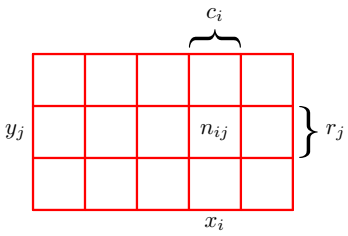
- Auswahlprozeß
  - Zufälliges Auswählen der Kiste
    - Rote Kiste 40%
    - Blaue Kiste 60%
  - dann zufällig Frucht ziehen



- Zufallsvariablen
  - B für Kiste
    - Belegungen: r (rot), b (blau)
    - $P(B = r) = 4/10$, $P(B = b) = 6/10$
    - Wahrscheinlichkeiten aller Alternativen summieren zu Eins
  - F für Frucht
    - Belegungen: a (Apfel), o (Orange)
- Fragen
  - Was ist die Wahrscheinlichkeit einen Apfel zu ziehen?
  - Wenn eine Orange gezogen wurde, was ist die Wahrscheinlichkeit, daß sie aus der blauen Kiste kommt?

Summen- und Produktregel 1/2

- Zwei Zufallsvariablen
  - X, Werte $\{x_i\}$, $i = 1, \ldots M$
  - Y, Werte $\{y_j\}$, $j = 1, \ldots L$



$M = 5, L = 3$

- Beobachtungen
  - Insgesamt N Instanzen von Paaren (x_i, y_j)
  - Anzahl Instanzen für spezielles Paar $X = x_i$ und $Y = y_j$ ist n_{ij}
  - Anzahl Instanzen in Spalte c_i und Zeile r_j
- Verbundwahrscheinlichkeit

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} \tag{5}$$

- Randwahrscheinlichkeit

$$p(X = x_i) = \frac{c_i}{N}, \quad c_i = \sum_{j=1}^{L} n_{ij} \tag{6}$$

Summen- und Produktregel 2/2

- Summenregel

$$p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j) \tag{7}$$

  - Ergibt sich aus Gleichung (5) und (6)
- Wenn $X = x_i$ festgehalten
- Bedingte Wahrscheinlichkeit

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i} \tag{8}$$

- Produktregel

$$p(X = x_i, Y = y_j) = p(Y = y_j | X = x_i) p(X = x_i) \tag{9}$$

Kompakte Schreibweise

- Unterschied zwischen Zufallsvariable B und Belegung, z.B. r
- Wahrscheinlichkeit, B hat Wert r ist $p(B = r)$.
- Kurznotation
  - Verteilung einer Zufallsvariable $p(B)$
  - Wahrscheinlichkeit einer Belegung $p(B = r) = p(r)$

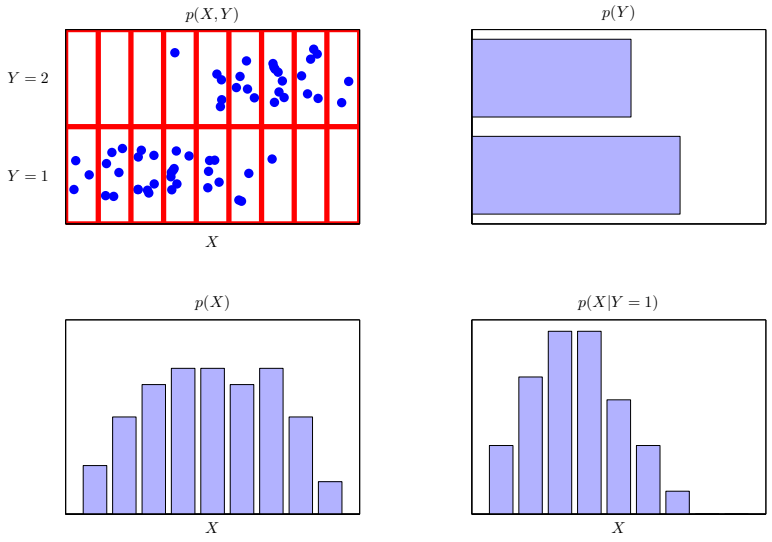
- Wahrscheinlichkeitsregeln
  - Summenregel
$$p(X) = \sum_Y p(X, Y) \tag{10}$$
  - Produktregel
$$p(X, Y) = p(Y|X)p(X) \tag{11}$$

Satz von Bayes

- Anwenden der Produktregel auf die Symmetrie $p(X, Y) = p(Y, X)$
$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \tag{12}$$
- Anwenden der Summenregel auf Nenner
$$p(X) = \sum_Y p(X|Y)p(Y) \tag{13}$$
  - Nenner in Bayesschen Regel eine Art Normalisierungskonstante

Beispiel für bedingte Wahrscheinlichkeiten

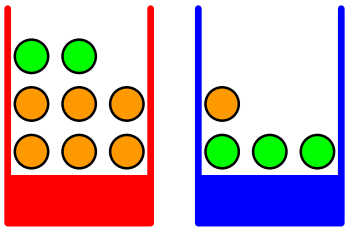
- Histogramme sind einfache Schätzer für Wahrscheinlichkeiten
- Gleichverteilungsannahme innerhalb eines Intervalls



Früchtebeispiel 1/2

- Wahrscheinlichkeit für Kisten
  - $p(B = r) = 4/10$
  - $p(B = b) = 6/10$
- Wahrscheinlichkeit für Früchte
  - $p(F = a|B = r) = 1/4$

- $p(F = o|B = r) = 3/4$
- $p(F = a|B = b) = 3/4$
- $p(F = o|B = b) = 1/4$



Früchtebeispiel

- Wahrscheinlichkeit für Apfel
  - Summen und Produktregel

$$\begin{aligned} p(F = a) &= p(F = a|B = r) \cdot p(B = r) + \\ &\quad p(F = a|B = b) \cdot p(B = b) \\ &= 1/4 \cdot 4/10 + 3/4 \cdot 6/10 \\ &= 11/20 \end{aligned}$$

  - Wahrscheinlichkeit für Orange $p(F = o) = 1 - 11/20 = 9/20$
- Wahrscheinlichkeit für rote Kiste, wenn Orange gezogen

$$\begin{aligned} p(B = r|F = o) &= \frac{p(F = o|B = r)p(B = r)}{p(F = o)} \\ &= 3/4 \cdot 4/10 \cdot 20/9 \\ &= 2/3 \end{aligned}$$

  - ... für blaue Kiste $p(B = b|F = o) = 1 - 2/3 = 1/3$

Interpretation der Bayesschen Regel

- Frage: Welche Kiste wurde gewählt?
  - Antwort: basierend auf $p(B)$
  - Prior-Wahrscheinlichkeit
- Antwort, nachdem Information über Frucht verfügbar
  - basiert auf $p(B|F)$
  - Posterior-Wahrscheinlichkeit

Unabhängigkeit

- Wenn Verbundwahrscheinlichkeit $p(X, Y) = p(X) \cdot p(Y)$ faktorisiert, dann X und Y unabhängig
- Produktregel ergibt für unabhängige Zufallsvariablen
  - $p(Y|X) = p(Y)$
- Früchtebeispiel
  - Falls beide Kisten gleiche Anteile an Äpfeln und Orangen enthalten, dann $p(F|B) = p(F)$

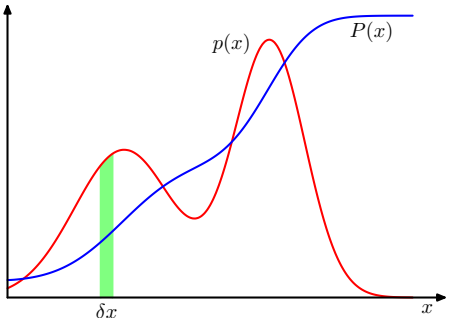
3.2 Wahrscheinlichkeitsdichte

Wahrscheinlichkeitsdichte

- Erweiterung Wahrscheinlichkeit von diskreten Ereignissen auf kontinuierliche Variablen
- Wahrscheinlichkeit, dass kontinuierliche Variable x
  - Wert im Intervall $(x, x + \delta x)$ annimmt,
  - ist $p(x)\delta x$ für $\delta x \to 0$.
  - $p(x)$ ist Wahrscheinlichkeitsdichte
- Allgemeines Intervall (a, b)

$$p(x \in (a, b)) = \int_a^b p(x)dx \tag{14}$$

- Geforderte Eigenschaften
  - $p(x) \geq 0$
  - $\int_{-\infty}^{\infty} p(x)dx = 1$



Variablentransformation

- Durch $x = g(y)$ wird $f(x)$ zu $\tilde{f}(y) = f(g(y))$.
- Sei $p_y(y)$ aus $p_x(x)$ durch Variablentransformation entstanden
  - Beobachtungen in Intervall $(x, x + \delta x)$ werden zu $(y, y + \delta y)$ (bei kleinen δx)
  - Daher gilt $p_x(x)\delta x \simeq p_y(y)\delta y$

$$\begin{aligned} p_y(y) &= p_x(x)\left|\frac{dx}{dy}\right| \\ &= p_x(g(y))|g'(y)| \end{aligned}$$

- Beachte die Folgerung
  - Maximum einer Wahrscheinlichkeitsdichte hängt von der Wahl der Variable ab.

Verschiedene Erweiterungen

- Kumulative Verteilungsfunktion

$$P(z) = \int_{-\infty}^{z} p(x)dx$$

  mit $P'(x) = p(x)$.

- Mehrdimensional
  - Verbundwahrscheinlichkeit $p(\vec{x}) = p(x_1, \ldots, x_D)$ mit
    * $p(\vec{x}) \geq 0$
    * $\int_{-\infty}^{\infty} p(\vec{x}) d\vec{x} = 1$
- Summen-, Produkt und Bayes-Regel

$$\begin{aligned} p(x) &= \int p(x,y) dy \\ p(x,y) &= p(y|x)p(x) \\ p(y|x) &= \frac{p(x|y)p(y)}{\int p(x,y) dy} \end{aligned}$$

3.3 Erwartungswerte und Kovarianzen

Erwartungswert 1/2

- Gewichteter Durchschnitt einer Funktion $f(x)$
- Erwartungswert

$$\mathbb{E}[f] = \sum_x p(x) f(x) \tag{15}$$

$$\mathbb{E}[f] = \int p(x) f(x) dx \tag{16}$$

- Annäherung bei N Beobachtungen

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n) \tag{17}$$

Erwartungswert 2/2

- Funktion mit mehreren Variablen

$$\mathbb{E}_x[f(x,y)] \tag{18}$$

  - x ist Variable, über die gemittelt wird
  - $\mathbb{E}_x[f(x,y)]$ ist eine Funktion in y
- Bedingter Erwartungswert

$$\mathbb{E}_x[f|y] = \sum_x p(x|y) f(x) \tag{19}$$

Varianz

- Maß für die Variabilität um den Mittelwert
- Definiert als

$$var[f] = \mathbb{E}\big[(f(x) - \mathbb{E}[f(x)])^2\big] \tag{20}$$

- Umgestellt als

$$var[f] = \mathbb{E}\big[(f(x)^2\big] - \mathbb{E}\big[f(x)\big]^2 \tag{21}$$

Kovarianz

- Beziehung zwischen zwei Zufallsvariablen x und y

$$\begin{aligned} cov[x,y] &= \mathbb{E}_{x,y}[\{x-\mathbb{E}[x]\}\{y-\mathbb{E}[y]\}] \\ &= \mathbb{E}_{x,y}[xy]-\mathbb{E}[x]\mathbb{E}[y] \end{aligned} \tag{22}$$

- Mehrdimensionale Zufallsvektoren $\vec{x}$ und $\vec{y}$

$$\begin{aligned} cov[\vec{x},\vec{y}] &= \mathbb{E}_{\vec{x},\vec{y}}[\{\vec{x}-\mathbb{E}[\vec{x}]\}\{\vec{y}^T-\mathbb{E}[\vec{y}^T]\}] \\ &= \mathbb{E}_{\vec{x},\vec{y}}[\vec{x}\vec{y}^T]-\mathbb{E}[\vec{x}]\mathbb{E}[\vec{y}] \end{aligned} \tag{23}$$

- $cov[\vec{x}] = cov[\vec{x},\vec{x}]$

3.4 Bayessche Wahrscheinlichkeiten

Bayessche Wahrscheinlichkeiten

- Bisher
  - Wahrscheinlichkeit als Häufigkeit
  - Wiederholbare Ereignisse
- Bayessche Interpretation
  - Wahrscheinlichkeit als Maß für Unsicherheit
  - Auch nicht-wiederholbare Ereignisse
- Viele Axiomsysteme zur Quantisierung von Unsicherheit führen zu Größen, die den Regeln für Wahrscheinlichkeiten gehorchen.
- Größen als (Bayessche) Wahrscheinlichkeiten bezeichnet
- Data Mining
  - Unsicherheit bei der Wahl der Modellparameter berücksichtigt

Beispiel, Kurvenanpassung

- Unsicherheiten über die Parameter $\vec{w}$ durch Verteilung $p(\vec{w})$ erfaßt
- Effekte der Daten $\mathcal{D} = \{t_1, \ldots, t_N\}$ durch $p(\mathcal{D}|\vec{w})$ ausgedrückt
- Bayessche Regel

$$p(\vec{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\vec{w})p(\vec{w})}{p(\mathcal{D})} \tag{24}$$

  Unsicherheit über $\vec{w}$ nach Beobachtung der Daten $\mathcal{D}$

- Bayessche Regel in Worten

$$posterior \propto likelihood \times prior \tag{25}$$

- Nenner in Bayesscher Regel

$$p(\mathcal{D}) = \int p(\mathcal{D}|\vec{w})p(\vec{w})d\vec{w} \tag{26}$$

Diskussion

- Häufigkeitsinterpretation
  - Modellparametern $\vec{w}$ sind feste Werte
  - Fehler und Abweichungen werden über Verteilung von mehreren Datenmengen geschätzt
  - Beispiel: Maximum Likelihood und Bootstrap
- Bayessche Interpretation
  - Nur eine Datenmenge
  - Unsicherheit als Verteilung über Parameter $\vec{w}$
  - Beispiel: Prior-Verteilung über $\vec{w}$
- Beispiel: Münzwurf
- Kritik an Bayesscher Interpretation
  - Wahl des Prior nur nach mathematischer Bequemlichkeit
  - kein Hintergrundwissen

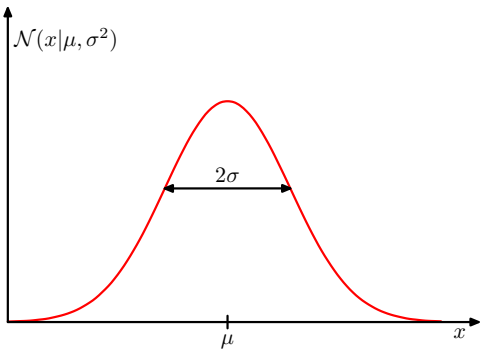
3.5 Gauß-Verteilung

Gauss-Verteilung

- Normal- oder Gauß-Verteilung
  - eine der wichtigsten Verteilungen
  - für kontinuierliche Variablen
- Eindimensional

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\} \tag{27}$$

- Eigenschaften
  - $\mathcal{N}(x|\mu,\sigma^2) > 0$
  - $\int \mathcal{N}(x|\mu,\sigma^2)dx = 1$



Eigenschaften

- Erwartungswert

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) dx = \mu \tag{28}$$

- Moment zweiter Ordnung

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) x^2 dx = \mu^2 + \sigma^2 \tag{29}$$

- Varianz (folgt aus den ersten beiden Gleichungen)

$$var[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2 \tag{30}$$

Schätzer

- Gegeben: N Beobachtungen $\vec{x} = (x_1, \ldots, x_N)^T$
  - Annahme: unabhängig und identisch verteilt (i.i.d.)
- Likelihood der Beobachtungen

$$p(\vec{x}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu, \sigma^2) \tag{31}$$

- Log-Likelihood

$$\ln p(\vec{x}|\mu, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi) \tag{32}$$

- Maximieren bezüglich μ und σ^2

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^{N} x_n, \qquad \sigma^2_{ML} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{ML})^2 \tag{33}$$

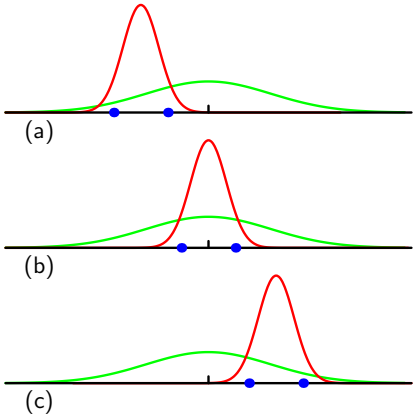
- Eigentlich Verbundoptimierung, aber bei Normalverteilung sind die Gleichungen für μ und σ entkoppelt.

Verzerrung (Bias)

- Schätzer μ und σ^2 sind Funktionen der Datenmenge $(x_1, \ldots, x_N)^T$
- Erwartungswerte für die Schätzer

$$\mathbb{E}[\mu_{ML}] = \mu, \qquad \mathbb{E}[\sigma^2_{ML}] = \frac{N-1}{N} \sigma^2 \tag{34}$$

- Varianz systematisch unterschätzt
  - grün: wahre Verteilung, rot: ML-Schätzung

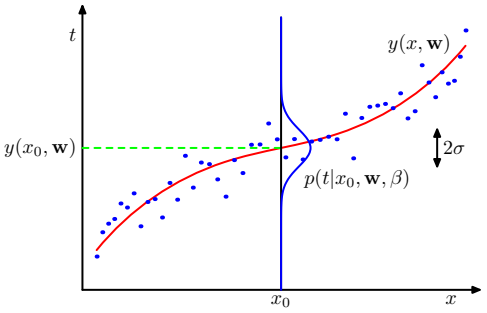


3.6 Nochmal Kurvenanpassung

Kurvenanpassung aus Wahrscheinlichkeitssicht

- Nochmal Kurvenanpassung
  - Diesmal mit Verteilungsannahmen
  - Fehlerfunktion und Regulierung ergeben sich als Konsequenz
- Erinnerung
  - N Beobachtungen, $\vec{x} = (x_1, \ldots, x_N)^T$, $\vec{t} = (t_1, \ldots, t_N)^T$
- Verteilungsannahme
  - Ausgabe t ist normalverteilt verrauscht mit Mittelwert $y(x, \vec{w})$ und Genauigkeit β^{-1}.

$$p(t|x, \vec{w}, \beta) = \mathcal{N}(t|y(x, \vec{w}), \beta^{-1}) \tag{35}$$



Maximum Likelihood

- Likelihood für i.i.d. Beobachtungen

$$p(\vec{t}|\vec{x}, \vec{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \vec{w}), \beta^{-1}) \tag{36}$$

- Maximierung der Log-Likelihood

$$\ln p(\vec{t}|\vec{x}, \vec{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \vec{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

  - Äquivalent zu Mininmierung der Negativen Log-Likelihood

  - Ist bis auf Konstanten die alte Fehlerfunktion

- ML-Schätzer für β

$$\frac{1}{\beta} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \vec{w}) - t_n\}^2 \tag{37}$$

- Vorhersagende Verteilung

$$p(t|x, \vec{w}_{ML}, \beta_{ML}) = \mathcal{N}(t|y(x, \vec{w}_{ML}), \beta_{ML}^{-1}) \tag{38}$$

Regulierung

- Prior-Verteilung für Polynom-Koeffizienten

$$p(\vec{w}|\alpha) = \mathcal{N}(\vec{w}|\vec{0}, \alpha^{-1}\vec{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\vec{w}^T\vec{w}\right\} \tag{39}$$

- Posterior

$$p(\vec{w}|\vec{x}, \vec{t}, \alpha, \beta) \propto p(\vec{t}|\vec{x}, \vec{w}, \beta)p(\vec{w}|\alpha) \tag{40}$$

- Maximierung negativer Log. der Posterior

$$\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \vec{w}) - t_n\}^2 + \frac{\alpha}{2}\vec{w}^T\vec{w} \tag{41}$$

  - Entspricht regulierter Fehlerfunktion mit $\lambda = \alpha/\beta$

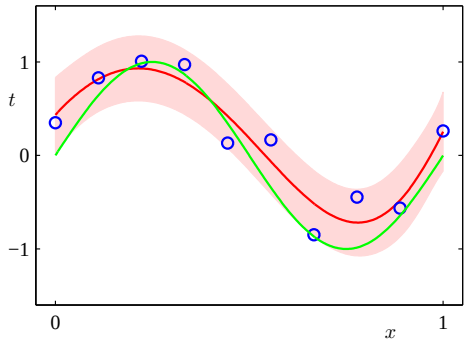
Bayesscher Ansatz

- Keine Punktschätzungen wie bisher
- Vorhersage-Wahrscheinlichkeit integriert über alle möglichen Parameterwerte

$$p(t|x, \vec{x}, \vec{t}) = \int p(t|x, \vec{w})p(w|\vec{x}, \vec{t})d\vec{w} \tag{42}$$

