Methoden des quantitativen Marketings MQM

Block A: (Analytisches) Customer Relationship Management (CRM)

- CRM ist **Kundenbeziehungsmanagement**, **Kundenpflege**. Bedeutet Gesamtheit der Strategien und Massnahmen zur **Herstellung** und **Aufrechterhaltung** der **Kundenbindung**. CRM umfassend **strategisch** Ansatz, zur vollständigen **Planung**, **Steuerung** und **Durchführung** aller **interaktiven Prozesse** mit **Kunden**.
- Es umfasst **gesamte Unternehmen** und **Kundenlebenszyklus** und beinhaltet **Database Marketing** und entsprechende **CRM-Software** als **Steuerungsinstrument**. Im Allgemeinen geht es darum,
  - durch **Analyse Kaufverhalten** und Einsatz Instrumente Marketing-Mix **Kundenzufriedenheit** und **Kauffrequenz** durch Up- und Cross-Selling zu **steigern**,
  - **Bindung Bestandskunden** mit massgeschneiderten Aktionen zu erhalten und
  - aus Interessenten Kunden machen, die sogenannte **Konversion**.
- Ziel CRM ist Lebenszeitwert jedes Kunden für Unternehmen zu maximieren und damit Rentabilität Unternehmens steigern. In dieser Hinsicht lassen sich CRM-Aktivitäten einer Kategorien zuordnen:
  - **Kundenakquisition (customer acquisition)** Prozess Gewinnung neuer Kunden, der grundlegende Schritt des gesamten CRM-Prozesses.
  - **Kundenbindung (customer retention)** Prozess Aufrechterhaltung und Entwicklung von Beziehungen zu bereits gewonnenen Kunden.
  - **Kundenabwanderung (customer churn/attrition)** Prozess Abwanderung bestehender Kunden aus Unternehmen zu steuern.
  - **Kundenrückgewinnung (customer win-back)** Prozess Rückgewinnung Kunden, die Unternehmen verlassen haben.
- CRM-System ist IT-gestützt zur Erfassung, Speicherung, Verarbeitung und Auswertung kundenbezogener Daten und zur (teil)automatisierten Unterstützung kommerziellen Prozesse aus allen Unternehmensbereichen - vorrangig Sales, Customer Service und Marketing. Grob in folgenden Komponenten aufteilen:
  - **Analytische Komponente:** Auswertung kundenbezogenen Daten
  - **Operative Komponente:** Verwendung kundenbezogener Daten
  - **Kommunikative Komponente:** Interaktion mit Kunden
  - **Kollaborative Komponente:** Bereichs- und Abteilungsübergreifenden Zusammenarbeit

Umsetzung CRM-Strategien aus **Modellierungsperspektive** sind folgende **Bestandteile zentral**:

- **Datenbanken:** Unterschied **transaktionsbezogener** und **kundenbezogener**. Können weitere Informationsquellen wie Resultate aus Studien (Marktbefragung) oder anderen externen Datenlieferanten nehmen.
- **Technologie:** ermöglichen **neuartige Kontakte** zu Kunden: E-Mail, Web-Seiten, Spracherkennung, soziale Netzwerke, mobile Online-Portale → viel mehr Daten, die komplexeres Verhalten festhalten können.
- **Metriken:** You cannot manage what you cannot measure. Metriken helfen Unternehmen, ihre **Performance zu verfolgen** und zu bewerten und insbesondere **Ergebnisse CRM-Aktivitäten zu bewerten**. Kann zwei Hauptgruppen unterscheiden, die einen messen auf **Markenebene** und anderen **Kundenebene**.
  - ➔ Wenn **Wert** auf **Kennzahlen** und Analytik legen, muss **Wert** auf **Qualität Rohdaten** legen. → **GIGO**!

Kundenstamm erweitern, um Einnahmen und Gewinne zu erhöhen funktioniert nicht immer:

- Z.B. reifen Märkten ist Kundenakquise ggf. nicht Schlüssel zu besseren Rentabilität
- Maximierung Rentabilität nicht zwingend durch Max. Schlüsselbereiche wie z.B. Kundenakquise/-bindung
  - ➔ **Customer-Lifetime-Value (CLV):** Besser geeignete Kennzahl für Ziel Maximierung der Rentabilität

Kundenakquise: auf Kunden zuschneiden, auf Massenebene, auf Segmentebene oder Einzelakquisition.

- **Traditionell:** Kundengewinnungsrate durch Massenebene, eher wünschenswerte als nachhaltige Kunden
- **Neue Tendenzen:** Ausrichtung auf Endverbraucher, ist Konsequenz aus Datenerfassung. Differenzierungen/Segmentierungen auf demografische, psychografische oder kaufkraftbezogene Merkmale → Analytik
  - ➔ Kunden sind unterschiedlich und deshalb auf Unterschiede eingehen, bis zum Extrem, dass man jeden unterschiedlich anspricht. Dies hat sich im E-Commerce und Online-Interaktionen durchgesetzt.
- Frage, welche Kunden halten sollen, ist schwierig zu beantworten. Kosten für Bindung Kunden ihre zukünftige Rentabilität übersteigen können und sie somit zu unrentablen Kunden machen.
- Frage, welchem Zeitpunkt welche Massnahme dazu beiträgt, Kunden zu halten, wird wichtiger. → CLV

Kundenabwanderung: Welche Kunden von Abwanderung abhalten, abhängig von Aufwand vs. Rentabilität
Kundenrückgewinnung: einfacher (+kostengünstiger) ehemalige Kunden wiederzugewinnen als neue Kunden

Kundenakquise

- Ist Grundlage gesamten CRM-Prozesses und Eckpfeiler in Geschäftsentwicklung.
- Ohne erfolgreiche Kundenakquise sind anderen Phasen im CRM-Prozess hinfällig.
- Gewinnung eines neuen Kunden ist mehrfach teurer als Bindung eines bestehenden Kunden.

Folgende Fragen, die im Prozess der Kundenakquise auftreten, sollten beantwortet werden können:

- Wie wahrscheinlich, dass potenzieller Kunden auf unsere Akquisitionsaktion ansprechen wird?
- Wie viele neue Kunden können wir mit dieser Aktion gewinnen?

- Wie viele Bestellungen wird jeder unserer neu gewonnenen Kunden tätigen?
- Wie beeinflussen Marketingvariablen wie Versandgebühr, Werbetiefe Antwortverhalten der Interessenten?
- Wie lange werden die neu gewonnenen Kunden bei unseren Unternehmen bleiben?
- Wie viel Gewinn / Wert wird diese Akquisitionskampagne für unsere Unternehmen bringen?
  - ➔ Antworten auf diese Fragen müssen dann integriert werden zu einer Gesamtantwort
- **Auf Kundenebene:** Lifetime Value (LTV) / Customer Lifetime Value (CLV),
- **Auf Markenebene:** Customer Equity (CE) d.h. der gesamte Lebenszeitwert aller Kunden der Marke
  - ➔ Bevor Fragen angehen, muss geklärt werden, ob Kundenkontakt auf (nicht)vertraglichen Basis beruht.
  - ➔ Oft bestimmt dies das zu verwendende statistische Modell, um Erkenntnisse aus Daten zu gewinnen.

Nichtvertragliche Basis: Üblich bei Einzelhandelsunternehmen und Katalogfirmen

- Kunden können Ausgaben frei auf mehrere Unternehmen aufteilen und gelegentlich oder häufig einkaufen.
- Aus Modellierungssicht wichtig ist, dass Unternehmen **nie weiss**, **wann** einer **kein Kunde mehr** ist!

Vertragliche Basis: Üblich z.B. bei Zeitungsabonnements und Telekommunikationsdienstleistungen

- Kunden zahlen i.d.R. Abonnement- oder Dienstleistungsgebühren monatlich oder jährlich.
- Folglich wissen Unternehmen, wann Kunden Beziehung beenden.
- Unternehmen haben Garantie für kontinuierlichen künftigen Cashflow über einen bestimmten Zeitraum.
- Dauer der Beziehung ist oft ein Schlüsselfaktor für zukünftige Rentabilität des Kunden.

Erstbestellmenge: geht davon aus, dass **anfängliche Auftragswert** wertvoller **Prädiktor** für zukünftigen Wert eines Kunden sein kann oder zumindest Geldbetrag für Kundenakquise rechtfertigt.

- Daher nützlich **Faktoren** verstehen, die **anfänglichen Auftragswert beeinflussen**, und
- damit wiederum den **erwarteten Erstbestellwert** eines jeden potenziellen Kunden zu **vorhersagen**.

Wie soll Erstbestellmenge modelliert werden?