  - Läßt sich geschlossen integrieren
  - Ergibt Normalverteilung

- $M = 9, \alpha = 5 \cdot 10^{-3}, \beta = 11.1$
- Rote Region ist plus/minus 1 Standardabweichung



4 Wahrscheinlichkeitsverteilungen

Wahrscheinlichkeitsverteilungen

- Verteilungen sind
  - Einfache Modelle für Daten
  - Bausteine für komplexe Modelle
- Beispiele
  - Gauß- oder Normalverteilung für kontinuierliche Daten
  - Bernoulli-Verteilung für binäre Daten
  - Binomial und Multinomial-Verteilungen für diskrete Daten
- Schlüsselkonzepte für Bayessche Inferenz

Dichteschätzung

- Problem
  - Modelliere Wahrscheinlichkeitsverteilung $p(\vec{x})$ einer Zufallsvariable $\vec{x}$ für gegebene Beobachtungen $\vec{x}_1, \ldots, \vec{x}_N$
- Problem ist fundamental unterbestimmt, d.h. mehrdeutig
  - Von endlicher Anzahl Stützstellen soll auf Funktion mit unendlich vielen Eingaben geschlossen werden
  - Alle Verteilungen mit $p(\vec{x}_n) > 0$ und $n = 1, \ldots, N$ sind potentielle Kandidaten
- Auswahl der Verteilung
  - Wahl der Modellklasse
  - Wahl der Modellkomplexität

Überblick

- Parametrische Verteilungen
  - Bestimmt durch eine kleine Zahl von Parametern
  - Z.B. Mittelwert μ und Varianz σ^2 einer Gaußverteilung
- Beispiele für Verteilungen
  - Gauß- oder Normalverteilung
  - Bernoulli-Verteilung
  - Binomial und Multinomial-Verteilungen
- Bestimmung der Parameter
  - Häufigkeitsinterpretation $\rightarrow$ Optimierungsproblem
  - Bayessche Interpretation $\rightarrow$ Posterior-Verteilung der Parameter
- Konjugierte Prior-Verteilungen
  - Vereinfacht Bayessche Analyse, da Posterior dieselbe funktionale Form wie Prior annimmt
- Nicht-Parametrische Dichteschätzung

4.1 Binäre Variablen

Binäre Variablen

- Binäre Zufallsvariable $x \in \{0, 1\}$
- Beispiele
  - Münzwurf
  - Entscheidungen
- Wahrscheinlichkeit, daß $x = 1$ ist Parameter μ, d.h.

$$p(x = 1|\mu) = \mu \text{ mit } 0 \leq \mu \leq 1 \tag{43}$$

  $\Rightarrow p(x = 1|\mu) = 1 - \mu$

- Bernoulli-Verteilung

$$\text{Bern}(x|\mu) = \mu^x (1 - \mu)^{1-x} \tag{44}$$

- Erwartungswert und Varianz

$$\mathbb{E}[x] = \mu \tag{45}$$
$$\text{var}[x] = \mu(1 - \mu) \tag{46}$$

Schätzer für Bernoulli-Verteilung

- Gegebene i.i.d. Beobachtungen $\mathcal{D} = \{x_1, \ldots, x_N\}$ von x
- Likelihood

$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \prod_{n=1}^{N} \mu^{x_n} (1 - \mu)^{1-x_n} \tag{47}$$

- Häufigkeitsinterpretation: Maximierung der Log-Likelihood

$$\ln p(\mathcal{D}|\mu) = \sum_{n=1}^{N} \ln p(x_n|\mu) = \sum_{n=1}^{N} \{x_n \ln \mu + (1 - x_n) \ln(1 - \mu)\} \tag{48}$$

- ML-Schätzer

$$\mu_{\text{ML}} = \frac{1}{N} \sum_{n=1}^{N} x_n = \frac{m}{N} \tag{49}$$

  mit $m = \sum_{n=1}^{N} x_n$ ist Anzahl der Einsen in $\mathcal{D}$ (sufficient statistics).

Over-fitting Problem

- Wenige Beobachtungen vorhanden
  - ML-Schätzer kann Extremwerte für μ schätzen
  - Z.B. $N = m = 3 \Rightarrow \mu_{\text{ML}} = 1$
- Ergebnis widerspricht gesundem Menschenverstand
- Vermeiden durch Einbeziehen eines Priors

Bionomial-Verteilung

- Wahrscheinlichkeit, dass bei N unabhängigen Bernoulli-Versuchen m Einsen rauskommen
  - proportional zu $\mu^m(1-\mu)^{N-m}$, siehe Gleichung (47)
- Normalisierungskonstante
  - Anzahl der verschiedenen Möglichkeiten mit N Versuchen m Einsen zu würfeln ist $\binom{N}{m}$
- Bionomial-Verteilung

$$\text{Bin}(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m} \tag{50}$$

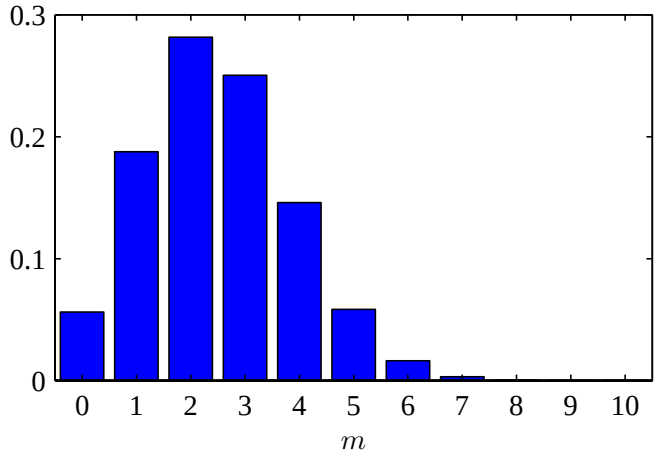
- Erwartungswert und Varianz
  - Herleitung über N unabhängigen Bernoulli-Versuchen

$$\mathbb{E}[m] \equiv \sum_{m=0}^{N} m\text{Bin}(m|N,\mu) = N\mu \tag{51}$$

$$\text{var}[m] \equiv \sum_{m=0}^{N} (m-\mathbb{E}[m])^2\text{Bin}(m|N,\mu) = N\mu(1-\mu) \tag{52}$$

Beispiel für Bionomial-Verteilung

- Histogramm für verschiedene Werte für m
- $N = 10, \mu = 0.25$



Wahl eines Priors für Bernoulli-Verteilung

- Vermeide Overfitting beim ML-Schätzer für Bernoulli
  - Ziel: wähle Prior für μ mit kleinem $p(\mu)$ für Extremwerte
- Motivation
  - Likelihood hat Form $\mu^x(1-\mu)^{1-x}$
  - Wenn Prior $\propto$ Potenzen von μ und $(1-\mu)$, dann hat Posterior dieselbe funktionale Form wie Prior.

  - Konjugierter Prior

- Beta-Verteilung

$$\text{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \tag{53}$$

- Gamma-Funktion $\Gamma(x)$ ist kontinuierliche Verallgemeinerung der Fakultät
  - $\Gamma(x) \equiv \int_0^\infty u^{x-1}e^{-u}du$
  - $\Gamma(x+1) = x\Gamma(x)$, $\Gamma(1) = 1$, $\Gamma(x+1) = x!, x \in \mathbb{N}$
- $\int_0^1 \text{Beta}(\mu|a,b)d\mu = 1$

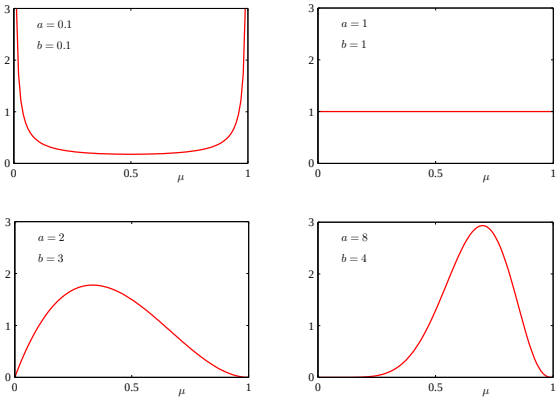
Beta-Verteilung

- Erwartungswert und Varianz

$$\mathbb{E}[\mu] = \frac{a}{a+b} \tag{54}$$

$$\text{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)} \tag{55}$$

- Hyperparameter a und b

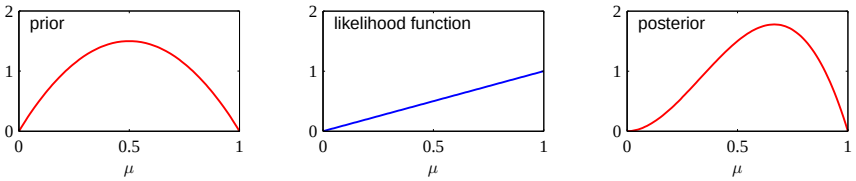


Posterior-Verteilung

- Allgemein: *posterior* $\propto$ *likelihood* $\times$ *prior*
- Was passiert für Bernoulli-Likelihood (47) und Beta-Prior (53)?
- Posterior ist auch Beta-Verteilung mit Hyperparameter $m + a$ und $l + b$
- Interpretation der Hyperparameter
  - Pseudo-Beobachtungen
  - Müssen keine ganzen Integer sein

Sequentieller Schätzer

- Posterior-Verteilung kann als Prior fungieren, wenn neue Beobachtungen kommen
- Beispiel
  - Beobachtungen $x_1, \ldots, x_N$ kommen nach und nach
- Neue Posterior ist Likelihood der neuen Daten mal alte Posterior



$a = 2, b = 2, N = m = 1$

- Anwendungen
  - Real-time Learning
  - Strom-Verarbeitung
  - Große Datenmengen

Vorhersagen

- Ziel
  - Sage Ergebnis der nächsten Beobachtung voraus

$$p(x = 1|\mathcal{D}) = \int_0^1 p(x = 1, \mu|\mathcal{D})d\mu$$
$$= \int_0^1 p(x = 1|\mu)p(\mu|\mathcal{D})d\mu = \int_0^1 \mu p(\mu|\mathcal{D})d\mu = \mathbb{E}[\mu|\mathcal{D}] \quad (56)$$

- In bisherigen Beispiel
$$p(x = 1|\mathcal{D}) = \frac{m + a}{m + a + l + b} \quad (57)$$
- Für $m, l \to \infty$ die Vorhersage wird zur ML-Schätzung
- Für endliche Daten liegt der Posterior-Durchschnitt für μ zwischen dem Durchschnitt des Priors und der Likelihood

Bayessche Eigenschaften von Erwartungswert und Varianz

- Beobachtung
  - Mit zunehmender Anzahl der Beobachtungen wird Varianz kleiner
- Für Beispiel, (55) geht gegen 0 für $a \to \infty$ oder $b \to \infty$
- Allgemein
  - Parameter $\vec{\theta}$, Daten $\mathcal{D}$, beschrieben durch $p(\vec{\theta}, \mathcal{D})$

$$\mathbb{E}_{\vec{\theta}}[\vec{\theta}] = \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\vec{\theta}}[\vec{\theta}|\mathcal{D}]\right] \quad (58)$$

mit $\mathbb{E}_{\vec{\theta}}[\vec{\theta}] \equiv \int \theta p(\vec{\theta})d\theta$ und $\mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\vec{\theta}}[\vec{\theta}|\mathcal{D}]\right] \equiv \int \left\{\int \theta p(\vec{\theta}|\mathcal{D})d\theta\right\} p(\mathcal{D})d\mathcal{D}$

- Analog für Varianz

$$\mathrm{var}_{\vec{\theta}}[\vec{\theta}] = \mathbb{E}_{\mathcal{D}}[\mathrm{var}_{\vec{\theta}}[\vec{\theta}|\mathcal{D}]] + \mathrm{var}_{\mathcal{D}}[\mathbb{E}_{\vec{\theta}}[\vec{\theta}|\mathcal{D}]] \tag{59}$$

- Fazit
  - Posterior-Varianz ist im Durchschnitt kleiner als Prior-Varianz, bei speziellen Daten kann es Ausnahmen geben

4.2 Multinomiale Variablen

Multinomiale Variablen

- Verallgemeinerung von Bernoulli auf mehrwertige Ergebnisse
  - Bernoulli-Variable $x \in \{0, 1\}$
  - Multinomial-Verteilte Variable $x \in \{1, \ldots, K\}$
- 1-aus-K-Schema
  - Statt Integer, Bitvektor, $\vec{x} \in \{0, 1\}^K$ mit $\sum_{k=1}^{K} x_k = 1$
  - Beispiel: $K = 6$, $\vec{x} = (0, 0, 1, 0, 0, 0)^T$
  - Jedem möglichem Wert (Vektor) wird eine Wahrscheinlichkeit μ_k zu geordnet, mit $\sum_{k=1}^{K} \mu_k = 1$

$$p(\vec{x}|\vec{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \tag{60}$$

  mit $\vec{\mu} = (\mu_1, \ldots, \mu_K)^T$

Likelihood

- Daten $\mathcal{D} = \{\vec{x}_1, \ldots, \vec{x}_N\}$ iid. Beobachtungen
- Likelihood

$$p(\mathcal{D}|\vec{\mu}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mu_k^{x_{nk}} = \prod_{k=1}^{K} \mu_k^{\sum_{n=1}^{N} x_{nk}} = \prod_{k=1}^{K} \mu_k^{m_k} \tag{61}$$

  mit $m_k = \sum_{n=1}^{N} x_{nk}$ (sufficient statistics)
- ML-Schätzer

$$\mu_k^{\mathrm{ML}} = \frac{m_k}{N} \tag{62}$$

- Herleitung nutzt Lagrange-Multiplikatoren

Multinomialverteilung

- Wahrscheinlichkeit für eine bestimmte Kombination $m_1, \ldots, m_K$ mit $\sum_{k=1}^{K} m_k = N$

$$\mathrm{Mult}(m_1, \ldots, m_K|\vec{\mu}, N) = \binom{N}{m_1 m_2 \ldots m_K} \prod_{k=1}^{K} \mu^{m_k} \tag{63}$$

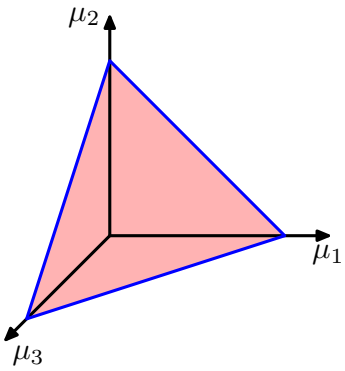
  mit $\binom{N}{m_1 m_2 \ldots m_K} = \frac{N!}{m_1! m_2! \ldots m_K!}$
- Ist Likelihood für Beobachtung der Kombination $m_1, \ldots, m_K$

Dirichlet Verteilung 1/3

- Konjugierter Prior für Multinomial-Verteilung
- Vergleich mit Form von (63)

$$p(\vec{\mu}|\vec{\alpha}) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \tag{64}$$

mit $0 \leq \mu_k \leq 1$ und $\sum_{k=1}^{K} \mu_k = 1$



$K-1$-dimensionaler Simplex mit $K = 3$

- Parametervektor $\vec{\alpha} = (\alpha_1, \ldots, \alpha_K)^T$

Dirichlet Verteilung 2/3

- Normalisierte Verteilung

$$\mathrm{Dir}(\vec{\mu}|\vec{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\ldots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \tag{65}$$

mit $\alpha_0 = \sum_{k=1}^{K} \alpha_k$

- Posterior für Parameter $\{\mu_k\}$ mit Beobachtungen $\{m_k\}$

$$p(\vec{\mu}|\mathcal{D}, \vec{\alpha}) \propto p(\mathcal{D}|\vec{\mu}) p(\vec{\mu}|\vec{\alpha}) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k + m_k - 1} \tag{66}$$

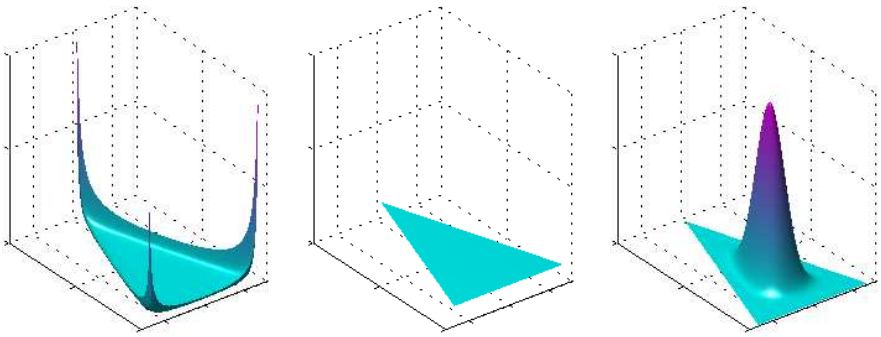
- Posterior ist Dirichlet-Verteilung
  - Normalisierungskonstante durch Vergleich

$$\begin{aligned} p(\vec{\mu}|\mathcal{D}, \vec{\alpha}) &= \mathrm{Dir}(\vec{\mu}|\vec{\alpha} + \vec{m}) \\ &= \frac{\Gamma(\alpha_0 + N)}{\Gamma(\alpha_1 + m_1)\ldots\Gamma(\alpha_K + m_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k + m_k - 1} \end{aligned} \tag{67}$$

mit $\vec{m} = (m_1, \ldots, m_K)^T$

Dirichlet Verteilung 3/3

- Wie bei Beta-Verteilung können die α_k als Pseudo-Beobachtungen interpretiert werden
- Beispiele für Dirichlet-Verteilungen



$\{\alpha_k\} = 0.1$ $\{\alpha_k\} = 1$ $\{\alpha_k\} = 10$

Anwendung Text-Mining

- Bernoulli- und Multinomial-Verteilung mit ihren Prior-Verteilung Beta- und Dirichlet-Verteilung sind wichtige Verteilungen für Text-Mining
- Texte als Menge von Worten repräsentieren (Bag-of-Words)
- Einfachstes Modell: Unigram-Modell
  - Multinomial-Verteilung über dem Vokabular
  - Beobachtungen sind Wortanzahlen über eine Menge von Dokumente
  - Dokumente werden in diesem einfachsten Modell nicht unterschieden
- Einfache Anwendung
  - Zwei Sorten Text: Normale Emails und Spam
  - Bestimme für jede Textsorte eine Multinomialverteilung über dem Vokabular
  - Für neue Email bestimme Vorhersagewahrscheinlichkeiten p(neue Email|Normale Emails) und p(neue Email|Spam)
  - Naive Klassifikator

4.3 Gauß-Verteilung

Gauß-Verteilung

- Verteilung für kontinuierliche eindimensionale Variable x

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\} \tag{68}$$

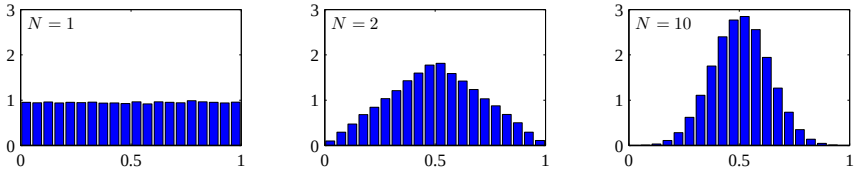
- D-dimensionale Verteilung für Vektor $\vec{x} \in \mathbb{R}^D$

$$\mathcal{N}(\vec{x}|\vec{\mu},\vec{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu}) \right\} \tag{69}$$

  mit $\vec{\mu}$ ist D-dimensionale Vektor und $Sigma$ ist $D \times D$ Kovarianzmatrix

Motivation für Gauß-Verteilung

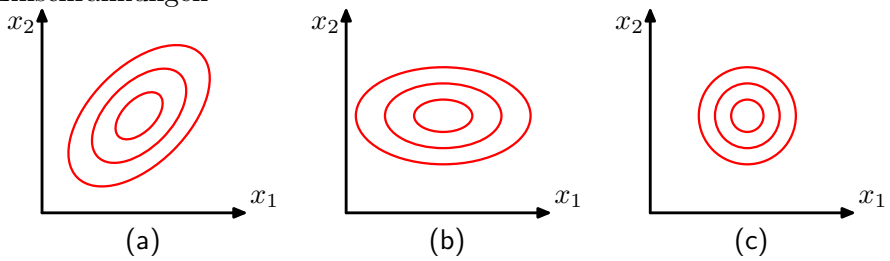
- Gauß-Verteilungen entstehen durch Addition von Zufallsvariablen
  - Zentraler Grenzwertsatz
- Beispiel
  - N gleichverteilte Variablen $x_1, \ldots, x_N$ in $[0, 1]$
  - Verteilung des Durchschnitts $\sum_{n=1}^{N} x_n/N$
  - Für große N verhält sich der Durchschnitt normalverteilt



- Konvergiert sehr schnell
- Spezialfall
  - Bionomial Verteilung ist Summe von N Beobachtungen einer binären Zufallsvariable
  - Wird für große N durch Gauß-Verteilung approximiert

Probleme der Gauß-Verteilung 1/2

- Anzahl der Parameter wächst quadratisch mit Dimension D
  - Kovarianzmatrix $\vec{\Sigma}$ hat $D(D+1)/2$ Parameter
  - Mittelwert $\vec{\mu}$ hat D Parameter
  - Robuste Schätzungen werden unmöglich
  - Invertierung von $\vec{\Sigma}$ sehr aufwendig
- Einschränkungen



a Allgemeine Form für $\vec{\Sigma}$

b Diagonalform $\vec{\Sigma} = \text{diag}(\sigma_i^2)$

c Isotropische Form $\vec{\Sigma} = \sigma^2 \vec{I}$

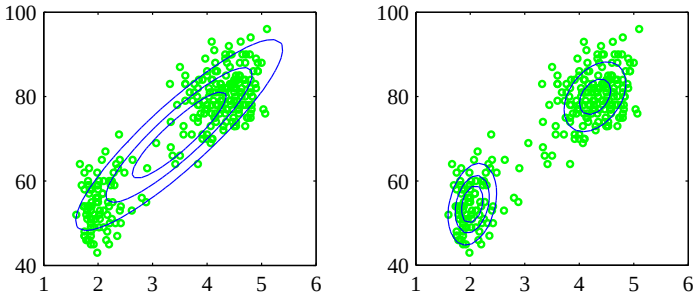
Probleme der Gauß-Verteilung 2/2

- Der Flexibilität der Kovarianzmatrix steht die Beschränkung auf ein Maxima gegenüber
- Viele reale Verteilungen sind multi-modal
- Mischmodelle schaffen hier Abhilfe
  - Einführung von neuen versteckten Variablen
  - Mischmodelle können prinzipiell für alle Arten von Verteilung gebildet werden

4.4 Einführung zu Mischmodellen

Mischmodelle mit Gauß-Verteilungen 1/3

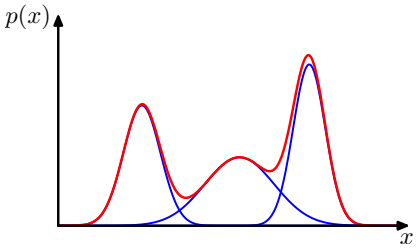
- Reale Daten: Old-Faithful-Geiser
  - Dauer eines Ausbruchs (x-Achse)
  - Abstand bis zum nächsten Ausbruch (y-Achse)



Mischmodelle mit Gauß-Verteilungen 2/3

- Lineare Kombination von Verteilungen

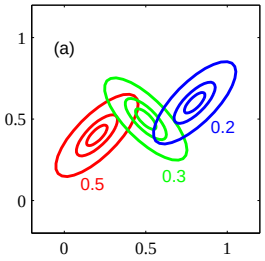
$$p(\vec{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x}|\vec{\mu}_k, \Sigma_k) \tag{70}$$



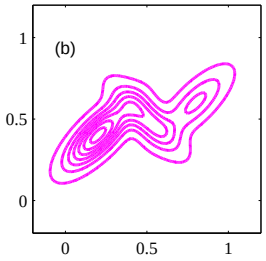
Blau: drei Gauß-Komponenten, Rot: Summe

- Parameter π_k sind Mischungskoeffizienten mit $0 \leq \pi_k \leq 1$ und $\sum_{k=1}^{K} \pi_k = 1$

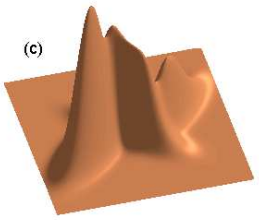
Mischmodelle mit Gauß-Verteilungen 3/3



Erzeugendes Modell



Randverteilung $p(\vec{x})$



Randverteilung $p(\vec{x})$

5 Text Mining, Beispiel Spam

Text-Mining Beispiel

- Gegebene Daten, Dokumente im Bag-of-Words Modell
  - Vokabular $\mathcal{V}$ mit V Wörtern
  - Dokumentmenge $\mathcal{D}$ mit N Dokumenten
  - Dokument ist Multimenge $d_n \subset \mathcal{V}^*$
    - Multimenge heißt, daß Worte mehrfach in der Menge vorkommen können
    - z.B. $d_n = \{blau, blau, rot, gelb\}$
  - Sammlung $\mathcal{W}$ ist Vereinigung aller Dokumente
    - Mehrfach-Elemente bleiben bei der Vereinigung erhalten
    - z.B. $d_1 = \{b, b, g\}, d_2 = \{r, r, g, g, g, g\}$
      $\mathcal{W} = \bigcup_{n=1}^{2} d_n = \{b, b, r, r, g, g, g, g, g\}$
- Unigram-Modelle
  - Mehrdimensionales Bernoulli-Modell
  - Multinomial-Modell
- Anwendung Spam-Erkennung

5.1 Mehrdimensionales Bernoulli-Modell

Mehrdimensionales Bernoulli-Modell

- Unigram-Modell für eine Sammlung $\mathcal{W}$
- Mehrdimensionales Bernoulli-Modell
  - Modelliert das Vorhandensein eines Wortes im Dokument, nicht die Worthäufigkeit
    - Korrespondiert zum Boolschen Modell, Information Retrieval
  - Dokumente als V-dimensionale Bit-Vektoren $\vec{d_n} \in \{0, 1\}^V$
    - Bit v zeigt an, ob $\vec{d_n}$ Wort v enthält
  - Eine Bernoulli-Verteilung pro Wort aus dem Vokabular $\mathcal{V}$
  - Insgesamt V Parameter $\vec{\mu} = (\mu_1, \ldots, \mu_V)^T,\ 0 \leq \mu_v \leq 1,\ 1 \leq v \leq V.$

$$p(v \in d_n | \vec{\mu}) = \mu_v,\ 1 \leq v \leq V \tag{71}$$

- Likelihood (iid. Dokumente, unabhängige Worte)

$$p(\mathcal{D}|\vec{\mu}) = \prod_{n=1}^{N} \prod_{v=1}^{V} \mu_v^{d_{nv}} (1 - \mu_v)^{1 - d_{nv}} = \prod_{v=1}^{V} \mu_v^{m_v} (1 - \mu_v)^{l_v} \tag{72}$$

mit m_v ist Anzahl Dokumente, die v enthalten, $l_v = N - m_v$

Bayessches mehrdimensionales Bernoulli-Modell

- Konjugierte Prior-Verteilung
  - Mehrdimensionale Beta-Verteilung

$$p(\vec{\mu}|\vec{a},\vec{b}) = \prod_{v=1}^{V} \text{Beta}(\mu_v|a_v,b_v) = \prod_{v=1}^{V} \mu_v^{a_v-1}(1-\mu_v)^{b_v-1} \tag{73}$$

- Posterior
- Hyperparameter können als Pseudoanzahlen von Dokumenten interpretiert werden

Beispiel: mehrdimensionales Bernoulli-Modell

- Daten
  - Original: $d_1 = \{b,b,g\}, d_2 = \{r,r,g,g,g,g\}$
  - Transformiert: $\vec{d_1} = (1,0,1)^T, \vec{d_2} = (0,1,1)^T$ mit $b \to v = 1, r \to v = 2, g \to v = 3$
  - Zusammengefaßt: $\vec{m} = (1,1,2)^T, \vec{l} = (1,1,0)^T$
- Hyperparameter (vom Anwender gewählt)
  - $\vec{a} = (1.5,1.5,2)^T, \vec{b} = (1.5,1.5,1)^T$
- Vorhersage für neues Dokument $d = \{b,b,r,r,r,r\}$
  - Transformation $\vec{d} = (1,1,0)^T$
  - Vorhersage
  - Paßt d zu den bisher gesehenen Daten?

5.2 Multinomial-Modell

Multinomial-Modell

- Unigram-Modell für eine Sammlung $\mathcal{W}$
- Multinomial-Modell
  - Berücksichtigt Häufigkeit eines Wortes in Sammlung
  - Sammlung enthält M Worte (mit Mehrfachvorkommen)
  - Häufigkeit eines Wortes v in Sammlung sei m_v mit $\sum_{v=1}^{V} m_v = M$
  - Dokumente werden nicht unterschieden
- Likelihood

$$p(D|\vec{\mu}) = \text{Mult}(\vec{m}|\vec{\mu},M) = \binom{M}{m_1 m_2 \dots m_V} \prod_{v=1}^{V} \mu_v^{m_v} \tag{74}$$

  mit $\vec{\mu} = (\mu_1,\dots,\mu_V)^T$ und $\vec{m} = (m_1,\dots,m_V)^T$

Bayessches Multinomial-Modell

- Konjugierte Prior-Verteilung
  - Dirichlet-Verteilung

$$p(\vec{\mu}|\vec{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\dots\Gamma(\alpha_V)} \prod_{v=1}^{V} \mu_v^{\alpha_v - 1} \tag{75}$$

    mit $\alpha_0 = \sum_{v=1}^{V} \alpha_v$
- Posterior
- Hyperparameter α_v können als Pseudoanzahlen von Worten interpretiert werden

Beispiel: Multinomial-Modell

- Daten
  - Original: $d_1 = \{b, b, g\}, d_2 = \{r, r, g, g, g, g\}$
  - Transformiert: $\mathcal{W} = \{b, b, r, r, g, g, g, g, g\}$
  - Zusammengefaßt: $\vec{m} = (2, 2, 5)^T$ und $M = 9$
- Hyperparameter (vom Anwender gewählt)
  - $\vec{\alpha} = (1.5, 1.5, 2)^T$
- Vorhersage für neues Dokument $d = \{b, b, r, r, r, r\}$
  - Zusammengefaßt: $\vec{m}'_d = (2, 4, 0)^T$ und $M'_d = 6$
  - Vorhersage
  - Paßt d zu den bisher gesehenen Daten?
- Vorhersage-Verteilung ist Dirichlet Compound Multinomial Verteilung (DCM) (auch Polya-Verteilung)[2]

5.3 Anwendung: Spam-Erkennung

Anwendung: Spam-Erkennung

- Gegeben zwei Sammlungen: C_1 mit normalen eMails und C_2 mit Spam
  - $C_1 = \{d_1, d_2\}$, $d_1 = \{b, b, g\}, d_2 = \{r, r, g, g, g, g\}$
  - $C_2 = \{d_3, d_4\}$, $d_3 = \{b, b, b, r, r\}$, $d_4 = \{r, r, r, g\}$
  - Prior-Wahrscheinlichkeiten $p(C_1) = 0.9$, $p(C_2) = 0.1$
- Klassifikation einer neuen eMail d mittels Bayesscher Regel

$$p(C_i|d, \vec{\alpha}_i) = \frac{p(d|C_i, \vec{\alpha}_i)p(C_i)}{\sum_j p(d|C_j, \vec{\alpha}_j)p(C_j)} \tag{76}$$

  mit $i = 1, 2$
- $p(d|C_i, \vec{\alpha}_i)$ ist Vorhersagewahrscheinlichkeit entsprechend dem verwendeten Modell
- Vereinfachend entscheidet der Klassifikator für die Klasse mit der höherer Posterior-Wahrscheinlichkeit
  - An dieser Stelle können Kosten für Entscheidungen und Fehlentscheidungen berücksichtigt werden.

[2] Siehe Madsen, RE., Kauchak, D. and Elkan, C. (2005) Modeling Word Burstiness Using the Dirichlet Distribution. ICML, 545-552, `http://www.cse.ucsd.edu/~dkauchak/kauchak05modeling.pdf`

Evaluation

- Einfache Evaluation
  - Aufteilung der Daten in Training- und Testdaten
  - Bestimmung des Klassifikationsfehlers auf den Testdaten
  - Berechnung der Kosten, z.B. wieviel Falsch-Negative wenn keine Falsch-Positiven erlaubt
- k-fache Kreuzvalidierung
  - Partitioniere Gesamtdaten in k gleiche Teile
  - $k-1$ Teile sind Trainingsdaten und ein Teil ist Testdaten
  - Führe für diese Aufteilung die einfache Evaluation (s.o.) durch
  - Tausche Testdatenteil durch einen Trainingsdatenteil aus, dann einfache Evaluation
  - Jeder Teil ist mal Testdatenteil $\Rightarrow$ k Klassifikationsfehler $\Rightarrow$ Standardabweichung des Klassifikationsfehler
- Bootstrap
  - Wie Kreuzvalidierung, nur die Trainingsdaten werden durch Ziehen mit Zurücklegen bestimmt.
  - Eignet sich für kleine Datensätze
- Tuning der Hyperparameter mittels Validierungsdaten
  - Verschiedene Parametereinstellungen testen und beste Einstellung wählen

Verbesserung der Vorverarbeitung

- Bessere Erkennung von Wortgrenzen, Markov-Random-Fields
- Einführen von einfachen Zusatzattributen
  - Anzahl nicht darstellbarer Zeichen
  - Länge von Sequenzen mit Großbuchstaben
- Beispieldaten
  - Spam Base: `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/spambase/`
  - Apache SpamAssassin Project `http://spamassassin.apache.org`

5.4 Nicht-Konjugierte Prior-Verteilungen

Nicht-Konjugierte Prior-Verteilungen

- Beliebige Prior-Verteilungen über der passenden Domäne sind erlaubt.
- Bisherige Prior-Verteilungen
  - Mehrdimensionale Beta-Verteilung für mehrdimensionale Bernoulli-Verteilung
  - Dirichlet-Verteilung für Multinomial-Verteilung
  - Mehrdimensionale Beta- und Dirichlet-Verteilung nehmen unabhängige Wörter an
- Prior-Verteilung mit Kovarianzen zwischen Wörtern
  - Mehrdimensionale Normal-Verteilung kann Kovarianzen modellieren
  - Aber Domäne paßt nicht

Logistische Normalverteilung 1/3

- Abbildung aus dem $\mathbb{R}^K$ in den $K-1$ Simplex mit logistischer Funktion

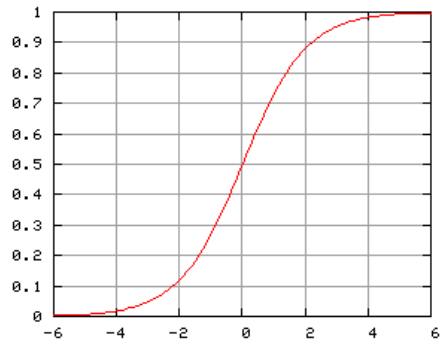
$$\vec{x} \in \mathbb{R}^K u_k = \frac{e^{x_k}}{1 + \sum_{k'=1}^{K} e^{x_{k'}}} \tag{77}$$

- Rücktransformation ist logit-Funktion

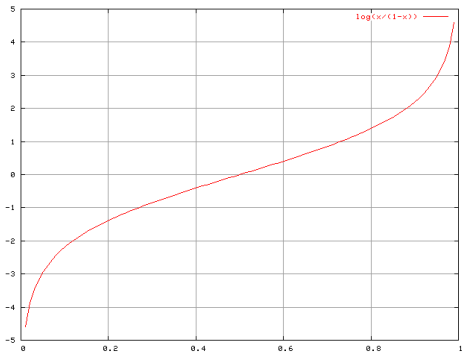
$$\vec{u} \in \mathbb{R}, 0 \leq u_k \leq 1, \sum_{k=1}^{K} u_k = 1, x_k = \ln\left(\frac{u_k}{1 - \sum_{k'=1}^{K} u_{k'}}\right) \tag{78}$$

Logistische Normalverteilung 2/3

Logistische Funktion



Logit-Funktion



Logistische Normalverteilung 3/3

- Logistische Normalverteilung $L(u|\mu, \Sigma)$
- Posterior für Multinomial mit Logistischer Normalverteilung
- Vorteil
  - Kovarianzen zwischen Wörtern werden modelliert
- Nachteil
  - keine normalisierte Wahrscheinlichkeit
  - Keine geschlossene Form bei der Vorhersage, da kein konjugierter Prior
  - Approximationen und Sampling möglich

6 Mischmodelle

Mischmodelle

- Probabilistische Modelle können beobachtbare $\vec{x}$ und versteckte Variablen $\vec{\theta}$ enthalten
- Die Verteilung der beobachtbaren Variablen $\vec{x}$ ist als Randverteilung modelliert

$$p(\vec{x}) = \sum_{\vec{\theta}} p(\vec{x}, \vec{\theta}) = \sum_{\vec{\theta}} p(\vec{x}|\vec{\theta})p(\vec{\theta}) \tag{79}$$

- Einführung von versteckten Variablen erlaubt komplexe Verteilungen aus einfachen Verteilungen zusammenzubauen.
- Mischmodelle entstehen durch das Einführen von diskreten Indikatorvariablen (Auswahl-Bits)
- Einführung
  - K-Means als einfacher nicht-probabilistischer Spezialfall
  - Gauß-Mischmodelle mit Expectation-Maximization (EM) Algorithmus

6.1 K-Means

K-Means Cluster-Analyse

- Gegeben
  - N mehrdimensionale Datenpunkte $\{\vec{x}_1, \ldots, \vec{x}_N\}$
- Problem
  - Partitioniere Daten in K Cluster
  - Cluster sind Teilmengen der Daten
    - Distanz innerhalb ist klein, kleine Intra-Cluster-Distanz
    - Distanz zwischen Punkten aus verschiedenen Clustern ist groß, große Inter-Cluster-Distanz
  - K ist erstmal ein vorgegebener Parameter
- Cluster beschrieben durch Prototyp-Punkt $\vec{\mu}_k,\ k = 1, \ldots, K$
- Ziel:
  - Summe der quadrierten Distanzen der Punkte zu ihrem jeweils nächsten Prototyp minimieren

K-Means Fehlerfunktion

- Zuordnung von Datenpunkten zu Cluster, Eins-aus-K-Kodierung
  - binäre Indikatorvariablen $r_{nk} \in \{0, 1\},\ k = 1, \ldots, K$
  - Punkt $\vec{x}_n$ gehört zu Cluster k, dann $r_{nk} = 1$ und $r_{nj} = 0$ für $k \neq j$
- Fehlerfunktion oder Verzerrungsmaß

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\vec{x}_n - \vec{\mu}_k\|^2 \tag{80}$$

- Ziel
  - Finde Belegung für $\{r_{nk}\}$ und $\{\vec{\mu}_k\}$, so daß J minimal

K-Means Algorithmus 1/2

- Iterative Zwei-Schritt-Optimierung
  1. Minimiere J bezüglich $\{r_{nk}\}$, festes $\{\vec{\mu}_k\}$
  2. Minimiere J bezüglich $\{\vec{\mu}_k\}$, festes $\{r_{nk}\}$
  3. Falls Abbruchkriterium nicht erreicht, gehe zu 1.
- Minimiere bezüglich $\{r_{nk}\}$, E-Schritt
  - J ist in (80) eine lineare Funktion in r_{nk}
  - Terme mit r_{nk} sind unabhängig bezüglich n
    - $\{r_{nk}\}_{k=1,\ldots,K}$ separat optimieren
  - Setze r_{nk} auf eins, wenn $\|\vec{x}_n - \vec{\mu}_k\|^2$ minimal

$$r_{nk} = \begin{cases} 1 & \text{wenn } k = \text{argmin}_j \|\vec{x}_n - \vec{\mu}_j\|^2 \\ 0 & \text{sonst} \end{cases} \tag{81}$$