- Y_i Erstbestellmenge Kunde i und A_i binäre Variable Akquisition erfolgreich war ($A_i = 1$) oder nicht ($A_i = 0$).
- Erwartete Erstbestellmenge $\mathbb{E}\langle Y_i\rangle$ mit bedingten Erwartungswerte berechnen: $\mathbb{E}\langle Y_i\rangle = \mathbb{E}\langle Y_i|A_i = 0\rangle \cdot P\langle A_i = 0\rangle + \mathbb{E}\langle Y_i|A_i = 1\rangle \cdot P\langle A_i = 1\rangle = 0 \cdot P\langle A_i = 0\rangle + \mathbb{E}\langle Y_i|A_i = 1\rangle \cdot P\langle A_i = 1\rangle = \mathbb{E}\langle Y_i|A_i = 1\rangle \cdot P\langle A_i = 1\rangle$
- modellieren beide Teile separat, z.B. indem $P\langle A_i = 1\rangle$ mit logistischen Regressionsmodell aus und
- $\mathbb{E}\langle Y_i|A_i = 1\rangle$ mit Gamma-Regression und Log-Link modelliert werden.
- Genauigkeit Schätzung wird oft mit **«mean absolute percentage error» (MAPE)** oder allenfalls mit «median absolute percent error» (MeAPE) bestimmt. Bestimmung von MAPE kann auf zwei Arten erfolgen
  - Ökonomen: $MAPE := \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{Y}_i - Y_i|}{Y_i}$
  - Alternative (Prognose): $MAPE2 := \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{\hat{Y}_i}$

Kundenbindung

- Kundenbindungsstrategien sind im vertraglichen und ausservertraglichen Bereich nötig. 2 Ausrichtungen:
  - Schwerpunkt **Auswirkung** versch. **Marketing-/Unternehmenscharakteristiken** auf Kundenbindung
  - Fokus ökonometrische und statistische Modelle zur **Schätzung** Entscheidung, ob Kunde bleibt oder nicht
- Um Beziehungen im Bereich der Kundenbindung zu untersuchen, werden Methoden aus **konfirmatorischen Faktorenanalyse** (Teilgebiet multivariaten Statistik) verwendet. Für Aufbau Modell solide empirische/konzeptionelle Grundlagen erforderlich → Strukturgleichungsmodell. Werden nicht weiter verfolgen.
- Analytisch einfacher sind Treiber für Entscheidung, ob Kunde bleibt oder nicht. Häufig gestellte Fragen:
  - Wird Kunde Zukunft wieder kaufen?
  - Wie lange wird Lebensdauer Kunden sein?
  - Angenommen, der Kunde wird wieder kaufen: Wie viele Artikel wird dieser Kunde kaufen? Wie viel wird dieser Kunde wahrscheinlich ausgeben? Wird dieser Kunde in mehreren Produktkategorien kaufen?
  - Kauft Kunde bei mir (hoher Share-of-Wallet) oder von vielen Unternehmen (niedriger Share-of-Wallet)?
  - Welche langfristigen Auswirkungen hat Kaufverhalten dieses Kunden auf den Unternehmenswert?
- Vertragliche oder nicht vertragliche Basis bestimmt auch hier oft Art des zu verwendenden Modells.

Wiedereinkauf oder nicht (bleiben oder gehen)

- Wiedereinkauf oder nicht wird normalerweise als binäres Ereignis modelliert, bei dem entweder ein Wiedereinkauf stattfindet (1) oder nicht (0). → binäre GLM oder andere Klassifikationsmethoden
- Traditionell werden als erklärende Variablen die **RFM-Variablen (Recency (Aktualität), Frequency (Häufigkeit) und Monetary Value (monetäre Wert))** verwendet:
  - Recency Zeit zur letzten Transaktion des Kunde
  - Frequency wie oft Kunde bei Firma kauft
  - Monetary Value durchschnittlichen Ausgaben eines Kunden pro Transaktion

Kundenverweildauer

- Schätzung Kundenverweildauer ist für Berechnung CLV wesentlich. Wichtig bei Modellierung Kundenverweildauer ist Zeitpunkt, wann Kundenbeziehung beendet wurde: eigentlich **undefiniert** im **nichtvertraglichen** Umfeld oder vermeintlich klar **definiert** im **vertraglichen** Umfeld
- **Vertraglichen Umfeld**: Verweildauern mit statistischen Wartezeitmodellen modellieren. Falls Angaben zu (stark) **gerundet** → **diskreten Ansätze** verwenden. Zeitangaben gerundet **kaum gleiche Wartezeiten** → **Cox** proportionale Hazardmodelle. **Rundungen** → **Weibullregressionsmodell** oder anderes AFT-Modell.

- **Nichtvertraglichen Umfeld:** Wahrscheinlichkeit schätzen, dass Kunde angesichts bisherigen Kaufverhaltens auch in nächsten Zeitperiode aktiv ist. Bei stark diskretisierten Zeiterfassungen → Markov-Ketten

Bestellmenge: gleich wie bei Erstbestellmenge, datengetriebenes Modell für Bestellmenge von jedem Kunden
Rentabilität (CLV): kann aus bekannten Bestellmengen und weiteren Kundeneigenschaften CLV berechnen

Um richtig Produkt für **Up-Sell** finden → Next-Best-Offer-Ansätze und Empfehlungssysteme

- Welches Produkt wird nächstes gekauft?
- Welcher Kanal wird nächstes genutzt?
- Wann kauft Kunde nächste Mal ein?
- Zu welchem Preis kauft Kunde Produkt?

Empfehlungssysteme bestimmen zu aktuell genutzten/gekauften Objekten passende, ähnliche Objekte. Nutzt aktuell & historisch Nutzerverhalten. Nutzen Interesse & Nutzungsverhalten bereits bewerteten Objekts. Info implizit (reine Interaktion, Ansicht/ Kauf) oder explizit (konkrete Bewertung) für Berechnung Empfehlung.

Kundenabwanderung (Customer Churn)

- Kundenabwanderung kann für Firma, die nicht wissen, wann Kunde Beziehung beendet oder welche Warnsignale es gibt, warum Kunde wahrscheinlich kündigt, äusserst kostspielig sein. Zwei Fragen im Fokus:
- Gründe für Churn? • Wenn Kunde Firma noch nicht verlassen hat, wann wird er die Beziehung beenden?

Kundenabwanderung im B2C-Bereich gibt es aus diversen Gründen:

- Tod Kunde oder Firmenkonkurs
- Zahlungsunfähig
- Wegzug Kunde
- Heirat oder Scheidung
- Wechsel zur Konkurrenz

Da Kunden Churn Gründe oft nicht offenlegen, ist datengestützte Vorhersage schwierig bis unmöglich.

Kundenrückgewinnung (Customer Win-Back)

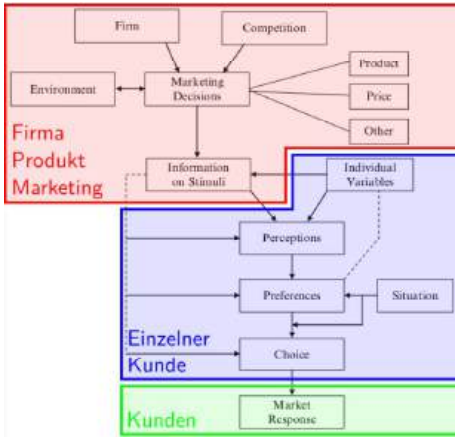
- Abgewanderte Kunden müssen nicht für immer verloren sein. Man kann sie auch wieder zurückgewinnen.
- Im Gegensatz zu neuen Kunden kennen abgewanderte Kunden Produkte. Kann einfacher sein, verlorene Kunden anzusprechen. Unternehmen weiss einiges über Kunden, weshalb zielgerichteter ansprechen.
- Kunden wechseln oft, weil mit Produkt unzufrieden → schwierig Einstellung ändern / sie wiederkommen
- Nicht sinnvoll alle abgewanderten Kunden zurückholen, da allenfalls mehr Aufwand als Ertrag.

Block B: Market experiments

Marketingentscheidungen und die Rolle der Konsumentenwahl

- **Interdependente** Entscheidung bei Marketingstrategien für ein Produkt: Produktattribute, Positionierung (z.B. Premium, Budgetangebot), Kommunikation, Vertrieb, Preisgestaltung bezüglich Zielkonsumenten.
- Diese Entscheidungen müssen im Rahmen unsicherer wettbewerblicher Reaktionen und einer ständig ändernden und unvorhersehbarer Umgebung getroffen werden.
- Verständnis, wie Kunden auf Produkt ansprechen und sie zwischen konkurrierenden Produkten auswählen, wesentlich zu erfolgreichen Marketingentscheidungen beitragen kann.
- Conjoint-Analysen bietet statistische Methoden, mit welchen Entscheidungsprozesse analysieren. Vorausgesetzt, dass Produkte mittels Attributprofilen beschrieben werden können und Konsumentenwahl wesentlich von Abwägungen zwischen Produkteattributen abhängt. Conjoint-Analysen Marketingentscheidungen:
  - Optimales Design neuer Produkte → Welches sind die optimalen Produktattribute?
  - Wahl des Absatzmarkts → Welches sind die optimalen Zielkonsumenten?
  - Preisgestaltung bei neuen Produkten
  - Wettbewerbsreaktionen

Framework für Verständnis Konsumentenwahl



Teilnutzenfunktionen

- Kundenspezifische Funktion, welche Gesamtbewertung Produkt in Beiträge einzelnen Attribute zerlegt.

Beispiel: Laufschuhe: Gesamtbewertung Y: 4 = sehr gut, 3 = gut, 2 = schlecht, 1 = sehr schlecht

- Attribut 1: Preis in Franken v_p
- Attribut 2: Qualität v_q: schlecht 1, mittel 2 und hoch 3

Modellspezifikationen: ε_i eine Zufallsfehler

- additiv: $Y_i = \beta_{i0} + \beta_{i1} v_p + \beta_{i2} v_q + \varepsilon_i$
- multiplikativ: $Y_i = \beta_{i0} + \beta_{i1} v_p + \beta_{i2} v_q + \beta_{i3} v_p v_q + \varepsilon i$
- Teilnutzenfunktionen für Attribut $v_p = \beta_{i1} v_p$
- Gesamtbewertung und Attribute können nominal-, ordinal- oder intervallskaliert sein
- Koeffizienten werden (oft) einzeln pro Probanden (Kunde) geschätzt (Index i bei Koeffizienten β). D.h. separate Modelle für die Probanden.
- Pro Untersuchungseinheit liegen mehrere Beobachtungen vor. Z.B. jeder Proband bewertet 5 Turnschuhen

Typen von Conjoint-Analysen

- **(Rating-Based) Conjoint Analysis (CA)**
- Ranking-Based Conjoint Analysis (RBCA)
- Choice-Based Conjoint Analysis (CBCA)
- Adaptive Conjoint Analysis (ACA)
- Self-explicated Conjoint Analysis

(Rating-Based) Conjoint Analysis (CA):

- Probanden bewerten Profile mehrerer hypothetischer Produkte: Z.B. Rennschuhe: 120.- & schlechte Qualität, 160.- & gute Qualität etc. auf einer Rating Skala 1-100
- Urspr. Ansatz: Alle möglichen Profile mit allen Attributen abfragen (Full-Factorial-Design für Full-Profiles)
- Praxis: Spezifische Untermenge von Full-Profiles abfragen (z.B. mit Fractional-Factorial-Designs)
- Schätzung von Teilnutzenfunktionen meist mit linearen Regressionsmodellen

Ranking-Based Conjoint Analysis (RBCA): Ähnlich wie Rating-Based, mit Unterschied, dass vorgelegten Produkte in Reihenfolge gebracht werden. Analysiert wird oft Score = Anzahl vorgelegte Produkte - Rang + 1.

Choice-Based Conjoint Analysis (CBCA oder CBC):

- Probanden wählen wiederholt aus Auswahl vorgelegter hypothetischer Produkte (z.B. Nike vs. Adidas, dann Nike vs. Asics etc.). Wahrscheinlich die am meisten verwendete Methode heutzutage
- Oft werden nicht alle Entscheidungssituationen durchgespielt, sondern Untermenge davon
- Schätzung von Teilnutzenfunktionen meist mit logistischen (multinomialen) Regressionsmodellen

Adaptive Conjoint Analysis (ACA): Hybrider Ansatz

1. Self-Explicated Task: Probanden bewerten Wichtigkeit der Attribute und Präferenzen für Attributeniveaus
2. Rating: Probanden bewerten Profile, basierend auf Self-Explicated Task massgeschneidert für Probanden ausgewählt wurden (Partial Profiles)

- Kompositionale Methode: Probanden bewerten relative Wichtigkeit Attribute und Attraktivität jedes Attributniveaus. Präferenz für jedes Produkt wird dann als gewichtete Summe Attraktivitätsbewertungen berechnet.

Anwendungsbereiche

- Verbrauch: Seifen, Haarshampoos, Benzinpreis
- Industrier: Kopierer, Datenübertragungen, Clouds
- Finanzen: Versicherungspolicen, Rabattkarten
- Verkehr: Flug- und Bahnverkehrangebote
- **Vorteil Conjoint-Analysen**: Erlaubt Antworten zu What-If Fragen basierend auf hypothetischen Produkten. Z.B. kann simuliert werden, welchen Marktanteil der Verkauf eines neuen Produkts einnehmen wird.

Begriffe

- **Charakteristik:** Objektive Eigenschaften des Produkts, z.B. Zuckermenge in Schokolade
  Attribut: Subjektiv wahrgenommene Eigenschaft, z.B. empfundene Süsse der Schokolade
- **Teilnutzenfunktionen:** Additive Beitragsfunktionen zur Gesamtbewertung der einzelnen Charakteristiken/Attribute, z.B. $U_{i1}(x_1) = \beta_{i1} x_1$
- **Utility-Funktion/Nutzenfunktion:** Zusammenstellung aller Teilnutzenfunktionen, z.B. $U_{i1}(x_1) + U_{i2}(x_2)$
- **Bemerkung**: Da dieses Modul eher methodischen Zugang hat, wird nicht streng zwischen Charakteristiken und Attributen unterschieden. Oft nur von Attributen geredet, und man meint Attribute und Charakteristiken.

Illustration Rating-Based Conjoint-Analyse (CA)

ÖV-Nutzung in Ithaca, NY, USA. Attribute (aus Voranalysen):

- Fahrpreis: \$1.55, 1.85, 2.15: Einfluss auf Nachfrage/Umsatz
- Wartezeit: 10/20/30 min: Einfluss auf Frequenz/Transportmittelgrösse
- Probanden sollen Kombinationen der Attributwerte mit Zahl zwischen 0: würde Option niemals wählen und 100: würde Option sicherlich wählen bewerten, Full-Factorial-Design ergeben sich 3 × 3 Kombinationen

		Wartezeit		
		10 min	20 min	30 min
Preis	\$1.55	100	95	80
	\$1.85	92	85	60
	\$2.15	75	70	50

Modell für Proband i: $Y_{ij} = \beta_{i0} + U_{i1}(x_{1j}) + U_{i2}(x_{2j}) + \varepsilon_{ij}, j = 1, \ldots, 9$

- Y_{ij} Bewertung von Proband i für Profil j, εij Zufallsfehler
- x_{1j} Niveau j-ten Profils bei Fahrpreis, x_{2j} Niveau j-ten Prols bei Wartezeit
- $U_{i1}(\cdot)$ Teilnutzenfunktion für Fahrpreis von Proband i, $U_{i2}(\cdot)$ Teilnutzenfunktion Wartezeit von Proband i
- Dummy Variable: $X_{11} = \begin{cases} 1 \; falls \; Fahrpreis = 1.55 \\ 0 \; sonst \end{cases} \ldots$, $X_{21} = \begin{cases} 1 \; falls \; Wartezeit = 10 \\ 0 \; sonst \end{cases}$, …. Lassen Dummy-Variablen für letzten Niveaus (Fahrpreis \$2.15, Wartezeit 30 min) zwecks Modellidentifikation weg.

Schätze lin. Regressionsmodell: $Y_i = \beta_{i0} + \beta_{i11} x_{11} + \beta_{i12} x_{12} + \beta_{i21} x_{21} + \beta_{i22} x_{22} + \varepsilon_{ij}$, wobei $U_{i1}(x_1) = \beta_{i11} x_{11} + \beta_{i12} x_{12}$, $U_{i2}(x_2) = \beta_{i21} x_{21} + \beta_{i22} x_{22}$. m <- lm(Rating ~ Fare.1 + Fare.2 + Wait.1 + Wait.2, data = d)

Geschätzte Teilnutzenfunktionen: Teilnutzenfunktionen = Dummy-Effekt

- $U_{i1}(x_1) = 26.7\, x_{11} + 14\, x_{12}$, $U_{i2}(x_2) = 25.7\, x_{21} + 20\, x_{22}$

	Estimate	Std. Error
(Intercept)	49.78	2.41
Fare.1	26.67	2.64
Fare.2	14.00	2.64
Wait.1	25.67	2.64
Wait.2	20.00	2.64

Trade-Offs zwischen Attributen, Ausgangspunkt: Preis = 1.85, Wartezeit = 20 min

- Verkürzt Wartezeit 20 auf 10, steigt Nutzen System um $25.7 - 20.0 = 5.7$ Punkte
- Um gleiche Nutzenänderung mit Preis erzielen, müsste Preis um $5.7/(26.7 - 14.0) = 0.135$ Dollar gesenkt werden, d.h. auf $1.85 - 0.13 = 1.72$
- **Achtung**: Kurven sind nichtlinear und darum **abhängig vom Ausgangspunkt**!

- Normalisierung der Teilnutzenfunktionen: Spannweite der Teilnutzenfunktion = 1

Relative Importance RIMP: Wichtigkeit Attribute anhand Spannweite der Teilnutzenfunktionen:

- $RIMP_i(x_r) = \frac{\max \beta_{irc} - \min \beta_{irc}}{\sum_{r'=1}^{R} \max \beta_{ir'c} - \min \beta_{ir'c}}$, $\sum_{r=1}^{R} RIMP_i(x_r) = 1$
- Bsp. ÖV: $RIMP_i(x_1) = \frac{26.7}{(26.7-0)+(25.7-0)} = 0.51$, $RIMP_i(x_1) = \frac{25.7}{(26.7-0)+(25.7-0)} = 0.49 = 1 - RIMP_i(x_1)$

Vorhersagen: Einsetzen in Modell oder Interpolation

- Bsp. ÖV-Nutzung: Bewertung für System mit Fahrpreis $1.70 und 15 min Wartezeit
- $\hat{y} = 49.8 + \frac{26.7+14}{2} + \frac{25.7+20}{2} = 93$ → Achtung bei Interpolationen!

Schritte von Conjoint Analysen

- **Problemdefinition:** Problemdefinition und geplante Nutzung der Ergebnisse, Auswahl Attributen/Stufen
- **Erstellung Fragebogen und Profilkarten**: Erstellung des orthogonalen Designs
- **Durchführung der Umfrage**
- **Analyse**: Schätzung von Attributnutzenwerte und Attributbedeutungen
- **Segmentierung**: Zuordnung von Attributnutzenwerte-Clustern zu Hintergrunddaten, Basisfällen, Hintergrundvariablen. Vorbereitung von Dateien für Simulationen/Optimierungen mit geeigneten Eingabedateien
- **Simulationen und Optimierungen**: Simulation und Sensitivitätsanalyse, Weitere Analysen, z.B. Optimierung einzelner Produkte oder Produktlinien
- **Report**: Vorbereitung Bericht, Präsentation und leave-behind Simulator/Optimierer mit Eingabedateien

Grundlegendes Modell bei Conjoint-Analysen

Postulat, dass Nutzwert (Utility) Multi-Attribut Produkts in spezifische Beiträge der einzelnen Attribute zerlegt werden kann: $g(E(Y_i)) = \beta_{i0} + U_{i1}(x_1) + U_{i2}(x_2) + \cdots + U_{i12}(x_1, x_2) + \cdots$ mit