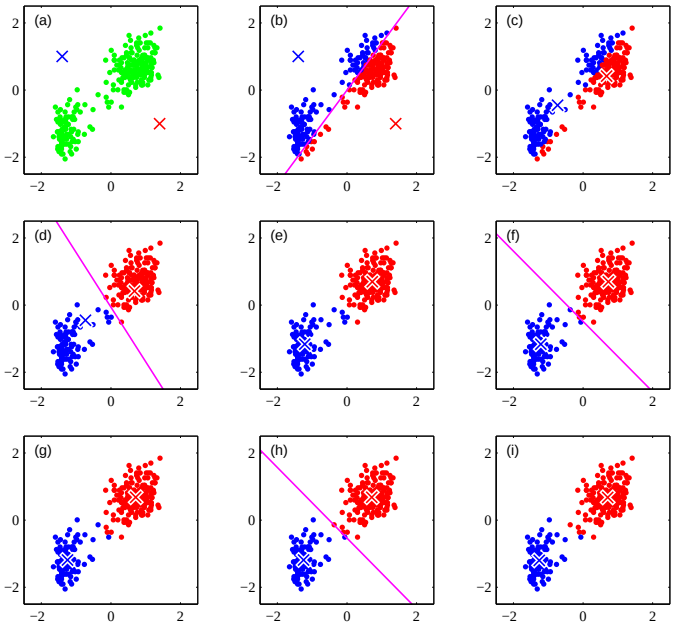
K-Means Algorithmus 2/2

- Minimiere bezüglich $\{\vec{\mu}_k\}$, M-Schritt
  - J ableiten und Null setzen

$$\vec{\mu}_k = \frac{1}{\sum_{n=1}^{N} r_{nk}} \sum_{n=1}^{N} r_{nk}\vec{x} \tag{82}$$

- $\sum_{n=1}^{N} r_{nk}$ ist Anzahl Cluster k zugeordneten Punkte
- $\vec{\mu}_k$ wird im zweiten Schritt auf den Durchschnitt gesetzt
- In jedem Schritt wird J verringert $\Rightarrow$ Konvergenz

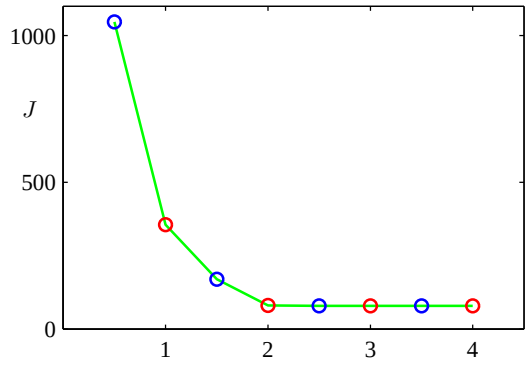
K-Means, Old-Faithful-Daten



a) Initialisierung,
b) erster E-Schritt,
c) anschließender M-Schritt,
d-i) Schritte bis Konvergenz

K-Means Konvergenz

- Fehlerfunktion nach jedem E-Schritt (blau) und M-Schritt (rot)



- Erweiterungen
  - Kombination mit Indexstrukturen (Suchbäumen)
  - Ausnutzen der Dreiecksungleichung
  - Sequentielle on-line Berechnung

K-Means Beispiel-Anwendung

- Verlustbehaftete Bildkompression
- Daten: drei-dimensionale RGB Farbinformation aller Pixel
- K ist Anzahl der Farben im komprimierten Bild
- Prototypen $\{\vec{\mu}_k\}$ sind im Originalfarbraum, Pixel im Bild referenzieren auf zugeordnetes $\vec{\mu}_k$
- Beispiel
  - Original hat 8 Bit Farbinformation pro Farbkanal und Pixel,
  - Originalbild hat $24 \cdot N$ Bit, N ist Anzahl Pixel
  - Komprimiertes Bild
    - Prototypen: $24 \cdot K$ Bit
    - Pixel: $N \log_2 K$ Bit
  - Bild mit Auflösung $240 \times 180 = 43200$ Pixel braucht $24 \cdot 43200 = 1036800$ Bit
  - Komprimierte Version: 43248 Bit ($K = 2$), 86472 Bit ($K = 3$), 173040 Bit ($K = 10$)

K-Means Bildkompression

6.2 Gauß-Mischmodell, Teil 1

Gauß-Mischmodell 1/2

- Motivation für EM-Algorithmus
- Gauß-Mischmodell ist linear-Kombination von Gauß-Verteilungen

$$p(\vec{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x}|\vec{\mu}_k, \vec{\Sigma}_k) \tag{83}$$

- Indikatorvaribale $\vec{z}$
  - Eins-aus-K-Schema
  - $\vec{z} \in \{0,1\}^K$ mit $\sum_{k=1}^{K} z_k = 1$
  - Verteilung spezifiziert als $p(z_k = 1) = \pi_k$ mit $0 \leq \pi_k \leq 1$ und $\sum_{k=1}^{K} \pi_k = 1$
- Wegen Eins-aus-K-Schema, Verteilung schreiben als

$$p(\vec{z}) = \prod_{k=1}^{K} \pi_k^{z_k} \tag{84}$$

Gauß-Mischmodell 2/2

- Bedingte Verteilung für Komponenten

$$p(\vec{x}|z_k = 1) = \mathcal{N}(\vec{x}|\vec{\mu}_k, \vec{\Sigma}_k) \tag{85}$$

- Wegen Eins-aus-K-Schema, Verteilung schreiben als

$$p(\vec{x}|\vec{z}) = \prod_{k=1}^{K} \mathcal{N}(\vec{x}|\vec{\mu}_k, \vec{\Sigma}_k)^{z_k} \tag{86}$$

- Verbundverteilung $p(\vec{x}, \vec{z}) = p(\vec{x}|\vec{z})p(\vec{z})$

- Randverteilung $p(\vec{x})$ durch summieren über $\vec{z}$

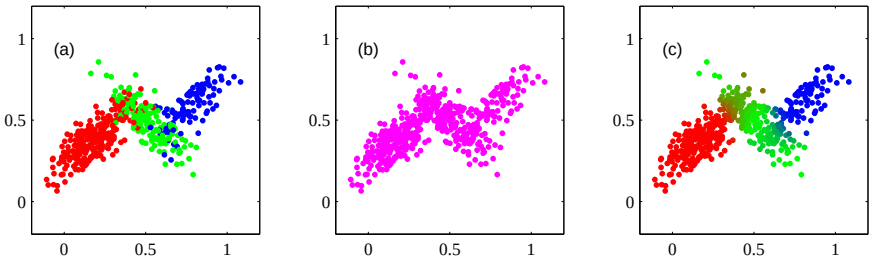
$$p(\vec{x}) = \sum_{\vec{z}} p(\vec{z})p(\vec{x}|\vec{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x}|\vec{\mu}_k, \vec{\Sigma}_k) \tag{87}$$

- Bei N Beobachtungen $\vec{x}_1, \ldots, \vec{x}_N$ gibt es für jede Beobachtung $\vec{x}_n$ eine separate Indikatorvariable $\vec{z}_n$

Beobachtungen ziehen aus Gauß-Mischmodell

- Für gegebene Parameter $\{\pi_k, \vec{\mu}_k, \vec{\Sigma}_k\}$ analog wie Früchteziehen
  - Erst Indikatorvariable ziehen
  - Dann Beobachtung entsprechend gewählter Gauß-Komponente ziehen
- Posterior für gezogene Beobachtung $\vec{x}$:
  - Von welcher Gauß-Komponente wurde $\vec{x}$ gezogen?

$$\gamma(z_k) \equiv p(z_k = 1|\vec{x}) \tag{88}$$



ML-Schätzer für Gauß-Mischmodell

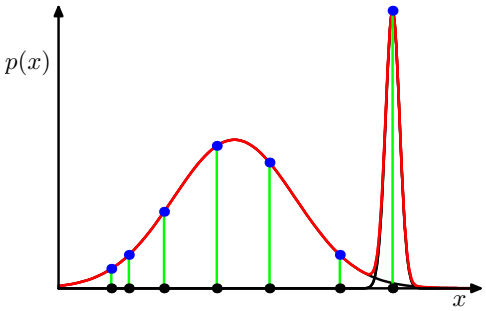
- Gegebene Daten
  - N iid. Beobachtungen, D-dimensionale Datenpunkte, $\{\vec{x}_1, \ldots, \vec{x}_N\}$
  - Repräsentiert als $N \times D$ Matrix $\vec{X}$, n-te Zeile ist $\vec{x}_n^T$
- Indikatorvariablen, versteckt, nicht beobachtet
  - $N \times K$ Matrix $\vec{Z}$, n-te Zeile ist $\vec{z}_n^T$
- Log-Likelihood der Daten

$$\ln p(\vec{X}|\vec{\pi}, \vec{\mu}, \vec{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x}|\vec{\mu}_k, \vec{\Sigma}_k) \right\} \tag{89}$$

Probleme des ML-Schätzer für Gauß-Mischmodell

- Singularitäten
  - Optimierungsproblem ist schlecht gestellt, weil Likelihood gegen ∞ gehen kann
  - Vereinfachung: $\Sigma_k = \sigma_k \vec{I}$, $\vec{I}$ ist Einheitsmatrix
    - Beobachtung gilt auch für allgemeinen Fall
  - Falls eine Gauß-Komponente auf einem Datenpunkt sitzt, $\vec{\mu}_j = \vec{x}_n$, dann kann das Mischmodell kollabieren. Likelihood geht in diesem Fall gegen ∞, wenn σ_j gegen Null geht.

- Singularitäten treten erst bei Mischmodell auf, nicht bei einzelner Gaußverteilung
- Gesucht ist gutartiges lokales Optimum, kein globales Optimum
- Bayesscher Ansatz vermeidet Singularitäten
- Sonst Heuristiken verwenden



Weitere Probleme

- Identifizierbarkeit
  - Für jedes lokale Optimum gibt es $K!$ gleichartige Lösungen
  - Umbenennen der Komponenten
  - Tritt nur auf, wenn Komponenten interpretiert werden
- Maximierung der Log-Likelihood von Mischmodellen ist komplizierter als bei einfachen Verteilungen, weil Summe im Logarithmus auftaucht.
- Ansätze
  - Direkte gradienten-basierte Optimierung
  - Expectation-Maximization (EM)

EM für Gauß-Mischmodelle 1/2

- Herleitung ohne EM-Theorie
- Ableitung der Daten-Likelihood (89) nach $\vec{\mu}_k$ und Null setzen

$$0 = -\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\vec{x}_n|\vec{\mu}_k, \vec{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}_n|\vec{\mu}_j, \vec{\Sigma}_j)} \vec{\Sigma}_k^{-1}(\vec{x}_n - \vec{\mu}_k) \tag{90}$$

- In Gleichung taucht Posterior $\gamma(z_{nk}) \equiv p(z_k = 1|\vec{x}_n) = \frac{\pi_k \mathcal{N}(\vec{x}_n|\vec{\mu}_k, \vec{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}_n|\vec{\mu}_j, \vec{\Sigma}_j)}$ auf.
- Multiplizieren mit $\vec{\Sigma}_k$ ergibt

$$\vec{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \vec{x}_n \tag{91}$$

  mit $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
- N_k ist Anzahl der Punkte in Cluster k
- $\vec{\mu}_k$ ist gewichteter Durchschnitt

EM für Gauß-Mischmodelle 2/2

- Ableitung der Daten-Likelihood (89) nach $\vec{\Sigma}_k$ und Null setzen

$$\vec{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\vec{x}_n - \vec{\mu}_k)(\vec{x}_n - \vec{\mu}_k)^T \tag{92}$$

  - Ähnlich zum ML-Schätzer einer Gauß-Verteilung

- Ableitung der Daten-Likelihood (89) mit Lagrange-Multiplikator $p(\vec{X}|\vec{\pi}, \vec{\mu}, \vec{\Sigma}) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)$ nach π_k und Null setzen

$$0 = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\vec{x}_n|\vec{\mu}_k, \vec{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}_n|\vec{\mu}_j, \vec{\Sigma}_j)} + \lambda \tag{93}$$

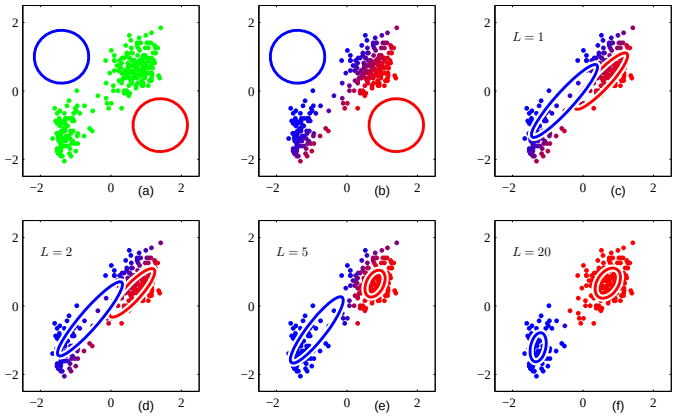
- Ergebnis

$$\pi_k = \frac{N_k}{N} \tag{94}$$

- Keine geschlossene Form, Parameter hängen über $\gamma(z_{nk})$ zusammen.

EM-Algorithmus, Beispiel

- Iteratives Verfahren: Initialisieren, E-Schritt und M-Schritt abwechseln
  - E-Schritt: $\gamma(z_{nk})$ berechnen
  - M-Schritt: $\vec{\pi}, \vec{\mu}, \vec{\Sigma}$ aktualisieren
- Beispiel: Old-Faithful-Daten, $K = 2$, L ist Anzahl Iterationen



Zusammenfassung des Algorithmus

1. Initialisiere $\vec{\pi}, \vec{\mu}$ und $\vec{\Sigma}$ und berechne Startwert der log-Likelihood
2. **E-Schritt** berechne Posteriors mit den aktuellen Parametern

$$\gamma(z_{nk}) \equiv p(z_k = 1|\vec{x}_n) = \frac{\pi_k \mathcal{N}(\vec{x}_n|\vec{\mu}_k, \vec{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}_n|\vec{\mu}_j, \vec{\Sigma}_j)} \tag{95}$$

3. **M-Schritt** Aktualisiere Parameter mit neuen Posteriors

$$\vec{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \vec{x}_n \tag{96}$$

$$\vec{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\vec{x}_n - \vec{\mu}_k)(\vec{x}_n - \vec{\mu}_k)^T \tag{97}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \text{ mit } N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{98}$$

4. Berechne Log-Likelihood, falls nicht konvergiert, gehe zu 2.

$$\ln p(\vec{X}|\vec{\pi}, \vec{\mu}, \vec{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x}|\vec{\mu}_k, \vec{\Sigma}_k) \right\} \tag{99}$$

Diskussion

- EM-Algorithmus braucht viel mehr Iterationen als K-Means und die Iterationen sind berechnungsitensiver
- K-Means wird oft zum Initialisieren des EM benutzt
- Abbruch-Kriterien für Konvergenz
  - K-Means: wenn keine Zuordnung sich mehr ändert
  - Feste, meist kleine Anzahl von Schritten, early stopping
  - Absolute Zuwachs der Likelihood L fällt unten einen Schwellenwert $L - L^{\text{new}} < \theta$
  - Relativer Zuwachs der Likelihood L fällt unten einen Schwellenwert $\frac{L - L^{\text{new}}}{L} < \theta'$
- EM findet nur lokales Maximum
- Maximierung ist nicht alles, Overfitting, Singularitäten

7 Theorie zum EM-Algorithmus

7.1 Allgemeiner EM-Algorithmus

EM-Algorithmus in abstrakter Form

- Versteckte Variablen
  - Schlüsselrolle für EM
  - Bisher nur durch intelligentes Draufsehen berücksichtigt
- Ziel des EM
  - Maximum-Likelihood Schätzung
  - kann auf Maximum-A-Posteriori (MAP) und fehlende Daten erweitert werden
- Notation
  - $\vec{X}$ Datenmatrix, n-te Zeile ist $\vec{x}_n^T$
  - $\vec{Z}$ versteckte Variablen, n-te Zeile is $\vec{z}_n^T$
  - $\vec{\theta}$ alle Parameter
    - z.B. Gauß-Mischmodell $\vec{\theta} = (\vec{\mu}, \vec{\Sigma}, \vec{\pi})$
- Log-Likelihood für die Daten als Randverteilung

$$\ln p(\vec{X}|\vec{\theta}) = \ln\left\{\sum_{\vec{Z}} p(\vec{X}, \vec{Z}|\vec{\theta})\right\} \tag{100}$$

Transformation des Maximierungsproblems 1/2

- Unvollständige Daten-Log-Likelihood (106) ist Funktion von $\vec{\theta}$

$$f(\vec{\theta}) \equiv \ln p(\vec{X}|\vec{\theta}) \tag{101}$$

- Problem
  - Summe innerhalb des Logarithmus läßt sich nicht weiter vereinfachen
  - Keine Formel für ML-Schätzung
- Idee
  - Maximiere anstelle unvollständigen Daten-Log-Likelihood (106) andere Funktion, die maximal wird, wenn unvollständige Daten-Log-Likelihood maximal wird
- Vollständige Daten-Log-Likelihood

$$g(\vec{\theta}, \vec{Z}) \equiv \ln p(\vec{X}, \vec{Z}|\vec{\theta}) \tag{102}$$

Transformation des Maximierungsproblems 2/2

- Problem
  - Berechnung von (102) setzt Kenntnis der versteckten Variablen $\vec{Z}$ voraus
  - Bekannte Information über $\vec{Z}$ ist Posterior $p(\vec{Z}|\vec{X}, \vec{\theta})$
  - Posterior hängt aber wiederum von Parametern $\vec{\theta}$ ab
- Idee: Zwei-Schritt Optimierung nach Initialisierung von $\vec{\theta}$

- E-Schritt: Berechne Posterior-Verteilung von $\vec{Z}$ für aktuelle Parameter $\vec{\theta}^{\text{old}}$
- M-Schritt: Maximiere Erwartungswert von g über Posterior-Verteilung von $\vec{Z} \rightarrow$ neue Parameter $\vec{\theta}^{\text{new}}$

- Bei gegebenen aktuellen Parametern $\vec{\theta}^{\text{old}}$ ist Erwartungswert von g über Posterior-Verteilung von $\vec{Z}$ eine Funktion von $\vec{\theta}$

$$\mathcal{Q}(\vec{\theta}, \vec{\theta}^{\text{old}}) = \mathbb{E}_{\vec{Z}}[g] = \sum_{\vec{Z}} p(\vec{Z}|\vec{X}, \vec{\theta}^{\text{old}}) \ln p(\vec{X}, \vec{Z}|\vec{\theta}) \tag{103}$$

Diskussion

- Transformation des Maximierungproblems von

$$\text{argmax}_{\vec{\theta}} \ln p(\vec{X}|\vec{\theta}) = \text{argmax}_{\vec{\theta}} \ln \left\{ \sum_{\vec{Z}} p(\vec{X}, \vec{Z}|\vec{\theta}) \right\}$$

  nach

$$\vec{\theta}^{\text{new}} = \text{argmax}_{\vec{\theta}}\, \mathcal{Q}(\vec{\theta}, \vec{\theta}^{\text{old}}) = \text{argmax}_{\vec{\theta}} \sum_{\vec{Z}} p(\vec{Z}|\vec{X}, \vec{\theta}^{\text{old}}) \ln p(\vec{X}, \vec{Z}|\vec{\theta})$$

- Gewinn: Logarithmus wird direkt auf $p(\vec{X}, \vec{Z}|\vec{\theta})$ angewendet $\Rightarrow$ idd. Annahme nutzbar und bekanntes $\vec{Z}$ erlaubt Formulierung von Auswahlprodukten
- Offene Frage
  - Führt die Transformation auch wirklich zu einem Maximum in der unvollständigen Daten-Log-Likelihood?
  - Antwort: ja, zu einem lokalen Maximum, Beweis später

Zusammenfassung des Algorithmus

1. Initialisiere Parameter $\vec{\theta}$ mit $\vec{\theta}^{\text{old}}$ und berechne Startwert der unvollständigen Daten-log-Likelihood
2. **E-Schritt** berechne Posteriors $p(\vec{Z}|\vec{X}, \vec{\theta}^{\text{old}})$
3. **M-Schritt** Berechne neue Parameter $\vec{\theta}^{\text{new}}$

$$\vec{\theta}^{\text{new}} = \text{argmax}_{\vec{\theta}}\, \mathcal{Q}(\vec{\theta}, \vec{\theta}^{\text{old}}) \tag{104}$$

   mit

$$\mathcal{Q}(\vec{\theta}, \vec{\theta}^{\text{old}}) = \sum_{\vec{Z}} p(\vec{Z}|\vec{X}, \vec{\theta}^{\text{old}}) \ln p(\vec{X}, \vec{Z}|\vec{\theta}) \tag{105}$$

4. Teste auf Konvergenz der unvollständigen Daten-Log-Likelihood oder der Parameter. Falls nicht konvergiert, $\vec{\theta}^{\text{old}} \leftarrow \vec{\theta}^{\text{new}}$ und gehe zu 2.