- Y_i Bewertung oder Wahl Proband i
- $x_1, x_2, \ldots$ Attribute
- $g(\cdot)$ Link-Funktion: Ratings: $g(\eta) = \eta$ und Choice: $g(\eta) = \frac{1}{1+e^{-\eta}}$
- $U_{i1}(x_1)$ Teilnutzenfunktion für x_1 isoliert
- $U_{i12}(x_1, x_2)$ Teilnutzenfunktion für x_1, x_2 gemeinsam
- Bewertung Y_i wird meist nur für **Untermenge** von Profilen (=Kombinationen von Attributwerten) abgefragt
- Dadurch können nicht mehr alle Teilnutzenfunktionen geschätzt werden (Interaktionen höherer Ordnung)
- Wahl Profiluntermenge Methode Versuchsplanung (Experimental Design): Fractional-Factorial-Design, Block-Design. Modernere Ansätze wie Adaptive Conjoint Analysis geht Profiluntermengenwahl anders vor

Attributstypen und Teilnutzenfunktionen

- **Kategorielle Attribute** (nominal- oder ordinal-skaliert)
  - $x \in \{1, 2, \ldots, C\}$, Abstände zwischen Kategorien nicht definiert
  - Farbe (rot, grün) → nominal
  - Fussabdruck: hoch, mittel, tief → ordinal
  - Teilnutzenfunktionen: $U_i(x) = \sum_{c=1}^{C-1} \beta_{ji1}\,(x = c)$ Dummy Coding
- **Numerisch** (intervall- oder ratio-skaliert)
  - $x \in \mathbb{R}$, Abstände definiert
  - Temperatur Celsius → intervall
  - Temperatur Kelvin → ratio
  - Teilnutzenfunktionen: $U_i(x) = \beta_i x$ (Linear) oder $U_i(x) = \beta_i (x - x_0)^2$

Wahl der Attribute und Attributstypen: Themenspezifisch, Umfrage durchführen, Experten befragen, Fokus aus praktischen Gründen oft auf wesentlichste Attribute, Pilotstudie durchführen, Anzahl Niveaus klein halten

Konstruktion von Stimulus-Sets (Profile)

- Basierend auf Attributen können Profile konstruiert werden. Bsp.: Smartphone
  - Gesamtbewertung Y: 0-100 Skala
  - Marke (Samsung, Apple, Google, Huawei)
  - Kameraqualität (8MP, 10MP, 12MP, 14MP)
  - Akkulaufzeit (12h, 16h, 20h, 24h)
  - Vollständiges Profil: Samsung/12h/200g/12MP
  - Gewicht (140g, 160g, 180g, 200g)
  - Partielles Profil: z.B. Samsung/12h/12MP
  - Wie viele vollständige Profile gibt es insgesamt? $4 \cdot 4 \cdot 4 \cdot 4 = 256$

Welche der möglichen Profile müssen erfasst werden? Teilweise gegeben durch:

- Conjoint-Analysen-Ansatz:
  - Full Profiles: Rating-/Choice Based Analysis
  - Partial Profiles: Adaptive Conjoint Analysis
- Erhebungsart (persönliches Interview, telefonisch)
- Hypothetische $U_i(x)$ (Interaktionen? quadratische Funktion?)
- Regeln:
  - $\frac{Anzahl\ Profile}{Anzahl\ Parameter}$ möglichst gross
  - Anzahl Profile sollte Probanden nicht ermüden
  - $\mathbb{E}\left(MSE(\hat{Y})\right) = \left(1 + \frac{Anz\ Profile}{Anz\ Parameter}\right)\sigma_\epsilon^2$ möglichst klein, Faustregel $Anz\ Profil \approx 2$ oder $3\ Anz\ Parameter$
  - Gezeigte Profile sollten plausibel sein: Z.B. Smartphone mit besten Features und günstigstes weglassen

Designs/Methoden zur Wahl von Profilen

Full-Factorial-Design, Fractional-Factorial-Design (orthogonale), unvollständig Block-Design, Random Sampling

Full-Factorial-Designs: Bsp.: Beurteilung von Marketing-Forschungsgesuchen Rao. Attribute:

- Kosten: $55'000, $70'000, $85'000
- Reputation: Etabliert, Neu
- Anzahl Profile: $3 \cdot 2 \cdot 2 \cdot 2 = 24$
- Zeit: 2 Monate, 4 Monate
- Methodik: Standard, Fortschrittlich

Full-Factorial-Design

	Kosten	Reputation	Zeit	Methodik
1	55'000	Etabliert	2	Standard
2	70'000	Etabliert	2	Standard
3	85'000	Etabliert	2	Standard
4	55'000	Neu	2	Standard
5	70'000	Neu	2	Standard
6	85'000	Neu	2	Standard
7	55'000	Etabliert	4	Standard
8	70'000	Etabliert	4	Standard
9	85'000	Etabliert	4	Standard
10	55'000	Neu	4	Standard
11	70'000	Neu	4	Standard
12	85'000	Neu	4	Standard
13	55'000	Etabliert	2	Fortschrittlich
14	70'000	Etabliert	2	Fortschrittlich
15	85'000	Etabliert	2	Fortschrittlich
16	55'000	Neu	2	Fortschrittlich
17	70'000	Neu	2	Fortschrittlich
18	85'000	Neu	2	Fortschrittlich
19	55'000	Etabliert	4	Fortschrittlich
20	70'000	Etabliert	4	Fortschrittlich
21	85'000	Etabliert	4	Fortschrittlich
22	55'000	Neu	4	Fortschrittlich
23	70'000	Neu	4	Fortschrittlich
24	85'000	Neu	4	Fortschrittlich

Bei Fractional-Factorial-Designs wird nur Bruchteil des Full-Factorial-Designs verwendet. Bsp.: 1/2 Factorial-Design (24/2 = 12 Profile) + 2 = 14, 2 zusätzliche Profile zur Validierung (rot)

Asym orthogonal:

	Kosten	Reputation	Zeit	Methodik
1	85'000$	Etabliert	2	Fortschrittlich
2	55'000$	Neu	4	Fortschrittlich
3	70'000$	Neu	4	Standard
4	70'000$	Etabliert	2	Fortschrittlich
5	70'000$	Etabliert	4	Fortschrittlich
6	85'000$	Neu	4	Fortschrittlich
7	55'000$	Etabliert	4	Standard
8	85'000$	Etabliert	4	Standard
9	55'000$	Neu	2	Fortschrittlich
10	85'000$	Neu	2	Standard
11	55'000$	Etabliert	2	Standard
12	70'000$	Neu	2	Standard

Fractional-Factorial-Design

	Kosten	Reputation	Zeit	Methodik
1	55$'000	Etabliert	2	Standard
2	570'000	Etabliert	2	Standard
3	585'000	Etabliert	2	Standard
10	555'000	Neu	4	Standard
11	570'000	Neu	4	Standard
12	585'000	Neu	4	Standard
14	570'000	Etabliert	2	Fortschrittlich
16	555'000	Neu	2	Fortschrittlich
17	570'000	Neu	2	Fortschrittlich
18	585'000	Neu	2	Fortschrittlich
19	555'000	Etabliert	4	Fortschrittlich
20	570'000	Etabliert	4	Fortschrittlich
21	585'000	Etabliert	4	Fortschrittlich
24	585'000	Neu	4	Fortschrittlich

Orthogonale Designs für Haupteffekte

- Orthogonale Designs = Sonderform von Fractional Designs
- Kleinste Anzahl Profile, sodass gerade **noch Haupteffekte schätzbar** sind:
  - Möglich $(x_1, x_2 \in \mathbb{R})$: $Y_i = \beta_{i0} + \beta_{i1}x_1 + \beta_{i2}x_2 + \varepsilon_{ij}$
  - Nicht möglich $(x_1, x_2 \in \mathbb{R})$: $Y_i = \beta_{i0} + \beta_{i1}x_1 + \beta_{i2}x_2 + \beta_{i3}x_1x_2 + \varepsilon_{ij}$
- Symmetrische orthogonale Designs: Alle Attribute haben die gleiche Anzahl Niveaus. Beispiel Marketing-Forschungsgesuche ist asymmetrisch, da Attribut Kosten 3 Attributwerte aufweist, die anderen nur 2
- Bei (sym. und asym.) orthogonalen Designs sind Kombinationen von Niveaus von 2 Attributen gleich häufig

Methoden zur Datensammlung: Vollständige Profile

- Beschreibung mit ALLEN Attributen
- Beispiel: Kosten = $55'000, Reputation = Etabliert, Zeit = 2, Methodik = Standard → Proband bewertet dann
- Darstellung wird Probanden gezeigt, und anschliessend wird bewertet
- Einfach, aber nicht immer praktikabel bei vielen Attributen bzw. Niveaus

Methoden zur Datensammlung: Trade-Off Matrix

- Methode falls viele Attribute/Niveaus, nur 2 Attribute werden gegenübergestellt
- Weitere Attribute sollen in Gedanken «fixiert» werden
- Einfach für Probanden, aber schwierig zu implementieren bei Auswertungen; heute nicht mehr verwendet

Methoden zur Datensammlung: Paarweise Vergleiche

- Es werden 2 Profile (Produkte) angezeigt, entweder vollständig mit allen Attributen oder partiell
- Beispiel partielles Profil: Welches Profil präferierst du? Vierfache Geschwindigkeit vom IBM PC und sechs Stunden Batterie oder zweifache Geschwindigkeit vom IBM PC und 12 Stunden Batterie

Stimuluspräsentation – Grundmethoden:

- Verbale Beschreibung: Einfach bewerkstelligen, günstig. Schwierigkeit Nahrungsprodukt/Verständlichkeit
- Bildliche Darstellungen: Ansprechend, weniger Information
- Echte Produkte oder Prototypen davon zeigen: Goldstandard, jedoch oft nicht umsetzbar, teuer