Erweiterungen

- Maximierung der Log-Posterior anstelle der Log-Likelihood

$$\ln p(\vec{\theta}|\vec{X}) = \ln \left\{ p(\vec{\theta}) \sum_{\vec{Z}} p(\vec{X}, \vec{Z}|\vec{\theta}) \right\} + c \tag{106}$$

- Fehlende Daten
  - Statt nicht beobachtete Variablen können die versteckten Variablen auch zu Attributen von fehlenden Werten zugeordnet werden
  - Geht nur, wenn das Fehlen der Werte zufällig ist und nicht systematisch

7.2 Gauß-Mischmodell, Teil 2

Gauß-Mischmodell, Teil 2

- In unvollständiger Daten-Log-Likelihood (89) ist die Summe innerhalb der Logarithmus
- Wegen (84) und (86) ist vollständige Daten-Likelihood

$$p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\Sigma}, \vec{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_K^{z_{nk}} \mathcal{N}(\vec{x}_n|\vec{\mu}_k, \vec{\Sigma}_k)^{z_{nk}} \tag{107}$$

- Vollständige Daten-Log-Likelihood
- Posterior

Gauß-Mischmodell, E-Schritt

- Zur Summe über alle Belegungen für $\vec{Z}$ in

$$\mathcal{Q}(\vec{\theta}, \vec{\theta}^{\text{old}}) = \sum_{\vec{Z}} p(\vec{Z}|\vec{X}, \vec{\theta}^{\text{old}}) \ln p(\vec{X}, \vec{Z}|\vec{\theta})$$

  tragen nur Terme mit $z_{nk} = 1$ bei $\Rightarrow$

- Berechne nur Posteriors mit $z_{nk} = 1$

$$\begin{aligned}
\gamma(z_{nk}) &= p(z_{nk} = 1|\vec{x}_n, \vec{\mu}^{\text{old}}, \vec{\Sigma}^{\text{old}}, \vec{\pi}^{\text{old}}) \\
&= \frac{p(z_{nk} = 1|\vec{\pi}^{\text{old}}) p(\vec{x}_n|z_{nk} = 1, \vec{\mu}^{\text{old}}, \vec{\Sigma}^{\text{old}})}{\sum_{j=1}^{k} p(z_{nj} = 1|\vec{\pi}^{\text{old}}) p(\vec{x}_n|z_{nj} = 1, \vec{\mu}^{\text{old}}, \vec{\Sigma}^{\text{old}})} \\
&= \frac{\pi_k^{\text{old}} \mathcal{N}(\vec{x}_n|\vec{\mu}_k^{\text{old}}, \vec{\Sigma}_k^{\text{old}})}{\sum_{j=1}^{K} \pi_j^{\text{old}} \mathcal{N}(\vec{x}_n|\vec{\mu}_j^{\text{old}}, \vec{\Sigma}_j^{\text{old}})}
\end{aligned}$$

Gauß-Mischmodell, M-Schritt

- Erwartungswert der vollständigen Daten-Log-Likelihood

$$\begin{aligned}
\mathcal{Q}(\vec{\theta}, \vec{\theta}^{\text{old}}) &= \mathbb{E}_{\vec{Z}}[\ln p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\Sigma}, \vec{\pi})] \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \big(\ln \pi_k + \ln \mathcal{N}(\vec{x}_n|\vec{\mu}_k, \vec{\Sigma}_k)\big)
\end{aligned} \tag{108}$$

- Ableiten nach μ_k, Σ_k und π_k, jeweils null setzen und umstellen
  - Bei π_k wieder den Lagrange-Multiplikator $\lambda(\sum_{k=1}^{K} \pi_k - 1$ addieren
- Dies ergibt die gleichen Update-Gleichungen wie (91), (92) und (94).
- Rolle des Erwartungswert der vollständigen Daten-Log-Likelihood wird beim Konvergenzbeweis des EM genauer beleuchtet.

7.3 K-Means als Spezialfall des EM

Beziehung von K-Means zu EM

- Beide Algorithm sind iterativ, K-Means weist Objekte hart den Clustern zu (ganz oder gar nicht), während EM weiche, teilweise Zuweisungen macht.
- K-Means als Spezialfall des EM für Gauß-Mischmodell
  - Annahme: $\vec{\Sigma}_k = \epsilon \vec{I}$
  - ϵ ist das gleiche für alle Komponenten

$$\mathcal{N}(\vec{x}|\vec{\mu}_k, \vec{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp\left\{-\frac{1}{2\epsilon}\|\vec{x} - \vec{\mu}_k\|^2\right\} \tag{109}$$

- Posteriors

$$\gamma(z_{nk}) = \frac{\pi_k \exp\{-\|\vec{x}_n - \vec{\mu}_k\|^2/2\epsilon\}}{\sum_{j=1}^{K} \pi_j \exp\{-\|\vec{x}_n - \vec{\mu}_j\|^2/2\epsilon\}} \tag{110}$$

- $\lim \epsilon \to 0 \Rightarrow$
  - $\gamma(z_{nk}) \to 0$, für $\|\vec{x}_n - \vec{\mu}_j\|^2$ nicht minimal
  - $\gamma(z_{nk}) \to 1$, für $\|\vec{x}_n - \vec{\mu}_j\|^2$ minimal

  $\Rightarrow \gamma(z_{nk}) \to r_{nk}$, siehe (81)

Fehlerfunktion

- Für $\epsilon \to 0$ die erwartete vollständige Daten-Log-Likelihood geht gegen

$$\mathbb{E}_{\vec{Z}}[\ln p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\Sigma}, \vec{\pi})] \to -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\vec{x}_n - \vec{\mu}_j\|^2 + \text{const.} \tag{111}$$

- Maximierung dieser Größe ist äquivalent zu Minimierung der Fehlerfunktion J für K-Means

8 Bernoulli-Mischmodell

Bernoulli-Mischmodell

- Gauß-Mischmodell ist für Vektoren mit kontinuierlichen Attributen
- Viele Daten passen nicht dazu
  - Dokumente nach Boolschem Modell
  - Schwarz/Weiß Bilder
  - Internet-Werbeanzeigen mit Schlüsselwörtern
  - Soziale Netzwerke mit Benutzern, Inhalten und Tags
  - Dünn-besetzte Graphen, z.B. Web, Communities, ...
- D binäre Variablen $x_i, i = 1, \ldots, D$
- Jedes x_i folgt einer eigenen Bernoulli-Verteilung $\text{Bern}(x_i|\mu_i)$
- Für ein Objekt kann mann alle Variablen beobachten, zusammengefaßt als Vektor $\vec{x} \in \{0,1\}^D$ mit $\vec{x} = (x_1, \ldots, x_D)^T$.

8.1 Mehrdimensionale Bernoulli-Verteilung und Mischmodell

Mehrdimensionale Bernoulli-Verteilung

- Mehrdimensionale Bernoulli-Verteilung

$$p(\vec{x}|\vec{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i}(1-\mu_i)^{1-x_i} \tag{112}$$

  mit $\vec{\mu} = (\mu_1, \ldots, \mu_D)^T$
  - Alle D binäre Variablen sind unabhängig
- Erwartungswert

$$\mathbb{E}[\vec{x}] = \vec{\mu} \tag{113}$$

- Kovarianzmatrix

$$\text{cov}[\vec{x}] = \text{diag}\{\mu_i(1-\mu_i)\} \tag{114}$$

  - Eine einzelne mehrdimensionale Bernoulli-Verteilung kann keine Korrelationen zwischen den Variablen modellieren.

Mehrdimensionales Bernoulli-Mischmodell

- Bernoulli-Mischmodell

$$\vec{x} \in \{0,1\}^D, p(\vec{x}|\vec{\mu}, \vec{\pi}) = \sum_{k=1}^{K} \pi_k p(\vec{x}|\vec{\mu}_k) \tag{115}$$

  mit $\vec{\mu} = \{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$, $\vec{\pi} = \{\pi_1, \ldots, \pi_K\}$ und

$$p(\vec{x}|\vec{\mu}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i}(1-\mu_{ki})^{1-x_i} \tag{116}$$

- Erwartungswert

$$\mathbb{E}[\vec{x}] = \sum_{k=1}^{K} \pi_k \vec{\mu}_k \tag{117}$$

- Kovarianzmatrix

$$\text{cov}[\vec{x}] = \sum_{k=1}^{K} \pi_k \big(\vec{\Sigma}_k + \vec{\mu}_k \vec{\mu}_k^T\big) - \mathbb{E}[\vec{x}]\mathbb{E}[\vec{x}^T] \tag{118}$$

mit $\Sigma_k = \text{diag}(\mu_{ki}(1 - \mu_{ki}))$

Vergleich

- Im Gegensatz zum mehrdimensiolen Bernoulli-Modell kann das Bernoulli-Mischmodell Kovarianzen zwischen den Variablen modellieren.
- Beim Bernoulli-Mischmodell hat die Kovarianzmatrix Rang K, d.h. sie ist Summe von K Rang-Eins-Matrizen
  - Rang-Eins-Matrix ist ein äußeres Produkt $\vec{x}\vec{x}^T$
- Anwendungen
  - Finden von korrelierten Worten in Dokumentsammlungen
  - Korrelierte Tags in Web 2.0 Anwendungen
  - ...

8.2 EM-Algorithmus für Bernoulli-Mischmodell

Likelihood des Bernoulli-Mischmodells

- Daten als Matrix $\vec{X} = \{\vec{x}_1, \ldots, \vec{x}_N\}$
- Unvollständige Daten-Likelihood

$$p(\vec{X}|\vec{\mu}, \vec{\pi}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k p(\vec{x}_n|\vec{\mu}_k) \tag{119}$$

- Unvollständige Daten-Log-Likelihood

$$\ln p(\vec{X}|\vec{\mu}, \vec{\pi}) = \sum_{n=1}^{N} \ln\Big\{\sum_{k=1}^{K} \pi_k p(\vec{x}_n|\vec{\mu}_k)\Big\} \tag{120}$$

- Keine geschlossene Form für Maximum-Likelihood-Schätzer, weil die Summe innerhalb des Logarithmus auftaucht

Einführen von versteckten Variablen

- Jede Instanz der Daten $\vec{x}$ mit einer versteckten Variablen $\vec{z}$ koppeln
- $\vec{z} = (z_1, \ldots, z_K)$ folgt Eins-aus-K-Schema, $\vec{z} \in \{0, 1\}^K$
- Bedingte Verteilung für $\vec{x}$ gegeben $\vec{z}$

$$p(\vec{x}|\vec{z}, \vec{\mu}, \vec{\pi}) = \prod_{k=1}^{K} p(\vec{x}|\vec{\mu}_k)^{z_k} \tag{121}$$

- Prior-Verteilung für versteckte Variable $\vec{z}$

$$p(\vec{z}) = p(\vec{z}|\vec{\pi}) = \prod_{k=1}^{K} \pi_k^{z_k} \tag{122}$$

- Verteilung für $\vec{x}$ als Randverteilung → Hausaufgabe

Vollständige Daten-Likelihood

- Für gegebene Daten $\vec{X}$ und versteckte Daten $\vec{Z}$ ist die vollständige Daten-Likelihood

$$p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(\vec{x}_n|\vec{\mu}_k)^{z_k} \pi_k^{z_k} \tag{123}$$

- Vollständige Daten-Log-Likelihood

$$\begin{aligned} &\ln p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\pi}) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\{ \ln \pi_k + \sum_{i=1}^{D} \big(x_{ni} \ln \mu_{ki} + (1 - x_{ni}) \ln(1 - \mu_{ki}) \big) \right\} \end{aligned} \tag{124}$$

Transformation

- Maximiere statt der unvollständigen Likelihood, den Erwartungswert der vollständigen Daten-Log-Likelihood über der Posterior-Verteilung der versteckten Variablen.
- Erwartungswert der vollständigen Daten-Log-Likelihood

$$\begin{aligned} \mathcal{Q}(\vec{\mu}, \vec{\pi}|\vec{\mu}^{old}, \vec{\pi}^{old}) &= \mathbb{E}_{\vec{Z}}[\ln p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\pi})] \\ &= \sum_{\vec{Z}} p(\vec{Z}|\vec{X}, \vec{\mu}^{old}, \vec{\pi}^{old}) \ln p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\pi}) \end{aligned} \tag{125}$$

- Zwei Argumente
  1. Nur Terme mit $z_{nk} = 1$ tragen zu (124) bei ⇒ betrachte Posteriors $\gamma(z_{nk}) = p(z_{nk} = 1|\vec{x}_n, \vec{\mu}^{old}, \vec{\pi}^{old})$
  2. Linearität des Erwartungswertes $\mathbb{E}[\sum x_i] = \sum_i \mathbb{E}[x_i]$

$$\begin{aligned} \mathbb{E}[z_{nk}] &= \sum_{z_{nk} \in \{0,1\}} z_{nk} p(z_{nk}|\vec{x}_n, \vec{\mu}^{old}, \vec{\pi}^{old}) \\ &= p(z_{nk} = 1|\vec{x}_n, \vec{\mu}^{old}, \vec{\pi}^{old}) \end{aligned} \tag{126}$$

E-Schritt

- Posteriors

$$\gamma(z_{nk}) = \frac{\pi_k \prod_{i=1}^{D} \mu_{ki}^{x_{ni}} (1 - \mu_{ki})^{1 - x_{ni}}}{\sum_{j=1}^{K} \pi_j \prod_{i=1}^{D} \mu_{ji}^{x_{ni}} (1 - \mu_{ji})^{1 - x_{ni}}} \tag{127}$$

- Numerische Probleme bei hoher Dimensionalität

M-Schritt

- Maximiere

$$\begin{aligned}&\mathcal{Q}(\vec{\mu}, \vec{\pi} | \vec{\mu}^{old}, \vec{\pi}^{old}) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \left\{ \ln \pi_k + \sum_{i=1}^{D} \big(x_{ni} \ln \mu_{ki} + (1 - x_{ni}) \ln(1 - \mu_{ki}) \big) \right\}\end{aligned} \tag{128}$$

  bezüglich μ_{ki} und π_k

- Aktualisierungsgleichungen

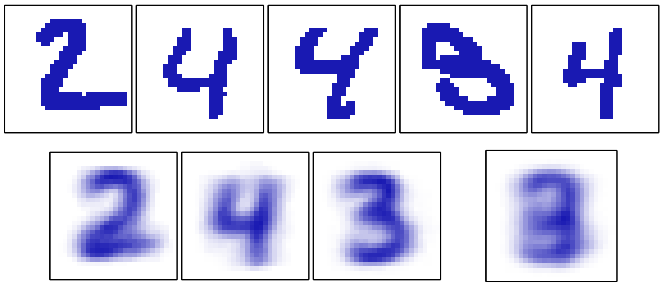
$$\mu_{ki}^{new} = \frac{\bar{x}_{ki}}{N_k} \tag{129}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{130}$$

  mit $\bar{x}_{ki} = \sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}$ und $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$

Beispiel

- Handgeschriebene Zahlen aus $\{2, 3, 4\}$, Bilder wurden binarisiert (Grauwert $> 0.5 \rightarrow$ Pixel auf 1)
- Jedes Pixel ist eine Dimension, $N = 600$ Bilder gegeben, $K = 3$ Bernoulli-Komponenten
- Vermeidung von pathologischen Situationen
  - Initialisierung $\pi_k = 1/K$, μ_{ki} zufällig aus $(0.25, 0.75)$, dann Normalisierung, s.d. $\sum_j \mu_{kj} = 1$.



- Oben: Original-Daten, Unten,links: Komponenten des Mischmodells, Unten,rechts: Einzelne Bernoulli-Verteilung

9 Multinomial-Mischmodell

Multinomial-Mischmodell

- Bernoulli-Mischmodell modelliert die Existenz eines Wortes in einem Dokument
  - Häufigkeiten der Worte werden ignoriert
- Multinomial-Verteilung
  - D Möglichkeiten eine nominale Zufallsvariable zu belegen (z.B. Würfel)
  - Eine Beobachtung besteht aus D absoluten Häufigkeiten (Anzahlen) der einzelnen Zustände

$$\vec{x} = (x_1, \ldots, x_D)^T, \;\; x_i \in \mathbb{N}, N = \sum_{i=1}^{D} x_i \tag{131}$$

  - $\vec{x}$ folgt Multinomial-Verteilung $\text{Mult}(\vec{x}|\vec{\mu}, N)$ mit $\vec{\mu} = (\mu_1, \ldots, \mu_D)^T$, $0 \leq \mu_i \leq 1$ und $\sum_{i=1}^{D} \mu_i = 1$
  - Das N kann im gegebenen Teil auch weggelassen werden, da es sich aus $\vec{x}$ ergibt, d.h. $\text{Mult}(\vec{x}|\vec{\mu}, N) = \text{Mult}(\vec{x}|\vec{\mu})$

Multinomial-Verteilung

- Multinomial-Verteilung

$$\text{Mult}(\vec{x}|\vec{\mu}) = \frac{(\sum_{i=1}^{D} x_i)!}{\prod_{i=1}^{D} x_i!} \prod_{i=1}^{D} \mu_i^{x_i} \tag{132}$$

  mit $\sum_{i=1}^{D} \mu_i = 1$

- Erwartungswert und Kovarianz

$$\mathbb{E}[\vec{x}] = (\sum_{i=1}^{D} x_i) \cdot \vec{\mu} \tag{133}$$

$$cov[\vec{x}] = -(\sum_{i=1}^{D} x_i) \cdot \vec{\mu}\vec{\mu}^T \tag{134}$$

Diskrete Verteilung

- Zugehörige Diskrete Verteilung zur Multinomial-Verteilung ist eine andere mehrdimensionale Verallgemeinerung der Bernoulli-Verteilung
  - Statt zwei Möglichkeiten beim Bernoulli-Versuch gibt es hier D mögliche Ergebnisse, $\sum_{i=1}^{D} \mu_i = 1$
  - Zufallsvariable $\vec{x} = (x_1, \ldots, x_D)^T$ mit $\vec{x} \in \{0, 1\}^D$ wird als 1-aus-D Schema modelliert, d.h. $\sum_{i=1}^{D} x_i = 1$

$$\text{Disc}(\vec{x}|\vec{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i} \tag{135}$$

- Erwartungswert und Kovarianz

$$\mathbb{E}[\vec{x}] = \vec{\mu} \tag{136}$$

$$\text{cov}[\vec{x}] = \text{diag}(\mu_1, \ldots, \mu_D) \tag{137}$$

Multinomial-Mischmodell

- Daten sind Vektoren mit absoluten Häufikeiten $\vec{x} \in \mathbb{N}^D$
  - Zum Beispiel Worthäufigkeiten eines Dokuments
- Verteilung des Mischmodells

$$p(\vec{x}|\vec{\mu}, \vec{\pi}) = \sum_{k=1}^{K} \pi_k \text{Mult}(\vec{x}|\vec{\mu}_k) \tag{138}$$

  mit $\vec{\mu} = \{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$, $\sum_{i=1}^{D} \mu_{ki} = 1$ und $\sum_{k=1}^{K} \pi_k = 1$
- Daten als Matrix $X = \{\vec{x}_1, \ldots, \vec{x}_N\}$
- Unvollständige Daten-Log-Likelihood

$$\ln p(\vec{X}|\vec{\mu}, \vec{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \text{Mult}(\vec{x}_n|\vec{\mu}_k) \right\} \tag{139}$$

9.1 EM-Algorithmus für Multinomial-Mischmodell

Einführen von versteckten Variablen

- Jede Instanz der Daten $\vec{x}$ mit einer versteckten Variablen $\vec{z}$ koppeln
- $\vec{z} = (z_1, \ldots, z_K)$ folgt Eins-aus-K-Schema, $\vec{z} \in \{0, 1\}^K$
- Bedingte Verteilung für $\vec{x}$ gegeben $\vec{z}$

$$p(\vec{x}|\vec{z}, \vec{\mu}, \vec{\pi}) = \prod_{k=1}^{K} p(\vec{x}|\vec{\mu}_k)^{z_k} \tag{140}$$

- Prior-Verteilung für versteckte Variable $\vec{z}$

$$p(\vec{z}) = p(\vec{z}|\vec{\pi}) = \prod_{k=1}^{K} \pi_k^{z_k} \tag{141}$$

- Verteilung für $\vec{x}$ als Randverteilung

Vollständige Daten-Likelihood Multinomial-Mischmodell

- Für gegebene Daten $\vec{X}$ und versteckte Daten $\vec{Z}$ ist die vollständige Daten-Likelihood

$$p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(\vec{x}_n|\vec{\mu}_k)^{z_{nk}} \pi_k^{z_{nk}} \tag{142}$$

- Vollständige Daten-Log-Likelihood

$$\ln p(\vec{X}, \vec{Z}|\vec{\mu}, \vec{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\{ \ln \pi_k + \sum_{i=1}^{D} x_{ni} \ln \mu_{ki} + \ln \left(\sum_{i=1}^{D} x_{ni} \right)! - \sum_{i=1}^{D} \ln x_{ni}! \right\} \tag{143}$$