Reliabilität und Validität

- Interne Reliabilität: Korrelation bei (Re)test ca. 0.85, Datensammlung kann Einfluss haben, paarweise Vergleiche schneiden am besten ab
- Interne prädiktive Validität: Vorhersagen korrelieren mit ca. 0.75 mit weggelassenen Beobachtungen
- Externe Validität (i.e. verhalten sich Probanden in der Realität tatsächlich so?) ist schwer messbar

Vorgehen beim Design von Conjoint Studien

1. Attribute/Niveaus wählen (kategoriell, numerisch)
2. Teilnutzenfunktionen (Dummy Regression, Vector, Ideal-Point)
3. Stimulus-Sets (Full Factorial, orthogonal)
4. Datensammlung (Vollständige Profile, Paare)
5. Stimuluspräsentation (verbal, Bilder, Produkte)

Analyse und Verwendung von Conjoint-Studien

Studiendesign «diktiert» Auswertung grösstenteils! Wichtigste Faktoren zur Wahl Auswertungsmethode:

- Ausprägungen des Ratings
  - Numerisch: Lineare Regression
  - Kategoriell: Logit-Modelle
- Gewünschtes Aggregationsniveau
  - Individualniveau (Standard) vs. Untergruppen, Stichprobe
  - Aggregation: entweder bei Analyse (alle Daten in ein Modell) oder bei Resultaten (Einzelmodelle für Probanden zusammenfassen)
  - Achtung: Inhomogenitäten zwischen Individuen berücksichtigen!
- Ziel ist oft, Modell für Vorhersage neuer Probanden zu verwenden, Marktanalysen

Analysemodell für Ratings – Additives Modell für 1 Proband:

$Y_{ij} = \beta_{i0} + U_{i1}(x_{1j}) + U_{i2}(x_{2j}) + \cdots + +U_{iR}(x_{Rj}) + \epsilon_{ij}, j = 1, \dots, J$

- Y_{ij} Bewertung von Proband i für Profil j
- x_{rj} Niveau des j-ten Profils bei Attribut $X_r, r = 1, \dots, R$
- ϵ_{ij} Zufallsfehler
- $U_{iR}(.)$ Teilnutzenfunktion für Attribut X_r von Proband i

Teilnutzenfunktionen:

- Kategorielle Attribute: $x_r \in \{1, 2, \dots, C_r\}$, $U_{ir}(x_{rj}) = \sum_{c=1}^{C_r-1} \beta_{irc} 1(x_{rj} = c) = \sum_{c=1}^{C_r-1} \beta_{irc} x_{rcj}$ (Dummy-Coding)
- Numerische Attribute: $x_r \in \mathbb{R}$, $U_{ir}(x_{rj}) = \beta_{ir} x_{rj}$ (linear) oder $U_{ir}(x_{rj}) = \beta_{ir}(x_{rj} - x_{r0})^2$ (quadratisch)

Modell mit Interaktionen: $Y_{ij} = \beta_{i0} + U_{i1}(x_{1j}) + U_{i2}(x_{2j}) + U_{i3}(x_{1j}, x_{2j}) + \epsilon_{ij}$

- X_1, X_2 numerisch: $U_{i3}(x_{1j}, x_{2j}) = \beta_{i3} x_{1j} x_{2j}$
- X_1 numerisch, X_2 kategoriell: $U_{i3}(x_{1j}, x_{2j}) = \sum_{c=1}^{C_2-1} \beta_{i3c} x_{1j} x_{2cj}$
- X_1, X_2 kategoriell: $U_{i3}(x_{1j}, x_{2j}) = \sum_{c=1}^{C_1-1} \sum_{c'=1}^{C_2-1} \beta_{i3cc'} x_{1cj} x_{2c'j}$

- Modell mit Interaktionen ist flexibler, aber auch komplexer – Mehr Modellparameter β
- Mit Full-Factorials können alle möglichen Interaktionen berücksichtigt werden
- Mit orthogonalen Designs für Haupteffekte sind Interaktionen nicht möglich

Modellselektion – Wie wählt man das beste Modell? Gütemasse zur Vorhersagekraft:

- **Gütemass 1: Testfehler**
  - Bei Modellanpassung Beobachtung weglassen (z.B. 2 pro Proband)
  - Berechne Anteil Nichtübereinstimmungen zw. weggelassenen Beob. und dazugehörig Modellvorhersage
- **Gütemass 2:** Expected Mean Squared Error of Prediction
  - $\widehat{EMSEP}_{\mathcal{M}} = \left(\tilde{R}^2_{\mathcal{M}_g} - \tilde{R}^2_{\mathcal{M}}\right) + \left(1 - \tilde{R}^2_{\mathcal{M}_g}\right) \cdot \left(1 + \frac{k}{n}\right)$
  - $\tilde{R}^2_{\mathcal{M}_g}$: korr. R^2 des grössten Modells $\mathcal{M}_g$
  - $\tilde{R}^2_{\mathcal{M}}$: korr. R^2 des betrachteten Modells $\mathcal{M}$
  - k = Parameteranz. Modell m, n = Anzahl Profile

➔ Wähle Modell, welches gewählte Gütemass minimiert

Analyseebenen

I. Analyse auf Probandenebene: Bisheriger Ansatz, i in Gleichungen ref. auf Probanden
II. Poolen von Subgruppen: i referenziert auf vordefinierte oder empirisch ermittelte Subgruppen
III. Poolen der gesamten Stichprobe: i referenziert auf gesamte Stichprobe

Für jede Analyseebene ...**Fixed-Effects Ansatz**: IA, IIA und IIIA oder **Hierarchischer Ansatz**: IB, IIB und IIIB

Ansätze IA und IIIA

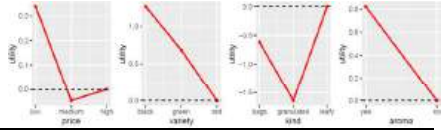
- Ansatz IA: Schätze jeden n Probanden ein separates lineares Regressionsmodell:
  - $y_1 = \beta_{10} + X_1\beta_1 + \epsilon_1$
  - $y_2 = \beta_{20} + X_2\beta_2 + \epsilon_2$
  - $y_n = \beta_{n0} + X_n\beta_n + \epsilon_n$
  - mit $y_i, \epsilon_i \in \mathbb{R}^J, \beta_i \in \mathbb{R}^p$ und $X_i \in \mathbb{R}^{J \times p}$ für $i = 1, 2, \dots, n$
- Ansatz IIIA: Schätze 1 lineares Regressionsmodell ohne Differenzierung zwischen den Probanden:
  - $y = X\beta + \epsilon$
  - mit $y, \epsilon \in \mathbb{R}^{n \cdot J}, \beta \in \mathbb{R}^p$ und $X \in \mathbb{R}^{n \cdot J \times p}$

Beispiel Tea Daten – Vier Attribute ➔ $3 \cdot 3 \cdot 3 \cdot 2 = 54$ möglichen Profile

- price: low, medium, high
- variety: black, green, red
- kind: bags, granulated, leafy
- aroma: yes, no
- Alle Attribute sind kategoriell → Dummy-Variablen
- Additive Modell: $rating_{ij} = \beta_{i0} + \beta_{i11} pricelow_j + \beta_{i12} pricemedium_j + \beta_{i21} varietyblack_j + \beta_{i22} varietygreen_j + \beta_{i31} kindbags_j + \beta_{i32} kindgranulated_j + \beta_{i41} aromayes_j + \varepsilon_{ij}$
- Ansatz IA: 100 Modelle, bzw. 800 Parameter
- Ansatz IIIA: 1 Modell, bzw. 8 Parameter

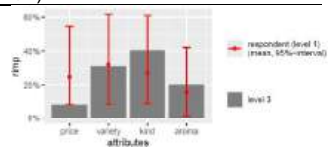
Geschätzte Teilnutzenfunktionen (Partworth Utilities):

- Nutzen bei price: low > high > medium
- Nutzen bei variety: black > green > red
- Nutzen bei kind: leafy > bags > granulated
- Nutzen bei aroma: yes > no



Relative Importance RIMP für IA:

- Mittelwert von $RIMP_i(x_r)$ über i ist nicht gleich dem $RIMP(x_r)$ des gepoolten Modells! Indikator dafür, dass gepoolte Modell nicht passt!



Poolen von Subgruppen

- Was, wenn Probandenebene zu fein (zu viele Parameter) und Stichprobenebene zu grob scheint?
- Mittelweg: Analyse nach Gruppe:
  - Vordefiniert (Supervised): Altersgruppe, Geschlecht
  - Empirisch (Unsupervised): Clusteranalyse
- Empirisches bilden von Subgruppen: Jeden Probanden $i = 1, 2, \dots, n$ einem der K Cluster $\zeta_1, \zeta_2, \dots, \zeta_K$ zuteilen, sodass Cluster homogen bezüglich den Modellkoeffizienten β sind.
- K-Means Algorithmus als Methode zum Bilden von Subgruppen:
  - 1. $\zeta_1 \cup \zeta_2, \dots, \zeta_K = \{1, \dots, n\}$, jeder Proband gehört zu einem Cluster
  - 2. $\zeta_k \cup \zeta_{k'} = \emptyset$ für alle $k \neq k'$, Cluster überlappen nicht

$$\min_{\zeta_1, \dots, \zeta_K} \sum_{k=1}^{K} \frac{1}{|\zeta_k|} \sum_{i,i' \in \zeta_k} \|\beta_i - \beta_{i'}\|^2$$

- Finde $\zeta_1, \dots, \zeta_K$ so mittlere euklidische Distanz zwischen Koeffizienten innerhalb Cluster minimal: ↑
- Finden der optimalen Anzahl Cluster K: Ellbogen-Plot
- Vergleich korrigiertes R^2 von gepoolten Subgruppen und Probanden → wenn Werte fast gleich = gut.

Simulationen für «What-If» Fragen

- Berechnung mit vorgegebenen Profilen anhand Umfragen, wieviele würden welches Produkt kaufen
- Berechne erwartete Ratings mit additiven Modell auf Probandenebene für Proband 1:

respondent	β_{i0}	β_{i11}	β_{i12}	β_{i21}	β_{i22}	β_{i31}	β_{i32}	β_{i41}
1	7.43	-4.18	-3.80	-1.62	-1.82	-0.20	-2.38	1.26
2	2.60	6.09	2.00	-1.09	0.91	1.60	-0.11	-1.36
3	2.59	3.94	0.20	1.86	4.46	-3.20	-5.26	1.55

Produkt	price	variety	kind	aroma
A	high	black	leafy	yes
B	medium	black	granulated	yes
C	low	red	granulated	yes

$$\widehat{rating}_{ij} = 7.4 - 4.2 \cdot pricelow_j - 3.8 \cdot pricemedium_j - 1.6 \cdot varietyblack_j - 1.8 \cdot varietygreen_j - 0.2 \cdot kindbags_j - 2.4 \cdot kindgranulated_j + 1.3 \cdot aromayes_j$$

$$\widehat{rating}_{1A} = 7.4 - 1.6 + 1.3 = 7.1$$

$$\widehat{rating}_{1B} = 7.4 - 3.8 - 1.6 - 2.4 + 1.3 = 0.9$$

$$\widehat{rating}_{1C} = 7.4 - 4.2 - 2.4 + 1.3 = 2.1$$

- Proband 1 wählt Produkt A
- Ausrechnung für alle Probanden, welches dieser am wahrscheinlichsten nehmen würde

Spezialthemen bei Simulationen:

- Was wenn Attributswerte im Design nicht vorhanden?
- Was wenn Probanden nicht repräsentativ für Zielpopulation sind?

Block C: Arbeiten mit Textdaten, Topic Modeling und Recommender Systeme

Arbeiten mit Textdaten

- Viele Kundendaten liegen in unstrukturierter Textform vor (z.B. Mails, Supporttickets, Kommentare auf Sozialen Medien, Chatverläufe, Logs vom Austausch zwischen Kunden und dem Kundenservice)
- Unstrukturierte Daten machen 80-90% aller neuen Daten aus, wachsen 3x schneller als strukturierte Daten
- Unstrukturierte Daten sind (Textdokumente, Videos, Mails, Audiodaten, Sensordaten, Bilder, Logs,…)

Anwendungsbeispiele

- Daten im Customer Relationship Management (CRM) liegen teilweise in Textform vor. Um diese in Modellen (z.B. Churn-Vorhersage) nutzen zu können, müssen diese aufbereiten.
- Analyse von Kundenfeedback (Textform): Kundenbindung oder herausfiltern von relevanten Themen, welche von verschiedenen Kunden angesprochen werden
- Frühzeitiges Erkennen von Kundentrends in den Socialen Medien kann für Kundenakquisition nützlich sein

Allgemein

- Ziel Analyse Textdaten neue, relevante Erkenntnisse zu gewinnen: Text Analysis/Mining/Analytics.
- Entsprechende Methoden werden insbesondere dort gebraucht, wo grosse Mengen textbasierter Daten verarbeitet werden müssen, deren manuelle Analyse zu ressourcen- und zeitaufwändig wäre.

Natural Language Processing (NLP)

- Ist Bereich künstlichen Intelligenz/des Maschine Learnings ML, welcher sich mit «Verstehen» und Nachahmen menschlicher Sprache beschäftigt. Ziel Text semantisch (Inhalt) verstehen. Grundlage NLP ist das ML.
- NLP, um (unstrukturierten) Text in Dokumenten/Datenbanken in normalisierte, strukturierte Daten umzuwandeln, sodass sie für Analysen und/oder Algorithmen des maschinellen Lernens genutzt werden können.

Beispiel Kundenfeedback zu Produkt:

- Schlechter Service. 4.5 Monate Verzögerung bei der Lieferung geht gar nicht!
- Das Produkt ist schlecht und die Verzögerungen sind enorm.
- Vielen Dank für den grossartigen Service.
- Grossartiges Produkt.
- Die Rückmeldungen haben mir sehr geholfen. Danke!
- Schlechte Qualität.
- 50 SFr! Schlechtes Preisleistungsverhältnis.
- Produkt ist wirklich nicht schlecht. Ich kann es nur weiterempfehlen.

Fragestellung: Welche Kunden werden Produkt weiterempfehlen? Computer versteht nur Zahlen
→ Brauchen Ansätze, um Text als Zahlen darzustellen.

Datenaufbereitung

Regelbasiert Ansatz: Spezifische Wörter suchen und manuell Regelwerk erstellen. Geht nur bei wenig Daten Fokussieren auf Ansätze **«komplexere» Fragestellung**, welche grosse Datenmengen skalierbar. Datenaufbereitung, sodass bekannte Modelle aus Data Mining oder statistischen Modellieren anwenden können.

- Text aufteilen (Tokenisierung)
- Uniformative Wörter entfernen (Stoppwörter weg)
- Seltene Wörter entfernen
- Wort auf Stamm reduzieren (Stemming/Lemmatisation)

Tokenisierung – strsplit(data$Feedback[1:2], "[^a-zA-Z0-9äöüÄÖÜ]+")

- Erste Schritt für beliebige Art Textanalyse ist i.d.R. Tokenisierung. Dabei wird Zeichenkette in Texteinheiten (Token-Typ) zerlegt. Wir betrachten hier insbesondere die Zerlegung in Wörter.
- Token kann auch Zeichen, Satz oder N-Gramm (N aufeinanderfolgende Token zusammen) sein.
- Allgemein Aufteilungen nicht immer ganz einfach (z.B. Kommazahlen 3.2 oder 3,4, Trennzeichen, Umlaute in Deutsch oder Apostroph in Englisch (can't), Emojis, usw.).
- Entsprechende Zeichen, z.B. Umlaute/Apostroph vorgängig ersetzen (gsub() mit Regular Expression)
- tokenize_words() Aufteilung gemäss Spezifikation International Components for Unicode durchführt.

Tokenisierung mit N-Gramm

- Um Kontext besser verstehen, manchmal hilfreich, wenn man Text in Fragmente von Wörter zerlegt.
- Beispiel: «sehr zufrieden» vs. «nicht zufrieden»
- N in N-Gramm steht dabei für die Anzahl Wörter
- tokenize_ngrams(x=data$Feedback[8], lowercase = TRUE, n = 2L, n_min = 1L, ngram_delim = " ") #Bigram
- Wichtig, richtigen Wert für n zu wählen. Unigramme (n = 1) sind schneller und effizienter, aber wir erfassen keine Informationen über die Wortfolge. Bei Verwendung höheren Wertes bleiben mehr Informationen erhalten, aber Vektorraum Token (unterschiedliche Elemente) vergrössert sich dramatisch.
- Wir fokussieren hier auf Unigramme, grundsätzlich sind die Ansätze aber übertragbar.

Wortliste: Nach Tokenisierung kann Tabelle mit Häufigkeiten der Wörter erstellen (Document-Term Matrix).

Stoppwörter

- Nicht alle Wörter enthalten gleiche Menge an Informationen für prädiktive Modellierung.
- Stoppwörter: häufige Wörter, die wenig (oder gar keine) sinnvolle Informationen enthalten.
- Ist üblich Stoppwörter für verschiedene NLP-Aufgaben zu entfernen → Reduktion Rechenpower
- Wörter können in unterschiedlichen Formen auftreten (z.B. schlecht, schlechte, schlechter, schlechtes). Allenfalls macht es Sinn, diese zusammenzufassen. → Stemming/Lemmatisation

Es gibt verschiedene Levels von Stoppwörtern

- **Globale Stoppwörter:** Wörter, die in bestimmter Sprache fast immer geringe Bedeutung haben (z.B,: so, da, und, von, ...). Können diese i.d.R. ohne Infoverlust entfernen. Wörter sind in vorgefertigte Stoppwortliste
- **Fachspezifischen Stoppwörter**: Wörter, welche für Fachgebiet uninformativ sind. Immobiliensektor z.B. Bad, Schlafzimmer und Eingang keine Zusatzinformation zur Unterscheidung von Häusern in verschiedenen Regionen. Liste muss manuell erstellt werden. Dazu braucht es entsprechendes Fachwissen.
- **Dokumentspezifische Stoppwörter**: Wörter, welche keine oder nur wenige Information für bestimmtes Dokument liefern. Wörter schwer zu identifizieren und lohnt sich i.d.R. nicht ihnen nachzugehen.

- Für **globale Stoppwörter** stehen Listen zur Verfügung. Listen sind unterschiedlich lang und nicht überlappend. Verwendeten Stoppwörterlisten sollten immer inspiziert werden, um kontrollieren, ob sie für die Anwendung auch sinnvoll sind.

Name	Herkunft	Eigenschaften	Geeignet für
snowball	Snowball-Projekt (Porter)	Kompakt, mehrsprachig	ML/NLP-Basismodelle
stopwords-iso	ISO-konforme Community-Liste	Sehr umfangreich, 50+ Sprachen	Multilinguale Anwendungen
smart	SMART IR System (Cornell)	Englisch; Klassisch aus IR-Forschung	Suchmaschinen, akademisch

- Vieles NLP-Bereich wurde für Englisch entwickelt, gibt aber auch viele Deutsche library
- Stoppwortlisten kann man auch selbst erstellen. Gutes Vorgehen ist bestehen Stoppwortlisten mit spezifischen Wörter zu ergänzen. Startpunkt dafür kann **Häufigkeitstabelle** der Wörter sein.