Transformation

- Maximiere statt der unvollständigen Likelihood, den Erwartungswert der vollständigen Daten-Log-Likelihood über der Posterior-Verteilung der versteckten Variablen.
- Erwartungswert der vollständigen Daten-Log-Likelihood

$$\begin{aligned}\mathcal{Q}(\vec{\mu},\vec{\pi}|\vec{\mu}^{old},\vec{\pi}^{old}) &= \mathbb{E}_{\vec{Z}}[\ln p(\vec{X},\vec{Z}|\vec{\mu},\vec{\pi})] \\ &= \sum_{\vec{Z}} p(\vec{Z}|\vec{X},\vec{\mu}^{old},\vec{\pi}^{old}) \ln p(\vec{X},\vec{Z}|\vec{\mu},\vec{\pi}) \end{aligned} \tag{144}$$

- Linearität des Erwartungswertes

$$\begin{aligned}\mathbb{E}[z_{nk}] &= \sum_{z_{nk}\in\{0,1\}} z_{nk} p(z_{nk}|\vec{x}_n,\vec{\mu}^{old},\vec{\pi}^{old}) \\ &= p(z_{nk}=1|\vec{x}_n,\vec{\mu}^{old},\vec{\pi}^{old}) \end{aligned} \tag{145}$$

E-Schritt: Multinomial-Mischmodell

- Posteriors

$$\gamma(z_{nk}) = \frac{\pi_k \prod_{i=1}^{D} \mu_{ki}^{x_{ni}}}{\sum_{j=1}^{K} \pi_j \prod_{i=1}^{D} \mu_{ji}^{x_{ni}}} \tag{146}$$

- Numerische Probleme
  - explizite Berechnung der Mantisse und Exponenten bei den Produkten
  - In der Summe vor der Berechnung 10ner Potenzen ausklammern und kürzen

M-Schritt: Multinomial-Mischmodell

- Maximiere

$$\begin{aligned}&\mathcal{Q}(\vec{\mu},\vec{\pi}|\vec{\mu}^{old},\vec{\pi}^{old}) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk}) \left\{ \ln \pi_k + \sum_{i=1}^{D} x_{ni} \ln \mu_{ki} + c \right\} \end{aligned} \tag{147}$$

  bezüglich μ_{ki} mit Nebenbedingung $\sum_{i=1}^{D} \mu_{ki} = 1$ und π_k mit Nebenbedingung $\sum_{k=1}^{k} \pi_k = 1$

- Aktualisierungsgleichungen

$$\mu_{ki}^{new} = \frac{\bar{x}_{ki}}{\sum_{j=1}^{D} \bar{x}_{kj}} \tag{148}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{149}$$

  mit $\bar{x}_{ki} = \sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}$ und $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$

9.2 Kovarianz von Mischmodellen

Kovarianz von Mischmodellen

Satz Gegeben ein Mischmodell mit $p(\vec{x}|\vec{\theta}) = \sum_{k=1}^{K} \pi_k p(\vec{x}|\vec{\theta}_k)$ und $\mathbb{E}_k[\vec{x}]$ ist Erwartungswert und $\mathrm{cov}_k[\vec{x}]$ ist Kovarianzmatrix der k-ten Komponente, dann sind Erwartungswert und Kovarianzmatrix des Mischmodells:

$$\mathbb{E}[\vec{x}] = \sum_{k=1}^{K} \pi_k \mathbb{E}_k[\vec{x}] \tag{150}$$

$$\mathrm{cov}[\vec{x}] = \sum_{k=1}^{K} \pi_k \big(\mathrm{cov}_k[\vec{x}] + \mathbb{E}_k[\vec{x}]\mathbb{E}_k[\vec{x}^T]\big) - \mathbb{E}[\vec{x}]\mathbb{E}[\vec{x}^T] \tag{151}$$

Beweis siehe Mitschrift

Beispiel 1: Multinomial-Mischmodell

- Multinomial-Komponenten, $\vec{x} \in \mathbb{N}^D$, $\sum_{i=1}^{D} x_i = N$

$$\mathbb{E}_k[\vec{x}] = N\vec{\mu}_k \tag{152}$$

$$\mathrm{cov}_k[\vec{x}] = -N\vec{\mu}_k\vec{\mu}_k^T \tag{153}$$

$$\tag{154}$$

- Mischverteilung

$$\mathbb{E}[\vec{x}] = N\sum_{k=1}^{K} \pi_k \vec{\mu}_k \tag{155}$$

$$\mathrm{cov}[\vec{x}] = \sum_{k=1}^{K} \pi_k N(N-1)\vec{\mu}_k\vec{\mu}_k^T - N^2\Big(\sum_{k=1}^{K} \pi_k\vec{\mu}_k\Big)\Big(\sum_{k=1}^{K} \pi_k\vec{\mu}_k^T\Big) \tag{156}$$

Beispiel 2: Diskrete Verteilung des Multinomial-Mischmodell

- Diskrete Verteilung der Multinomial-Komponenten, $\vec{x} \in \{0,1\}^D$ mit $\sum_{i=1}^{D} x_i = 1$

$$\mathbb{E}_k[\vec{x}] = \vec{\mu}_k \tag{157}$$

$$\mathrm{cov}_k[\vec{x}] = \mathrm{diag}(\mu_{k1}, \ldots, \mu_{kD}) \tag{158}$$

$$\tag{159}$$

- Mischverteilung

$$\mathbb{E}[\vec{x}] = \sum_{k=1}^{K} \pi_k \vec{\mu}_k \tag{160}$$

$$\mathrm{cov}[\vec{x}] = \sum_{k=1}^{K} \pi_k \big(\mathrm{diag}(\mu_{k1}, \ldots, \mu_{kD}) + \vec{\mu}_k\vec{\mu}_k^T\big) - \Big(\sum_{k=1}^{K} \pi_k\vec{\mu}_k\Big)\Big(\sum_{k=1}^{K} \pi_k\vec{\mu}_k^T\Big) \tag{161}$$

Beispiel 3: Gauß-Mischmodell

- Verteilung der Gauß-Komponenten mit $\Sigma_k = \text{diag}(\sigma_{k1}, \ldots, \sigma_{kD})$ und $\vec{x} \in \mathbb{R}^D$

$$\mathbb{E}_k[\vec{x}] = \vec{\mu}_k \tag{162}$$
$$\text{cov}_k[\vec{x}] = \text{diag}(\sigma_{k1}, \ldots, \sigma_{kD}) \tag{163}$$
$$\tag{164}$$

- Mischverteilung

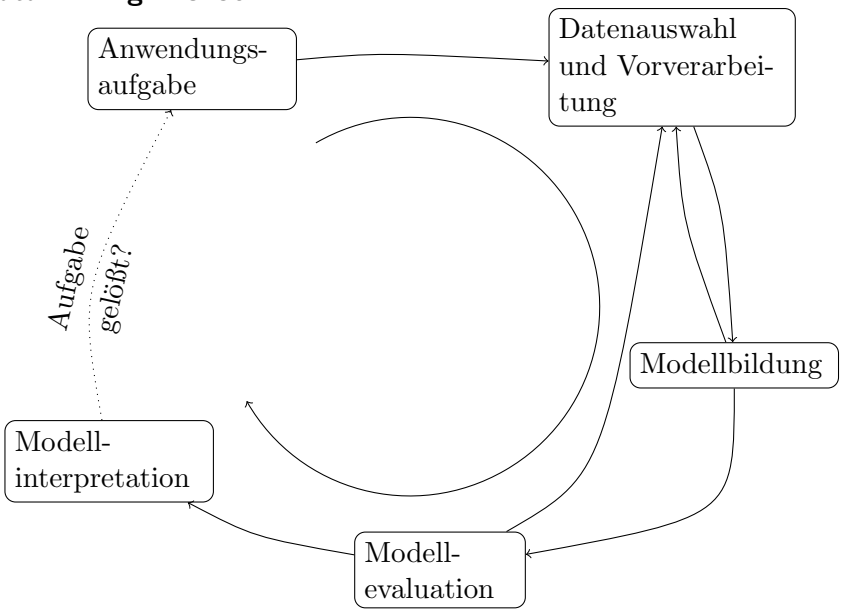
$$\mathbb{E}[\vec{x}] = \sum_{k=1}^{K} \pi_k \vec{\mu}_k \tag{165}$$
$$\text{cov}[\vec{x}] = \sum_{k=1}^{K} \pi_k \big(\text{diag}(\sigma_{k1}, \ldots, \sigma_{kD}) + \vec{\mu}_k \vec{\mu}_k^T\big) - \left(\sum_{k=1}^{K} \pi_k \vec{\mu}_k\right)\left(\sum_{k=1}^{K} \pi_k \vec{\mu}_k^T\right) \tag{166}$$

10 Anwendung des Multinomial-Mischmodell

Anwendung von Mischmodellen

- Bisher: Theorie zum Schätzen der Parameter von Mischmodellen
  - EM-Algorithmus
  - Bernoulli-Verteilung, Multinomial-Verteilung
- Offene Punkte
  - Datenvorverarbeitung, Beispiel Text-Mining
  - Initialisierung der Parameter
  - Implementierung der EM-Algorithmen
  - Wahl der Anzahl der Mischkomponenten
  - Evaluation der Mischmodelle

Data-Mining-Prozeß



10.1 Datenvorverarbeitung

Beispiel Text-Mining

- Anwendungsaufgabe
  - Überblick über die Meldungen auf der DBWorld-Mailing-Liste
- Datenauswahl und Vorverarbeitung
  - Datenbeschaffung
  - Aufbereiten (HTML, Satz- und Sonderzeichen usw. entfernen)
  - Datenbank erstellen und Daten laden
- Modellbildung
  - Bernoulli-Mischmodell in SQL
  - Multinomial-Mischmodell in SQL

  - Initialisierung
- Modellevaluation
  - Welche Wahl für die Anzahl der Mischkomponenten K ist passend?
  - Welche Art Modell ist mehr geeignet?
- Modellinterpretation
  - Welche semantische Bedeutung haben die einzelnen Mischkomponenten?
  - Welche Schlüsse können aus dem Modell gezogen werden?
  - Wie können die gefundenen Korrelationen ausgewertet werden?

Datenauswahl und Vorverarbeitung

- Daten beschaffen
  - Jede eMail ist als separate HTML-Seite gespeichert
  - Links zu diesen Seiten sind im HTML-Code der Überblickseite
    $\Rightarrow$ Liste von Links
  - Dateien mit wget aus dem Netz laden

    ```
    wget http://www.cs.wisc.edu/dbworld/messages/2009-05/1241607365.html
    wget http://www.cs.wisc.edu/dbworld/messages/2009-05/1241597129.html
    ```

- Daten reinigen
  - HTML-Tags entfernen
  - Alle Zeichen zu Kleinbuchstaben konvertieren
  - Alle Zeichen außer Kleinbuchstaben zu Leerzeiche konvertieren
  - Alle mehrfachen Leerzeichen zu einem Leerzeichen zusammenfassen
  - Zeilen in Worte aufspalten
  - Worte pro Dokument in sortierter Reihenfolge (mit Duplikaten) in einzelnen Zeilen ausgeben.
- Worte bei Bedarf auf Wortstamm reduzieren: Porters Stemmer

Text-Vorverarbeitung

- Apache-UIMA-FrameWork (Java) bietet umfangreiche Bibliothek
- Probleme
  - Wörter zusätzlich mit grammatischen Annotationen versehen
    Rightarrow Part-of-Speech-Tagging (POS)
  - Zusammengesetzte Begriffe erkennen
    - im Englischen: z.B. mixture model
  - Fachbegriffe erkennen
    - Linguistische Modelle nutzen
    - Speziell trainierte Random-Markov-Fields, z.B. Bio-Wissenschaften
    - Computer Linguistic Jena: `http://www.julielab.de`
  - Zahlen und Einheiten erkennen
  - Synonyme und Hierarchien von Begriffen beachten
  - ...

Vokabular erstellen

- Mischen aller sortierter Dokumente
  - Merge-Sort
  - Implementiert in Unix Sort, Option -m
- Entfernen aller Duplikate → Vokabular
  - Einfacher Schritt bei sortierten Daten
  - Implementiert in uniq
- Erstellen der Wort-IDs
  - Implementiert durch seq
- Zusammenfügen der IDs mit Vokabular
  - Implementiert durch paste

```
sort -m SourceDir/*.token |uniq > tmp_1
seq 1 ‘wc -l <tmp_1‘ > tmp_2
paste tmp_2 tmp_1 >TargetDir/vocabulary.txt
```

Term-Dokument-Matrix erstellen

- Berechne für jedes Dokument
  - die Häufigkeit seiner Wörter und
  - den Verbund mit der Vokabular-Datei

  ⇒ (Dokument-ID, Wort-ID, Häufigkeit)-Tripel

```
docid=1
for Dokument in $( ls TargetDir/*.token); do
  for word in ‘cat $Dokument‘; do
    echo $docid $word >>tmp1
  done
  uniq -c tmp1 >tmp2
  join -1 3 -2 2 tmp2 Vokabulary.txt       | cut -d’ ’ -f2,3,4 >> TermDokMatrix
  echo $docid $Dokument >> Dokuments
  let docid=docid+1
done
```

Datenbank

- Tabellen
  - term(termid, term varchar(255))
  - doc(docid, doc varchar(255))
  - term_doc(docid, termid, tf)
- Tabellen mit den ersten 50 Dokumenten
  - term_50(termid, term varchar(255))
  - doc_50(docid, doc varchar(255))

  - term_doc_50(docid, termid, tf)
- Daten mit Load-Befehlen in Tabellen laden, anstatt mit Insert
- Indexe für Primärschlüssel erst nach dem Laden erzeugen

10.2 Initialisierung der Parameter des EM-Algorithmus

Allgemeine Parmeterinitialisierung für den EM

- Zwei prinzipielle Möglichkeiten
  1. Initialisierung der Parameter $\vec{\mu} = \{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$ und $\vec{\pi} = \{\pi_1, \ldots, \pi_K\}$
  2. Initialisierung der Posteriors $\gamma(z_{nk})$
- Kosten
  - Parameter: Anzahl der Zufallszahlen ist $K \cdot D$
  - Posteriors: Anzahl der Zufallszahlen ist $K \cdot N$
- Diskussion
  - Aufgrund des Aufwandes könnte man sich für die Methode mit weniger der Zufallszahlen entscheiden: ist $D < N$
  - Einfache Implementierung
    * Posteriors haben bei Mischmodellen immer gleiche Struktur (Dirichlet-Verteilung)
    * Parameter folgen je nach Modell anderen Verteilungen
    * Vorteil für Posteriors
  - Nicht genutzte Komponenten
    * Kann bei Parameter-Initialisierung auftreten
    * Ist bei Posterior-Initialisierung unwahrscheinlich
    * Nachteil Parameterinitialisierung

Parmeterinitialisierung beim Multinomial-Mischmodell

- Posteriors wie auch Parameter sind Punkte aus einem Simplex
- Problem: gleichverteilt aus einem K-dimensionalen Simplex ziehen
  - entspricht: aus K-dimensionalen Dirichletverteilung mit $\vec{\alpha} = \vec{1}$ ziehen
- Möglichkeiten
  - Rejection-Sampling
    * Ziehe gleichverteilt aus $(0, 1)^D$ Würfel
    * Lehne Sample ab, wenn es nicht auf dem Simplex liegt
    * Trefferrate sinkt gegen Null, bei steigender Dimension
  - Projection-Sampling
    * Ziehe gleichverteilt aus $(0, 1)^D$ Würfel
    * Projiziere auf Simplex
    * Liefert keine Gleichverteilung auf dem Simplex
  - Sampling von Differenzen
  - Normalisierte Exponential-Verteilung

Gleichverteilt aus K-dimensionalen Simplex ziehen 1/2

- Sampling von Differenzen
  - $K-1$ Werte gleichverteilt aus $(0,1)$ ziehen
  - Sei $s_0, s_1, \ldots, s_{K-1}, s_K$ die sortierte Sequenz dieser Werte, mit $s_0 = 0$ und $s_K = 1$
  - $\vec{d} = (d_1, \ldots, d_K)^T$ mit $d_i = s_i - s_{i-1}$ ist gleichverteilt im K-dimensionalen Simplex
  - Beweis mittels Ordnungs-Statistiken
    `http://www-stat.stanford.edu/~susan/courses/s116/node79.html`

Gleichverteilt aus K-dimensionalen Simplex ziehen 2/2

- Normalisierte Exponential-Verteilung
  - Ziehe K Werte $x_1, \ldots, x_K$ aus einer Exponentialverteilung
    - Ziehe Wert y_i gleichverteilt aus $(0,1)$
    - Setze $x_i = -\log y_i$
  - Sei $S = \sum_{i=1}^{K} x_i$
  - $\vec{d} = (d_1, \ldots, d_K)^T$ mit $d_i = x_i/S$ ist gleichverteilt im K-dimensionalen Simplex
- Material
  - `http://geomblog.blogspot.com/2005/10/sampling-from-simplex.html`
  - `http://en.wikipedia.org/wiki/Simplex\#Random_sampling`
  - Buch von Luc Devroye: Non-Uniform Random Variate Generation, frei unter `http://cg.scs.carleton.ca/~luc/rnbookindex.html`

Initialisierung: Diskussion und Zusammenfassung

- Posterior-Initialisierung scheint leichte Vorteile zu haben
  - läßt sich allgemein für Mischmodelle nutzen
  - Vermeidet kaum genutzte Komponenten
- Gleichverteiltes Ziehen aus dem Simplex
  - Normalisierte Exponential-Verteilung hat lineare Komplexität $O(K)$ anstelle von $O(K \log K)$ von der Differenzenmethode

10.3 EM-Implementierung

Implementierung des EM

- EM-Algorithmus für Multinomial-Mischmodell kann in jeder Programmiersprache implementiert werden.
- Operationen sind hauptsächlich Berechnungen von großen Summen
- Datenbanken bieten effiziente Algorithmen zum Durchlesen von großen Daten
- SQL kann Summen mittels Aggregatfunktion berechenen
- Nachteil: SQL hat keine While-Schleife
  - Wähle Anzahl der Iterationen fest

- Beispiel: $K = 3$, Startwert für $\pi_1 = \pi_2 = \pi_3 = 1/3$
- Startwerte für $\vec{\mu}_k$ werden in Term-Tabelle gespeichert
  - term(termid, term varchar(255), mu1, mu2, mu3)

Initialisierung

- Parameter-Initialisierung, Normalisierte Exponentialverteilung
- Würfeln der Exponentialverteilung für $\vec{\mu}_k$

```
update term set (mu1,mu2,mu3) =
 (select
   (-1.0)*log(DBMS_RANDOM.VALUE+termid+1-1-termid,10),
   (-1.0)*log(DBMS_RANDOM.VALUE+termid+2-2-termid,10),
   (-1.0)*log(DBMS_RANDOM.VALUE+termid+3-3-termid,10)
 from dual);
```

- Normalisieren

```
update term set
     mu1 = mu1 /( select sum(mu1) from term),
     mu2 = mu2 /( select sum(mu2) from term),
     mu3 = mu3 /( select sum(mu3) from term);
```

E-Schritt: Multinomial-Mischmodell

- Posteriors

$$\gamma(z_{nk}) = \frac{\pi_k \prod_{i=1}^{D} \mu_{ki}^{x_{ni}}}{\sum_{j=1}^{K} \pi_j \prod_{i=1}^{D} \mu_{ji}^{x_{ni}}} \tag{167}$$

- Numerische Probleme

E-Schritt: Berechnen der Posteriors

```
create view posterior_it0 as (
select z3.docid,
   power(z3.a1 - z3.min_a,10)/z3.norm_const as gamma_z1,
   power(z3.a2 - z3.min_a,10)/z3.norm_const as gamma_z2,
   power(z3.a3 - z3.min_a,10)/z3.norm_const as gamma_z3
from (
select z2.docid, z2.a1, z2.a2, z2.a3, z2.min_a,
       power(z2.a1-z2.min_a,10)+ power(z2.a2-z2.min_a,10) +
       power(z2.a3-z2.min_a,10) as norm_const
from (
select z1.docid, z1.a1, z1.a2, z1.a3,
     least(z1.docid, z1.a1, z1.a2, z1.a3) min_a
from (
select td.docid,
       log(1/3,10)+sum(td.tf *log(t.mu1,10)) as a1,
       log(1/3,10)+sum(td.tf *log(t.mu2,10)) as a2,
       log(1/3,10)+sum(td.tf *log(t.mu3,10)) as a3
from term_doc td, term t where td.termid = t.termid
group by td.docid
) z1 ) z2 ) z3 );
```