Stemming

- Gleichen Wörter können in verschiedener Form vorkommen (z.B. Nomen Einzahl oder Mehrzahl, Verb). → häufig sinnvoller diese Wörter zusammenzufassen
- Stemming verschiedene morphologisch Varianten eines Worts auf gemeinsamen Wortstamm zurückführen
- schlechter → schlecht, schlechte → schlecht, Verzögerungen → Verzoger
- Endergebnis muss kein repräsentatives Wort ergeben.
- Ggibt verschiedene Stemming-Algorithmen. Am weitesten verbreitete ist Porter-Algorithmus für Englisch. Snowball-Algorithmus ist Erweiterung Porter-Algorithmus und basiert auf gleichen Prinzipien, lässt sich auch auf andere Sprachen (z.B. Deutsch) übertragen.

Schritte des Snowball-Algorithmus für die deutsche Sprache:

- Umwandeln Wort in Kleinbuchstaben.
- Entfernt Pluralendung, wie -e, -n und -s.
- Entfernt Kasusendung (Akkusativ) -s, -n, -m ,-r.
- Entfernt Adjektivendung, -e, -en, -em und -er.
- Entfernt Verbende, -e, -st, -t, -en, -et, -est, -te, -end
- Entfernt Adverb-Endung, -e, -er, -s.
- Entfernt Derivat-Endung, -keit, -heit, -ung, -isch.
- Reduziert Umlaute, ä zu a, ö zu o, ü zu u.
- Reduziert Doppellaut, aa, ee, oo, auf 1 Buchstabe.
- Reduziert Suffixe, -ion, -ismus und -heit, auf eine Form, -i, -ist, -heit.
- Entfernt bestimmte Endung, -bar, -ig, -isch.
- Algorithmus wendet spezielle Regeln für bestimmte Wörter wie gehen und sehen an.

Ist Stemming immer eine gute Idee?

- Wird häufig angewendet ohne genau über Nutzen nachzudenken. Reduziert Merkmalsraum von Textdaten.
- Gibt wachsende Zahl Untersuchungen, die zeigen, Stemming Textmodellierung kontraproduktiv sein kann.
- Stemming-Algorithmen sind i.d.R. «aggressiv» und wurden so entwickelt, dass sie Sensitivität (wahre Positivrate) auf Kosten der Spezifität (wahre Negativrate) verbessern.
- Stemming-Algorithmen reduzieren Wörter auf Wortstämme. Kann auf Kosten Bedeutung gehen. Wörter werden zu selben Stämmen zusammengefasst, welche unterschiedliche/gegenteilige Bedeutung haben
- In Deutsch werden Konjugationen von unregelmässigen Verben schlecht abgebildet

Lemmatisierung (Alternative zum Stemming)

- Anstelle Regeln, um Wörter auf Stämme zu reduzieren, verwendet Lemmatisierung Wissen über Struktur einer Sprache, um Wörter auf ihre Lemmata (sprachliche Bedeutungseinheit) zu reduzieren. Dabei wird Wörterbuch verwendet, um verschiedene Varianten Wortes auf Stammform zurückzuführen.
- Lemmatisierung ist komplexer. Müssen wissen, zu welcher Wortart Wort gehört, um Lemma identifizieren.

Stemming oder Lemmatisierung: Man kann nur immer eines von beiden machen. (Oder gar keins!)

- Bei guten Wörterbuch kann Lemmatisierung besser als Stemming, da sprachlichen Kontexte berücksichtigt.
- Genauigkeit kommt oft mit höherem Aufwand und Rechenleistung.
- Wahl hängt von spezifischen Anwendung und Anforderungen ab. Wenn es um Suchmaschinen oder schnelle Textanalysen geht, kann Stemming gute Wahl sein, während Lemmatisierung bei komplexeren Textanalysen oder sprachlichen Anwendungen, wie Übersetzungen, bevorzugt wird.

Entfernung weiterer Wörter

- Wörter die in Daten nur selten vorkommen, sind für automatisierte ML-Modelle überflüssig. Entsprechend werden diese Wörter häufig auch entfernt, um Featureraum weiter zu reduzieren.
- Je nach Fragestellung ist sinnvoll Token mit Zahlen zu entfernen (Argument stip_numeric).

Bag-of-Words

- Schlussendlich wollen Wörter in unseren Beobachtungen als Feature für Modelle verwenden. Das heisst, wir verwenden unsere Wortlist als Feature und zählen wie häufig diese in den Beobachtungen vorkommen.
- Anwendung des Bag-of-Words in einem Modell: Naive Bayes
- Bag-of-Words in R mit tidyverse: library(tidytext); unnest_tokens()

TF-IDF (Term Frequency - Inverse Document Frequency): Motivation und Idee

- Beliebte Erweiterung Bag-of-Words ist TF-IDF-Transformation.
- TF-IDF statistisch Mass, das bewertet, wie relevant Wort für Dokument/Beobachtung in Textsammlung ist
- Ziel: Häufige Wörter herunterstufen – seltene, möglicherweise bedeutungsvolle Wörter aufwerten.
- Ergebnis: Vektoren mit gewichtetem Textinhalt – besser für Klassifikation, Clustering etc.
- TF-IDF basiert auf der Multiplikation von zwei Metriken: tf und idf
  - tf: Termfrequenz eines Terms t im Dokument d: $tf(t,d) = \frac{f_{t,d}}{\Sigma_{t' \epsilon d} f_{t',d}}$
  - idf : Inverser Anteil der Dokumente, die Term enthalten: $idf(t) = \log \frac{N}{|d \epsilon D : t \epsilon d|}$ mit N = Anzahl Dokumente
- Je häufiger Wort vorkommt, desto kleiner wird $idf(t)$ → das Wort erhält weniger Gewicht.

Schlussbemerkung zur praktischen Anwendung

- Wortlisten nur anhand Trainingsdaten erstellen. Verwendet Testdaten, erhält zu optimistisches Ergebnis.
- Rechtschreibfehler können ganze Auswertung verfälschen. Daten immer gut anschauen und korrigieren.

Online A/B Testing (oder Split Tests)

- Online A/B Testing umfasst Technik kontrollierten Testen von Kundensoftware, Webseiten, Onlinedienst
- Nutzer wird bei einer Webseite (o.ä.) zufällig auf eine Version weitergeleitet (aktuelle vs. neue Version)
- Werden Daten zum Nutzerverhalten aufgezeichnet (z.B. Verweildauer, Nutzer klickt auf bestimmten Button)
- Auf gesammelten Nutzerdaten wird statistischer Test angewendet (Two-Sample Binomial Proportion Test)
- A/B Test wird seit Internetswachstum 90-er Jahre verwendet, viele Webseiten wie Amazon, Bing, Facebook, Google und LinkedIn führen jedes Jahr Tausende von A/B Tests durch
- Testzwecke: Änderung Nutzeroberfläche oder Änderung von Algorithmen (Suche, Adds, Vorschläge)
- Sehr nützlich in Kombination mit agiler Softwareentwicklung

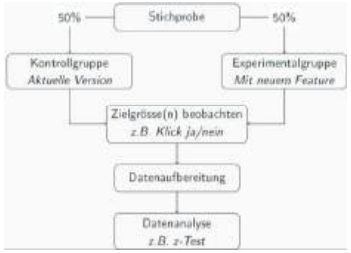
Design

- Nutzer werden persistent einer Gruppe zugewiesen. Bei mehrmaligem Seitenaufruf: Immer gleiche Gruppe
- Es wird eine oder mehrere Zielgrössen bestimmt und erfasst
- Zielgrösse: Werbeeinnahmen per Nutzer, Key User Engagement Metrics (Page Views, Aufenthaltsdauer)
- Resultat: Mehr Einnahmen, tiefer User Engagement (da mehr vertikal Platz verwenden), höhere Ladezeit

Grundsätze für die Anwendung von A/B Testing (siehe Kohavi & Longbotham, 2017)

1. Man ist interessiert an datengetriebenen Entscheidungen und es gibt eine klar definierte Zielgrösse. Achtung: Direktprofit meist keine nachhaltige Zielgrösse, darum oft Sessions per Nutzer, Verweildauer, …
2. Kontrolliertes Experiment ist durchführbar und Resultate vertrauenswürdig. Z.B. kaum möglich Hardware
3. Wir sind nicht gut, eigene Ideen zu beurteilen, darum empirische Untersuchung. Z.B. Manzi (2021): Bei Google haben lediglich 10% der Ideen einen nachweisbaren Einfluss auf Zielgrösse

Struktur von A/B Studien



Spezialthemen

- Laufend kontrollieren, wie Stichprobengrössen/-anteile verlaufen
- «Roboter» können Studien stark verfälschen

Statistische Methoden für A/B Tests

- Berücksichtigung von Unsicherheit bei Versuchen
- Nullhypothese von kein Gruppenunterschied widerlegen

Bsp.: Vergleich Kaufabwicklung aktuelle auf Webseite (A) mit verbesserter Variante (B).

- Beobachten Verbesserung von 10% von B gegenüber A.
- 99%-Vertrauensintervall umfasst die Werte +2% bis $+\infty\%$
- Interpretation: WS, bei Verbesserung von $< 2\%$ das Intervall [+2%,+∞%] zu beobachten ist <1%.