M-Schritt: Multinomial-Mischmodell

- Maximiere

$$\begin{aligned}&\mathcal{Q}(\vec{\mu},\vec{\pi}|\vec{\mu}^{old},\vec{\pi}^{old})\\&=\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left\{\ln\pi_k+\sum_{i=1}^{D}x_{ni}\ln\mu_{ki}+c\right\}\end{aligned}\tag{168}$$

  bezüglich μ_{ki} mit Nebenbedingung $\sum_{i=1}^{D}\mu_{ki}=1$ und π_k mit Nebenbedingung $\sum_{k=1}^{k}\pi_k=1$

- Aktualisierungsgleichungen

$$\mu_{ki}^{new}=\frac{\bar{x}_{ki}}{\sum_{j=1}^{D}\bar{x}_{kj}}\tag{169}$$

$$\pi_k^{new}=\frac{N_k}{N}\tag{170}$$

  mit $\bar{x}_{ki}=\sum_{n=1}^{N}\gamma(z_{nk})x_{ni}$ und $N_k=\sum_{n=1}^{N}\gamma(z_{nk})$

M-Schritt: Berechnen der π_k

```
create view pi_it1 as (
select N1/N as pi1,
       N2/N as pi2,
       N3/N as pi3
from (
select sum(gamma_z1) as N1,
       sum(gamma_z2) as N2,
       sum(gamma_z3) as N3,
       count(*) as N
from posterior_it0
) z1
);
```

M-Schritt: Berechnen der $\vec{\mu}_k$

```
create view sum_xbar as (
select sum(p0.gamma_z1 * td.tf) as  sxbar1,
       sum(p0.gamma_z2 * td.tf) as  sxbar2,
       sum(p0.gamma_z3 * td.tf) as  sxbar3
from posterior_it0 p0, term_doc td
where p0.docid = td.docid
);

create view mu_it1 as (
select td.termid,
  sum(p0.gamma_z1 * td.tf)/sx.sxbar1 as  mu1,
  sum(p0.gamma_z2 * td.tf)/sx.sxbar2 as  mu2,
  sum(p0.gamma_z3 * td.tf)/sx.sxbar3 as  mu3
from posterior_it0 p0, term_doc td, sum_xbar sx
where p0.docid = td.docid
group by td.termid, sx.sxbar1,sx.sxbar2, sx.sxbar3
);
```

E-Schritt: Multinomial-Mischmodell

- Posteriors

$$\gamma(z_{nk}) = \frac{\pi_k \prod_{i=1}^{D} \mu_{ki}^{x_{ni}}}{\sum_{j=1}^{K} \pi_j \prod_{i=1}^{D} \mu_{ji}^{x_{ni}}} \tag{171}$$

E-Schritt: Berechnen der nächsten Posteriors

```
create view posterior_it1 as (
select z3.docid,
   power(z3.a1 - z3.min_a,10)/z3.norm_const as gamma_z1,
   power(z3.a2 - z3.min_a,10)/z3.norm_const as gamma_z2,
   power(z3.a3 - z3.min_a,10)/z3.norm_const as gamma_z3
from (
select z2.docid, z2.a1, z2.a2, z2.a3, z2.min_a,
       power(z2.a1-z2.min_a,10)+ power(z2.a2-z2.min_a,10) +
       power(z2.a3-z2.min_a,10) as norm_const
from (
select z1.docid, z1.a1, z1.a2, z1.a3,
least(z1.docid, z1.a1, z1.a2, z1.a3) min_a
from (
select td.docid,
       log(p1.pi1,10)+sum(td.tf *log(m1.mu1,10)) as a1,
       log(p1.pi2,10)+sum(td.tf *log(m1.mu2,10)) as a2,
       log(p1.pi3,10)+sum(td.tf *log(m1.mu3,10)) as a3
from term_doc td, mu_it1 m1, pi_it1 where td.termid=m1.termid
group by td.docid, log(pi1,10), log(pi2,10), log(pi3,10)
) z1 ) z2 ) z3 );
```

Numerische Probleme beim E-Schritt

- Numerische Umformung hilft nicht bei sehr großen Exponenten
- Beispiel

DocID	Z1.A1	Z1.A2	Z1.A3	Z1.Min_A
28	-1003.5657	-1134.0201	-1463.7964	-1463.7964

- Differenz -1003+ 1463= 460 hoch 10 ist zu groß

```
select z2.docid, z2.a1, z2.a2, z2.a3, z2.min_a,
       power(z2.a1-z2.min_a,10)+ power(z2.a2-z2.min_a,10) +
       power(z2.a3-z2.min_a,10) as norm_const
from (
select z1.docid, z1.a1, z1.a2, z1.a3,
least(z1.docid, z1.a1, z1.a2, z1.a3) min_a
from (
...
) z1 ) z2;
```

Inspektion des Ergebnisses 1/3

- Ausgabe der π_k nach der ersten Iteration

```
select * from pi_50_it1 ;

       PI1        PI2        PI3
---------- ---------- ----------
.023663113 .079507953 .896828934
```

Inspektion des Ergebnisses 2/3

- Nach der ersten Iteration die 30 Worte mit den größten μ_{ki} für jede Komponente ausgegeben

```
select term
from (
select rownum, termid, mu1
from mu_50_it1 m1
order by mu1 desc
) z1, term_50 t
where rownum <=10 and
 z1.termid = t.termid
order by z1.mu1 desc
```

Inspektion des Ergebnisses 3/3

- Nach der ersten Iteration die 30 Worte mit den größten μ_{ki} für jede Komponente ausgegeben

$\vec{\mu}_1$ and the for of de on a to workshop in web http information papers submission conference paper geospatial www ibima be applications will multimedia feb deadline are n y service

$\vec{\mu}_2$ and of in the for univ to social ue on network paper nas papers be a submission workshop conference security by information will management http with issues international web korea

$\vec{\mu}_3$ of and the university to in for a be on systems papers data information will is web research usa workshop http are paper at submission italy conference by or as

Inspektion des Ergebnisses bei Posterior-Initialisierung 1/2

- Ausgabe der π_k nach der ersten Iteration

```
 select * from pi_50_p_it2;

       PI1        PI2        PI3
---------- ---------- ----------
6.6564E-22  .93250092  .06749908
```

Inspektion des Ergebnisses bei Posterior-Initialisierung 2/2

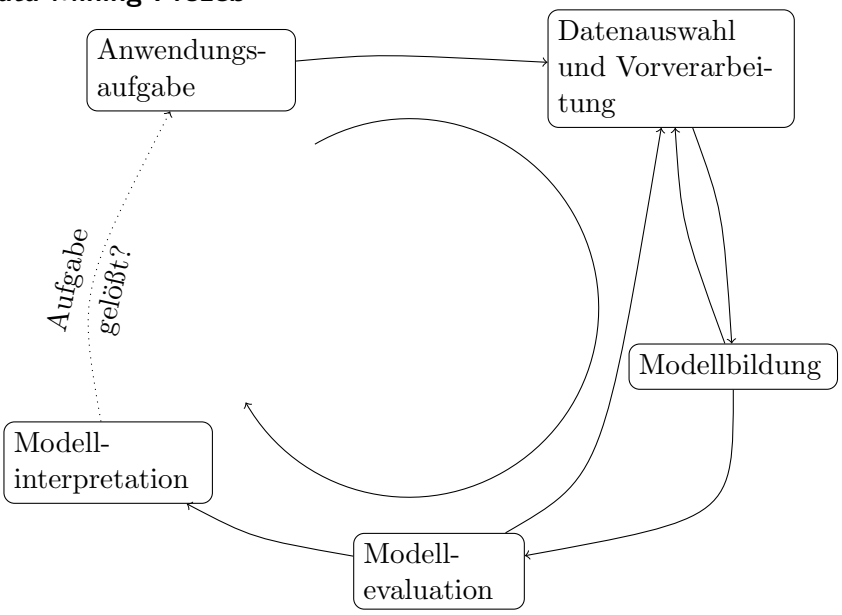
- Nach der ersten Iteration die 30 Worte mit den größten μ_{ki} für jede Komponente ausgegeben

$\vec{\mu}_1$ university a u s and of the in be concordia must security conference canada rutgers to for email papers paper state data italy di secure applications milano information universit proposals

$\vec{\mu}_2$ of and the to in university for a be on papers systems data information will workshop web http is are paper research submission at conference by as www or with

$\vec{\mu}_3$ university of and the usa to italy france in for australia technology systems research germany data de japan china austria national universit uk information hong di at a management be

Data-Mining-Prozeß



11 EM-Algorithmus für MAP-Schätzung

Motivation für MAP-Schätzer

- Bisher: Maximum-Likelihood-Schätzer

$$\text{argmax}_{\vec{\theta}} p(\vec{X}|\vec{\theta}) \tag{172}$$

  - Bernoulli-Mischmodell: $\vec{\theta} = \{\vec{\mu}, \vec{\pi}\}$
  - Multinomial-Mischmodell: $\vec{\theta} = \{\vec{\mu}, \vec{\pi}\}$

- Nachteile:
  - Unbalancierte Mischkomponenten
  - Singularitäten bei kontinuierlichen Variablen
  - Kein Zusatzwissen

- Idee: mittels Prior-Verteilungen ungünstige Parametereinstellungen bestrafen

MAP-Schätzer

- Maximum-A-Posteriory (MAP) Schätzer

$$\text{argmax}_{\vec{\theta}} p(\vec{\theta}|\vec{X}) \tag{173}$$

- Transformation des Problems mit Bayesscher Regel

$$p(\vec{\theta}|\vec{X}) = \frac{p(\vec{X}|\vec{\theta})p(\vec{\theta})}{p(\vec{X})} \tag{174}$$

- Für die Maximierung reicht es nur den Logarithmus des Zählers zu maximieren

$$\ln p(\vec{X}|\vec{\theta}) + \ln p(\vec{\theta}) \tag{175}$$

EM-Algorithmus für MAP-Schatzer

- Linker Term von 175 ist unvollständige Log-Daten-Likelihood
- Einführen von versteckten Variablen und Transformation des Maximierungsproblems wie bei Maximum-Likelihood-Schätzer
- Erwartungswert von 175

$$\begin{aligned}
\mathcal{Q}'(\vec{\theta}|\vec{\theta}^{old}) &= \mathbb{E}_{\vec{Z}}[\ln p(\vec{X}, \vec{Z}|\vec{\theta}) + \ln p(\vec{\theta})] \\
&= \mathbb{E}_{\vec{Z}}[\ln p(\vec{X}, \vec{Z}|\vec{\theta})] + \ln p(\vec{\theta}) && (176) \\
&= \mathcal{Q}(\vec{\theta}|\vec{\theta}^{old}) + \log p(\vec{\theta}) && (177) \\
&= \left(\sum_{\vec{Z}} p(\vec{Z}|\vec{X}, \vec{\theta}^{old}) \ln p(\vec{X}, \vec{Z}|\vec{\theta})\right) + \ln p(\vec{\theta}) && (178) \\
& && (179)
\end{aligned}$$

- Weil Posterior der versteckten Variablen nicht von neuen Prior betroffen ⇒ E-Schritt wie beim ML-Schätzer

M-Schritt für Map-Schätzer

- Maximiere $\mathcal{Q}'(\vec{\theta}|\vec{\theta}^{old})$, d.h.

$$\mathcal{Q}(\vec{\theta}|\vec{\theta}^{old}) + \log p(\vec{\theta}) \tag{180}$$

MAP für Bernoulli-Beta-Mischmodell

- Prior

$$p(\vec{\theta}) = p(\vec{\pi}, \vec{\mu}) = \mathrm{Dir}(\vec{\pi}|\vec{\alpha}) \cdot \prod_{k=1}^{K} \mathrm{Beta}(\vec{\mu}_k|\vec{a}_k, \vec{b}_k) \tag{181}$$

- M-Schritt

$$\pi_k = \frac{N_k + \alpha_k - 1}{N + \sum_{l=1}^{K} \alpha_l - K} \tag{182}$$

$$\mu_{ki} = \frac{\bar{x}_{ki} + a_{ki} - 1}{N_k + a_{ki} - 1 + b_{ki} - 1} \tag{183}$$

- $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$ und $\bar{x}_{ki} = \sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}$

MAP für Multinomial-Dirichlet-Mischmodell

- Prior

$$p(\vec{\theta}) = p(\vec{\pi}, \vec{\mu}) = \mathrm{Dir}(\vec{\pi}|\vec{\alpha}) \cdot \prod_{k=1}^{K} \mathrm{Dir}(\vec{\mu}_k|\vec{\beta}_k) \tag{184}$$

- M-Schritt

$$\pi_k = \frac{N_k + \alpha_k - 1}{N + \sum_{l=1}^{K} \alpha_l - K} \tag{185}$$

$$\mu_{ki} = \frac{\bar{x}_{ki} + \beta_{ki} - 1}{\left(\sum_{i'=1}^{D} \bar{x}_{ki'} + \beta_{ki'}\right) - D} \tag{186}$$

- $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$ und $\bar{x}_{ki} = \sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}$

12 Konvergenz des EM-Algorithmus

Allgemeine Behandlung des EM-Algorithmus

- Probabilistische Modell mit
  - beobachtbaren Variablen $\vec{X}$
  - versteckten Variablen $\vec{Z}$
    - Annahme: $\vec{Z}$ ist diskret,
    - falls nicht, werden aus den Summen Integrale
  - Parameter $\vec{\theta}$
- Ziel
  - Maximiere Likelihood

$$p(\vec{X}|\vec{\theta}) = \sum_{\vec{Z}} p(\vec{X}, \vec{Z}|\vec{\theta}) \tag{187}$$

- Dies ist äquivalent zum Maximieren der Log-Likelihood $\ln p(\vec{X}|\vec{\theta})$

Zerlegung der Log-Likelihood

- Idee: zerlege Log-Likelihood bezüglich einer beliebigen Verteilung $q(\vec{Z})$ über den versteckten Variablen
- Sei $q(\vec{Z})$ irgend eine Verteilung über den versteckten Variablen $\vec{Z}$, dann gilt

$$\ln p(\vec{X}|\vec{\theta}) = \mathcal{L}(q, \vec{\theta}) + \mathrm{KL}(q||p) \text{ mit} \tag{188}$$

$$\mathcal{L}(q, \vec{\theta}) = \sum_{\vec{Z}} q(\vec{Z}) \ln \frac{p(\vec{X}, \vec{Z}|\vec{\theta})}{q(\vec{Z})} \tag{189}$$

$$\mathrm{KL}(q||p) = -\sum_{\vec{Z}} q(\vec{Z}) \ln \frac{p(\vec{Z}|\vec{X}, \vec{\theta})}{q(\vec{Z})} \tag{190}$$

Exkurs: KL-Divergenz

- KL-Divergenz (Kullback, Leibler, 1951) oder relative Entropie ist von zwei Verteilungen $a(x)$ und $b(x)$ abhängig, die die gleiche Domäne x haben.

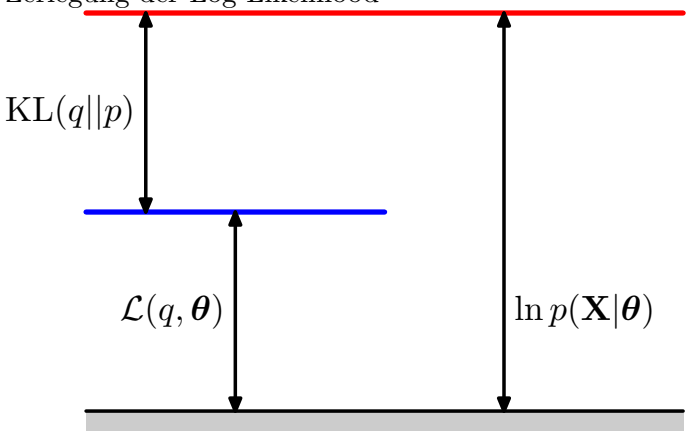
$$\mathrm{KL}(a||b) = -\sum_x a(x) \ln \frac{b(x)}{a(x)} \tag{191}$$

$$= \sum_x a(x) \ln a(x) - \sum_x a(x) \ln b(x) \tag{192}$$

- Eigenschaften
  - $\mathrm{KL}(a||b) \geq 0$ mit Gleichheit genau dann wenn die beiden Verteilungen gleich sind $a(x) = b(x)$
  - Nicht symmetrisch $\mathrm{KL}(a||b) \neq \mathrm{KL}(b||a)$
  - Dreieckungleichung gilt nicht

Untere Schranke für Log-Likelihood

- Zerlegung der Log-Likelihood



- Untere Schranke für Log-Likelihood

$$\ln p(\vec{X}|\vec{\theta}) \geq \mathcal{L}(q, \vec{\theta}) \tag{193}$$

- Untere Schranke gilt für beliebige Verteilungen $q(\vec{Z})$

Konvergenz des EM

- Idee
  - Maximiere anstelle der Log-Likelihood $\ln p(\vec{X}|\vec{\theta})$ die untere Schranke $\mathcal{L}(q, \vec{\theta})$
  - Maximiere $\mathcal{L}(q, \vec{\theta})$ abwechselnd
    - nach $q(\vec{Z})$ (E-Schritt) und
    - nach $\vec{\theta}$ (M-Schritt)
- Beweisziele
  - Untere Schranke $\mathcal{L}(q, \vec{\theta})$ steigt im E-Schritt
  - Untere Schranke $\mathcal{L}(q, \vec{\theta})$ steigt im M-Schritt
  - Zusammenhang zu EM-Algorithmus

E-Schritt

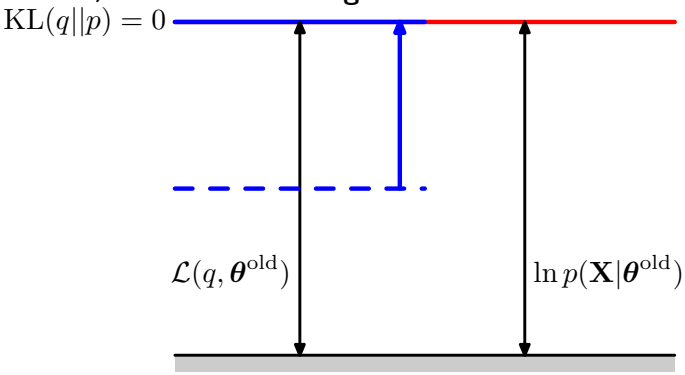
- Maximiere $\mathcal{L}(q, \vec{\theta})$ nach $q(\vec{Z})$
  - Parameter werden mit $\vec{\theta}^{old}$ initialisiert
- $\mathcal{L}(q, \vec{\theta}^{old})$ hängt nur von $q(\vec{Z})$ ab
  - $q(\vec{Z})$ beeinflßt nur $\mathrm{KL}(q||p)$

$$\mathcal{L}(q, \vec{\theta}^{old}) = \ln p(\vec{X}|\vec{\theta}^{old}) - \mathrm{KL}(q||p) \tag{194}$$

  - $\mathcal{L}(q, \vec{\theta}^{old})$ ist maximal, wenn $\mathrm{KL}(q||p) = 0$
    $\Rightarrow q(\vec{Z}) = p(\vec{Z}|\vec{X}, \vec{\theta}^{old})$
  - Wenn $q(\vec{Z})$ so gewählt ist, dann ist

$$\mathcal{L}(q, \vec{\theta}^{old}) = \ln p(\vec{X}|\vec{\theta}^{old}) \tag{195}$$

E-Schritt, Veranschaulichung



- Wenn $q(\vec{Z})$ gleich der Posterior $p(\vec{Z}|\vec{X}, \vec{\theta}^{old})$ gewählt wird
  $\Rightarrow$ untere Schranke $\mathcal{L}(q, \vec{\theta}^{old})$ wird angehoben, bis sie gleich $\ln p(\vec{X}|\vec{\theta}^{old})$ ist

M-Schritt

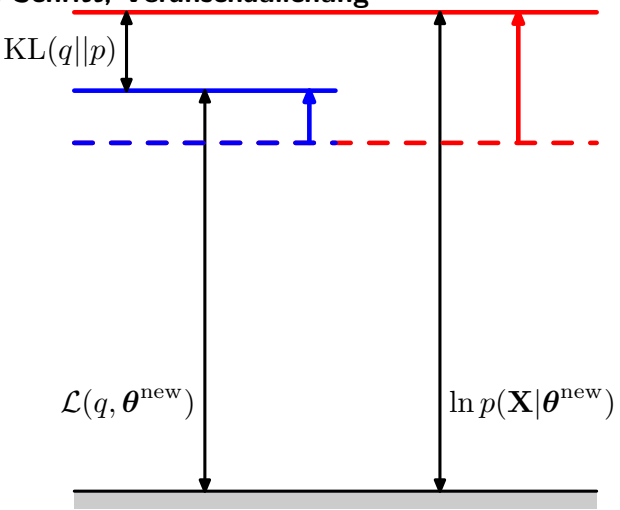
- Wahl für $q(\vec{Z})$ aus E-Schritt wird festgehalten
- Maximiere $\mathcal{L}(q, \vec{\theta})$ nach $\vec{\theta}$
- Wenn $q(\vec{Z}) = p(\vec{Z}|\vec{X}, \vec{\theta}^{old})$ dann ergibt sich für die untere Schranke

$$\mathcal{L}(q, \vec{\theta}) = \sum_{\vec{Z}} p(\vec{Z}|\vec{X}, \vec{\theta}^{old}) \ln \frac{p(\vec{X}, \vec{Z}|\vec{\theta})}{p(\vec{Z}|\vec{X}, \vec{\theta}^{old})} \tag{196}$$

$$= \mathcal{Q}(\vec{\theta}, \vec{\theta}^{old}) + c \tag{197}$$

- Konstante c ist negative Entropie von $p(\vec{Z}|\vec{X}, \vec{\theta}^{old})$
- Maximierung von $\mathcal{Q}(\vec{\theta}, \vec{\theta}^{old})$ ist das was bisher im M-Schritt gemacht wurde
- D.h. der M-Schritt vergrößtert auch die untere Schranke der Log-Likelihood

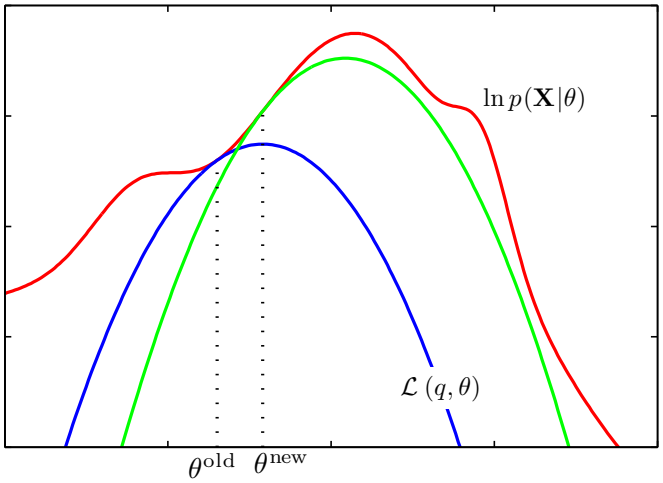
M-Schritt, Veranschaulichung



- Maximierung von $\mathcal{L}(q, \vec{\theta})$ bezüglich $\vec{\theta}$ ergibt $\vec{\theta}^{new}$
- Weil $p(\vec{Z}|\vec{X}, \vec{\theta}^{old}) \neq p(\vec{Z}|\vec{X}, \vec{\theta}^{new})$ ist $\mathrm{KL}(q||p) \geq 0$ bezüglich $\vec{\theta}^{new}$
- Deshalb steigt $\ln p(\vec{X}|\vec{\theta})$ durch den M-Schritt mehr als $\mathcal{L}(q, \vec{\theta})$

Arbeitsweise des EM im Parameterraum

- Starte mit initialen Parameter $\vec{\theta}^{old}$
- $\mathcal{L}(q, \vec{\theta}^{old})$ hat nach E-Schritt Kontakt mit Likelihood $\ln p(\vec{X}|\vec{\theta})$
- Beide Funktionen haben auch gleichen Gradienten



Spezialfall iid. Daten

- $\vec{X} = \{\vec{x}_n\}$, $\vec{Z} = \{\vec{z}_n\}$
- iid. Annahme

$$p(\vec{X}, \vec{Z}) = \prod_{n=1}^{N} p(\vec{x}_n, \vec{z}_n) \tag{198}$$

- Randverteilung

$$p(\vec{X}) = \sum_{\vec{Z}} p(\vec{X}, \vec{Z}) = \sum_{\vec{Z}} \prod_{n=1}^{N} p(\vec{x}_n, \vec{z}_n) = \prod_{n=1}^{N} p(\vec{x}_n)$$

- E-Schritt

$$p(\vec{Z}|\vec{X}, \vec{\theta}) = \frac{\prod_{n=1}^{N} p(\vec{x}_n, \vec{z}_n|\vec{\theta})}{\sum_{\vec{Z}} \prod_{n=1}^{N} p(\vec{x}_n, \vec{z}_n|\vec{\theta})} = \frac{p(\vec{X}, \vec{Z}|\vec{\theta})}{\sum_{\vec{Z}} p(\vec{X}, \vec{Z}|\vec{\theta})} \tag{199}$$

$$= \prod_{n=1}^{N} p(\vec{z}_n|\vec{x}_n, \vec{\theta}) \tag{200}$$

MAP-Schätzung mit EM

- Maximiere $p(\vec{\theta}|\vec{X})$ mit beliebiger Prior Verteilung $p(\vec{\theta})$

$$p(\vec{\theta}|\vec{X}) = \frac{p(\vec{\theta}, \vec{X})}{p(\vec{X})} \Rightarrow \tag{201}$$

$$\ln p(\vec{\theta}|\vec{X}) = \ln p(\vec{\theta}, \vec{X}) - \ln p(\vec{X}) \tag{202}$$

$$= \ln p(\vec{X}|\vec{\theta}) + \ln p(\vec{\theta}) - \ln p(\vec{X}) \tag{203}$$

$$= \mathcal{L}(q, \vec{\theta}) + \mathrm{KL}(q||p) + \ln p(\vec{\theta}) - \ln p(\vec{X}) \tag{204}$$

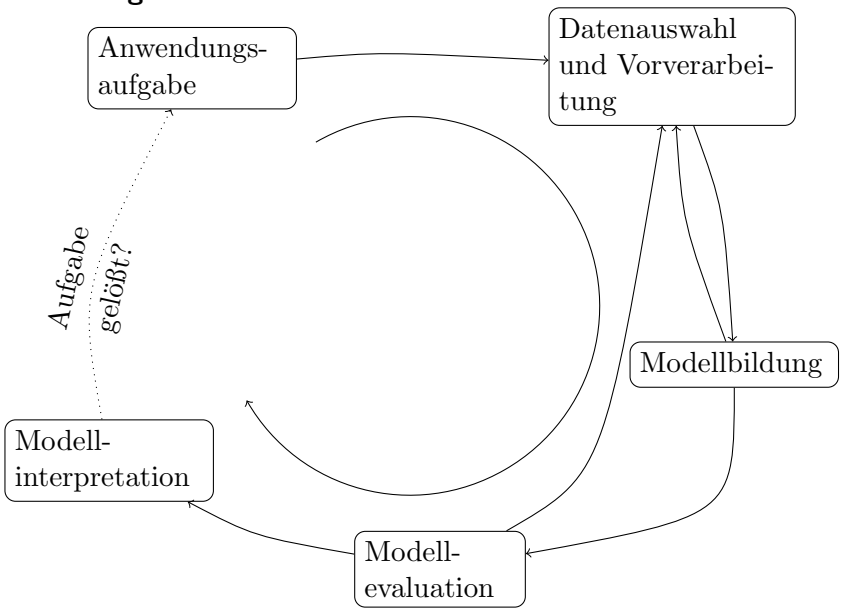
- E-Schritt: optimiere $q(\vec{Z})$ wie bisher
- M-Schritt: maximiere $\mathcal{L}(q, \vec{\theta}) + \ln p(\vec{\theta})$

Erweiterungen des EM

- Statt Maximierung in E- und M-Schritt nur eine Verbesserung der jeweiligen Zielfunktion
- Verallgemeinerter EM, GEM
  - Statt Maximierung im M-Schritt nur eine Steigerung von $\mathcal{L}(q, \vec{\theta})$
  - Einsatz von nicht-linearen Optimierungstechniken
- Online-EM
  - Statt Minimierung von $\mathrm{KL}(q||p)$ nur eine Senkung
  - Bei iid. Daten, nur Posterior eines Beobachtung neu berechnen und dann gleich den M-Schritt durchführen.
  - Beispiel: Multinomial-Mischmodell
  - Reihenfolge der Abarbeitung wird spielt eine Rolle

13 Evaluation

Data Mining Prozeß



Evaluation von Data-Mining-Modellen

- Für dieselbe Data-Mining-Aufgabe gibt es oft mehrere alternative Modelle.
- Ein Data-Mining-Modell hat meist mehrere Parameter, die sich nicht mittels Hintergrundwissen einstellen lassen.
- Fragen
  - **Modellselektion:** Welches Modell ist am besten geeignet oder welche Parametereinstellung soll genutzt werden?
  - **Modellbewertung:** Welchen Fehler macht ein Modell?
- Antworten hängen von der Aufgabe ab
- Wahl des Evaluationsmaßes
- Wahl der Evaluationsmethode

13.1 Evaluationsmaße

Evaluationsmaße

- Die meisten Maße sind für Testdaten entworfen
- Likelihood auf Testdaten als allgemeines Maß für probabilistische Modelle
- Maße für spezifische Aufgabenstellungen
  - Klassifikationsfehler für Klassifikation
  - Approximationsfehler bei Regression
- Maße ohne Testdaten
  - Akaikes Informationskriterim (AIC)
  - Bayesisches Informationskriterium (BIC)

Likelihood auf Testdaten

- Gegeben sei ein probabilistisches Modell $p(x|\vec{\theta})$ mit geschätzten Parametern $\vec{\theta}$
- Die Parameter wurden auf den Trainingsdaten $\vec{X}$ geschätzt, z.B. mittels Maximum-Likelihood oder Maximum-Aposteriory
- Die Likelihood auf den Testdaten $\vec{X}' = \{\vec{x}'_1, \ldots, \vec{x}'_{N'}\}$ ist

$$p(\vec{X}'|\vec{\theta}) = \prod_{n'=1}^{N'} p(\vec{x}'_{n'}|\vec{\theta}) \tag{205}$$

- Die Testdaten, die das Modell bisher noch nie gesehen hat, sollten auch eine hohe Wahrscheinlichkeit bekommen, wenn das Modell sinnvoll gelernt ist.
- Die Likelihood auf den Testdaten $p(\vec{X}'|\vec{\theta})$ ist ein Maß, wie gut das Modell auf neue Daten verallgemeinert.
- Wenn $p(\vec{X}'|\vec{\theta})$ klein ist, hat sich das Modell wahrscheinlich zu sehr auf die Trainingsdaten $\vec{X}$ spezialisiert (Overfitting).

Klassifikationsfehler

- Klassifikation ist eine Funktion $c = f(\vec{x})$ mit Beobachtung $\vec{x}$ als Eingabe und $c \in \{c_1, \ldots, c_K\}$ als Zielvariable
- Klassifikation liefert für eine Beobachtung $\vec{x}$
  - die Klasse c oder
  - die Posterior-Verteilung über den Klassen $p(c = k|\vec{x})$ für $k = 1, \ldots, K$
- Bewertungsmaße für Testdaten $\vec{X}' = \{\vec{x}'_1, \ldots, \vec{x}'_{N'}\}$ mit bekannten Klassen $\vec{C}' = \{c'_1, \ldots, c'_{N'}\}$ sind
  - 0-1 loss

$$L(\vec{C}', f(\vec{X}')) = \frac{1}{N'} \sum_{n'=1}^{N'} I(c'_{n'} = f(\vec{x}'_{n'})) \tag{206}$$

  - Cross-Entropy

$$L(\vec{C}', f(\vec{X}')) = -\sum_{k=1}^{K} \sum_{n'=1}^{N'} I(c'_{n'} = f(\vec{x}'_{n'})) \ln p(c = k|\vec{x}'_{n'}) \tag{207}$$

Approximationsfehler

- (Eindimensionale) Regression ist eine Funktion $f(\vec{x})$ mit Beobachtung $\vec{x}$ als Eingabe und Zielvariable y als Ausgabe
- Bewertungsmaße für Testdaten $\vec{X}' = \{\vec{x}'_1, \ldots, \vec{x}'_{N'}\}$ mit bekannten Zielvariablen $\vec{Y}' = \{y'_1, \ldots, y'_{N'}\}$ sind
  - Quadratischer Fehler

$$L(\vec{Y}', f(\vec{X}')) = \frac{1}{N'} \sum_{n'=1}^{N'} (y'_{n'} - f(\vec{x}'_{n'}))^2 \tag{208}$$

  - Absoluter Fehler

$$L(\vec{Y}', f(\vec{X}')) = \frac{1}{N'} \sum_{n'=1}^{N'} |y'_{n'} - f(\vec{x}'_{n'})| \tag{209}$$

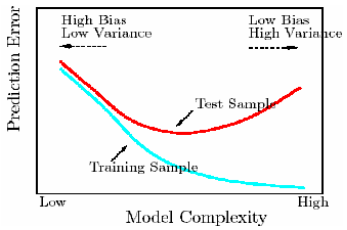
13.2 Trainings-, Validierungs- und Testdaten

Trainings-, Validierungs- und Testdaten

- Wenn viele Daten vorhanden sind sollte die Gesamtdatenmenge idealerweise in
  - Trainingsdaten
  - Validierungsdaten
  - Testdaten

  aufgespalten werden
- Verwendung
  - Modelltraining mit Trainingsdaten
  - Modellselektion mit Validierungsdaten
  - Modellbewertung des endgültigen Modells mit Testdaten, Testdaten bleiben solange unter Verschluß bis endgültiges Modell feststeht
- Typische Aufspaltung 50% Trainingsdaten, 25% Validierungsdaten und 25% Testdaten

Beziehung zwischen Trainings- und Validierungsfehler



Für quadratischen Fehler gilt folgende Zerlegung

$$\text{Fehler} = \text{Nichtreduzierbarer Fehler} + \text{Bias}^2 + \text{Varianz} \tag{210}$$

- Nichtreduzierbarer Fehler: Schwankungen, die durch den Zufallsprozeß entstehen
- Bias: Abweichungen, die durch die Differenz zwischen der Ausgabe des geschätzten Modells und den (unbekannten) wahren Zielgrößen entstehen
- Varianz: Schwankungen, die beim Schätzen des Modells entstehen

Diskussion

- Trainingsfehler ist meist deutlich kleiner als Validierungsfehler
- Problem
  - Gesamtdatenmenge ist meist zu klein um eine sinnvolle Aufteilung in Trainings- und Validierungsmenge zuzulassen.
- Ideen
  - Kreuz-Validierung: Validierungsfehler durch Variation der Daten direkt schätzen
  - Differenz zwischen Trainings- und Validierungsfehler modellieren
  - Bootstrap: Differenz zwischen Trainings- und Validierungsfehler durch Variation der Daten schätzen

13.3 Kreuzvalidierung

Kreuzvalidierung

- Gegeben sind die Daten $\vec{X}$
- Methode
  - Partitioniere $\vec{X}$ zufällig in etwa K gleichgroße Teile
  - Der $k$te Teil wird als Validierungsmenge genutzt. Die restlichen $K-1$ Teile dienen zum Trainieren des Modells. Mit dem so trainierten Modell kann das Evaluationsmaß für jede Beobachtung des $k$ten Teil berechnet werden.
  - Führe den zweiten Schritt für alle Teile $k = 1, \ldots, K$ durch und fasse dann die Schätzungen des Validierungsfehlers zusammen.
- Kreuz-Validierungsschätzer
  - Sei $\kappa\colon \{1, \ldots, N\} \to \{1, \ldots, K\}$ die Indexfunktion, die jede Beobachtung ihrer Partition zuordnet
  - Sei $\hat{f}^{-k}(\vec{x})$ die Ausgabe des Modells, das ohne den $k$ten Teil gelernt wurde
  - Der Kreuz-Validierungsschätzer für die Fehlerfunktion L ist dann

$$CV = \frac{1}{N} \sum_{n=1}^{N} L(y_n, \hat{f}^{-\kappa(n)}(\vec{x}_n)) \tag{211}$$

Diskussion Kreuz-Validierung

- Für Kreuz-Validierung muß das Modell K mal gelernt werden
- Für $K = N$ wird die Kreuz-Validierung zum Leave-One-Out oder Jack-Knife
- Typische Werte sind $K = 5$ oder $K = 10$
- Wie soll K gewählt werden?
  - Für $K = N$ unterscheiden sich die Trainingsmengen kaum $\Rightarrow CV$ is fast ohne Bias, kann aber hohe Varianz haben
  - Für $K = 5$ hat CV eine geringere Varianz, aber Bias kann aufgrund der kleineren Traininigsmengen ein Problem sein.
  - Beispiel: Normalverteilung $\mathcal{N}(x|\mu = 10, \sigma = 1)$, $N = 100$ Beobachtungen, Evaluationsmaß ist mittlere log-Likelihood pro Beobachtung
    - $K = N$: $CV = -1.5817763$, $sd = 0.9395846$, wahre $LL = -1.580407$
    - $K = 10$: $CV = -1.4232838$, $sd = 0.2536275$, wahre $LL = -1.417698$
    - $K = 5$: $CV = -1.5018074$, $sd = 0.1541024$, wahre $LL = -1.488055$

13.4 Bootstrap

Einfacher Bootstrap

- Bootstrap ist auch wie Kreuzvalidierung eine Daten-Simulationsmethode
- B Trainingsmengen werden aus der Datenmenge $\vec{X}$ durch zufälliges Ziehen mit Zurücklegen erzeugt
- Für jede der B Trainingsmenge wird ein Modell gelernt.
- Der Fehler für alle Modelle wird auf der Originaldatenmenge $\vec{X}$ bestimmt
- Leider unterschätzt diese Methode den wahren Fehler, weil die Originaldaten viele Beobachtungen mit den Bootstrap-Samples gemeinsam haben

Beispiel

Beispiel:Normalverteilung $\mathcal{N}(x|\mu = 10, \sigma = 1)$, $N = 100$, Evaluationsmaß ist mittlere log-Likelihood pro Beobachtung

Originaldaten	Bootstrap-Sample	Optimismus
-1.409073246	-1.403020812	0.006052434
-1.405453177	-1.401165041	0.004288135
-1.40891088	-1.50837986	-0.09946899
-1.405668300	-1.396156306	0.009511994
-1.42066249	-1.34595431	0.07470818
-1.40515675	-1.41785483	-0.01269808
-1.40977385	-1.38526100	0.02451284
-1.41025214	-1.45041413	-0.04016198
-1.4053961	-1.5218744	-0.1164783
-1.40769963	-1.36328762	0.04441201
-1.40981595	-1.40112698	0.00868897

Trainingsdaten: -1.405157
Trainingsdaten + Optimismus: -1.413846
Kreuzvalidierung: -1.4115505

Verbesserter Bootstrap mit Optimismus

- Erzeuge B Trainingsmengen aus den Daten durch zufälliges Ziehen mit Zurücklegen
- Lerne ein Modell für jede Trainingsmenge
- Berechne Fehler von jedem Modell auf der Trainingsmenge und auf der Originalmenge
- Differenz beider Fehler ist der Optimismus
- Mittele den Optimismus über alle B Trainingsmengen
- Lerne ein Modell für die Originaldaten
- Berechne den Trainingsfehler für dieses Modell auf den Originaldaten
- Der Bootstrap-Fehler ist der Trainingsfehler plus mittlerer Optimismus

0.632 Bootstrap

- Die Wahrscheinlichkeit, daß eine Beobachtung in ein Bootstrap-Sample aufgenommen wird, ist

$$1 - \left(1 - \frac{1}{N}\right)^N \approx 1 - e^{-1} = 0.632 \tag{212}$$

- Erzeuge B Trainingsmengen aus den Daten durch zufälliges Ziehen mit Zurücklegen
- Sei C^{-n} die Indexmenge der Bootstrap-Samples, die Beobachtung x_n nicht enthalten
- Der Leave-One-Out-Bootstrap-Fehler ist

$$Err^{(1)} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{|C^{-n}|} \sum_{b \in C^{-n}} L(y_n, \hat{f}^b(x_n)) \tag{213}$$

- Sei Err der Trainingsfehler des auf der Originaldatenmenge trainierten Modells
- Der 0.623-Bootstrap-Fehler ist

$$Err^{0.632} = 0.368 \cdot Err + 0.632 \cdot Err^{(1)} \tag{214}$$

Zusammenfassung

- Für die meisten praktischen Anwendungen existiert schon ein Fehlermaß
- Kreuz-Validierung ist eine bewährte Methode Fehler realistisch zu schätzen
  - Es wird auch Standardabweichung mitgeschätzt
  - Der wahre Fehler wird meist etwas überschätzt
- Informationsmaße, die mit Zusatzinformations den Trainingsfehler korrigieren, sind nur in Spezialfällen einsetzbar
- Verbesserter Bootstrap und 0.632-Bootstrap korrigieren auch den Trainingsfehler, aber durch Datensimulation und sind deshalb generell einsetzbar.