Statistische Methoden für A/B Tests

- Oft A/B Studien einseitige Tests für Vergleich zweier unabhängigen Stichproben verwendet. Als Nullhypothese setzt man Situation, die man verwerfen möchte.
- Bsp. 1: X = Link angeklickt, $X^{(A)} \sim \text{Bernoulli}(\pi_A), X^{(B)} \sim \text{Bernoulli}(\pi_B)$
  - $H_0: \pi_B \leq \pi_A$ (B hat kleinere Wahrscheinlichkeit)
  - $H_A: \pi_B > \pi_A$ (B hat grössere Wahrscheinlichkeit)
- Bsp. 2: X = Verweildauer, $X^{(A)} \sim Exp(\lambda_A), X^{(B)} \sim Exp(\lambda_B)$
  - $H_0: \lambda_B \geq \lambda_A$ (B grössere Rate/kleinere Verweildauer)
  - $H_A: \lambda_B < \lambda_A$ (B grössere Verweildauer)

z-Test für 2 unabhängige Stichproben:

- $H_0: \mathbb{E}(X^{(A)}) \geq \mathbb{E}(X^{(B)})$ (B kleiner Erwartungswert)
- $H_A: \mathbb{E}(X^{(A)}) < \mathbb{E}(X^{(B)})$ (B grösser Erwartungswert)
- Teststatistik: $z = \frac{\hat{\mathbb{E}}(X^{(B)}) - \hat{\mathbb{E}}(X^{(A)})}{\sqrt{\frac{\widehat{\text{Var}}(X^{(A)})}{n_A} + \frac{\widehat{\text{Var}}(X^{(B)})}{n_B}}}$ → Wurzelausdruck ist die Schätzung der gepoolten Standardabweichung
- Bei Grenze $\mathbb{E}(X^{(A)}) = \mathbb{E}(X^{(B)})$ gilt näherungsweise (zentraler Grenzwertsatz) $z \sim \mathcal{N}(0,1)$ und verwerfen H_0 auf Signifikanzniveau α, falls $z \geq \Phi^{-1}(1-\alpha)$

Bsp. Binomialtest für 2 unabhängige Stichproben

- Anwendung z-Test an Binomialdaten: $\mathbb{E}(X) = \pi$. Schätzung Proportionen: $\hat{\pi}_A = \frac{\sum_{i=1}^{n_a} x_i^{(A)}}{n_A}$, $\hat{\pi}_B = \frac{\sum_{i=1}^{n_b} x_i^{(B)}}{n_B}$
- Teststatistik: $z = \frac{\hat{\pi}_B - \hat{\pi}_A}{\sqrt{\frac{\hat{\pi}_A(1-\hat{\pi}_A)}{n_A} + \frac{\hat{\pi}_B(1-\hat{\pi}_B)}{n_B}}}$. Grenze $\pi_A = \pi_B$ verwerfen H_0 auf Sig.niveau α, falls $z \geq \Phi^{-1}(1-\alpha)$
- p-Wert (W'keit, unter H_0 Wert $[z,\infty)$ zu beob.) $= 1 - \Phi(z) < \alpha$, Vertrauensl $= \left[\hat{\pi}_B - \hat{\pi}_A \pm \Phi^{-1}(\alpha) \cdot \sqrt{\widehat{\text{Var}}(\pi)}\right]$
- Vor Durchführung des AB-Tests interessiert oft, wie viele Beobachtungen n_A bzw. n_B erforderlich sind.

- Verwerfen $H_0: \pi_B \leq \pi_A$ falls $z = \frac{\hat{\pi}_B - \hat{\pi}_A}{\sqrt{\frac{\hat{\pi}_A(1-\hat{\pi}_A)}{n_A} + \frac{\hat{\pi}_B(1-\hat{\pi}_B)}{n_B}}} \leq \Phi^{-1}(1-\alpha)$ bzw. $p = 1 - \Phi(z) < \alpha$
- Wollen wissen, wie viele Beobachtungen benötigt werden, um für einen definierten «Mindesteffekt» $\delta = \pi_B - \pi_A > 0$ mit W'keit $P(z > \Phi^{-1}(1-\alpha) | \delta > 0) = 1 - \beta$ die Nullhypothese verwerfen.
- Dies führt zu $1 - \beta = 1 - \Phi(\Phi^{-1}(1-\alpha) - \delta / \sqrt{\frac{\hat{\pi}_A(1-\hat{\pi}_A)}{n_A} + \frac{\hat{\pi}_B(1-\hat{\pi}_B)}{n_B}}$ → pwr.2p.test()
- Für gegebene Werte $\alpha, \beta, \delta, \pi_A, \pi_B$ löst Gleichung nach π_A, π_B auf (oder setzt $\pi_A = \pi_B$, und löst nach $n$auf).

Statistische Methoden für A/B Tests – Bemerkungen:

- Wenn möglich statistische Methode vor Datenerhebung bestimmen
- Stichprobengrössenberechnung je nach Auswertungsmethode schwierig, allenfalls via Simulation
- Statistische Methode / Test ist datenabhängig
- Wegen Grenzwertsatz kann (wie zuvor beim Binomialtest) der z-Test verwendet werden
- Besser: Verteilungsspezifischer Tests (ist effizienter), nichtparametrischer/robuster Test wie Mann-Whitney

A/B/n Tests

- A/B/n Tests = Tests mit mehr als 2 Alternativen, z.B. A/B/C/D
- Bsp.: Farbe Webseite ist weiss. Nun sollen 3 Blautöne als neue Hintergrundfarben verglichen werden.
- Bei A/B/n muss das Problem **multiplen Testens berücksichtigt** werden
- Separaten Tests ergeben folgende p-Werte: p(A vs. B) = 0.1, p(A vs. C) = 0.05, p(A vs. D) = 0.003
- Kann Schluss ziehen, dass auf 5% Niveau C und D Verbesserung ergeben? Wenn man mit signifikanzniveau α insgesamt m Tests durchführt, dann beträgt Fehlerrate $1 - (1-\alpha)^m$ Z.B. $1 - (1-\alpha)^3 = 0.14$
- Korrektur: Signikanzniveau α der einzelnen Tests anpassen, z.B. mit **Bonferroni Korrektur**: $\tilde{\alpha} = \frac{\alpha}{m}$
- Es gibt weniger konservative Korrekturen (jedoch zusätzliche Annahmen), wie z.B.
  - Holm-Bonferroni Korrektur
  - Hochberg's Prozedur
  - Sidak Korrektur
- Alternativ könnte auch der F-Test bzw. Likelihood-Ratio-Tests verwendet werden

Bisherige Ansätze

- Haben uns bei bisherigen Ansätzen auf Zählen der Wörter beschränkt. Haben Worteigenschaft nicht erfasst.
- Je nach Textkorpus (gesamter Text) ist der Featurevektor sehr gross. ⇒ word2vec

word2vec

- Ansatz, welcher auch Zusammenhang zwischen Wörtern und Bedeutung erfasst.
- Algorithmus wurde Google-Team um Mikolov et al. (2013) entwickelt. Damals ziemlich revolutionär.
- Algorithmus verwendet neuronales Netzmodell, um Wortassoziationen aus grossen Textkorpus zu lernen.
- Mit Ansatz lassen sich Token (Wörter/Unterwörter/Multiwort-Ausdrücke) auf Zahlen abbilden → Embedding
- Embedding (Zahlen) für Ähnlichkeit zwischen Tokens finden und Feature prädikative Modelle verwenden.

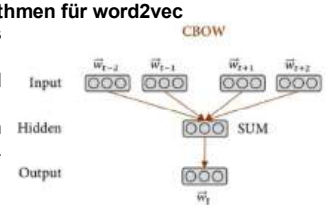
Embedding

- Idee → jedem Wort (Token) ein Vektor zuweisen. Vektoren sollen verschiedene Merkmale der Wörter in Bezug auf gesamten Text erfassen. Z. B. semantische Beziehung des Wortes, Definitionen, Kontext usw.
- Mit diesen numerischen Darstellungen kann man z.B. (Un-)Ähnlichkeit zwischen Wörtern feststellen.
- Auch Bag-of-Word und TF-IDF sind Vektoren, diese sind aber teilweise sehr gross.
- Effektivität von Word2Vec beruht auf der Fähigkeit, Wörter in «einfachen» Vektoren zusammenzufassen. Bei ausreichend grossen Datensatz können damit, basierend auf Vorkommen Wortes im Text, Aussagen über dessen Bedeutung machen. Schätzungen ergeben Wortassoziationen mit anderen Wörtern im Korpus.

2 verschiedene Trainingsalgorithmen für word2vec

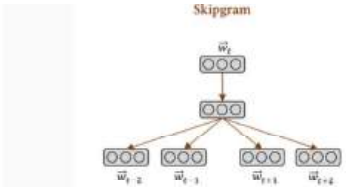
Continuous Bag of Words Model (CBOW)

- Architektur ähnlich wie feed forward neuronales Netz.
- Zielwort wird aus Liste von Kontextwörtern vorhergesagt.



Skip-gram

- Intuitiv: Gegenteil CBOW
- Einfaches neuronales Netz mit einem versteckten Layer, das trainiert wird, um für gegebenes Inputwort, die Wahrscheinlichkeit, dass ein bestimmtes verknüpftes Wort vorhanden ist, vorherzusagen.



Umsetzung

- Korpus wird als Vektor Grösse V dargestellt, wobei jedes Element in V einem Wort im Korpus entspricht.
- Haben Trainingsprozess Paare Ziel- und Kontextwörter. Vektoren haben beim entsprechenden Wort Wert 1 und sonst überall Wert 0. Versteckte Schicht lernt Repräsentation jedes Wortes.
- Dieser N-dimensionale Vektor ist unser Embedding (N wählbar).

CBOW vs. Skip-gram

- CBOW lässt sich schneller trainieren und kann häufiger vorkommende Wörter besser darstellen.
- Skip-Gram funktioniert besser bei kleinen Datensätzen und kann weniger häufige Wörter besser darstellen.
- Welches Modell wir wählen, hängt vom Problem ab, das gestellt ist. Ist es für unser Modell wichtig, seltene Wörter darzustellen? Wenn ja, sollten wir Skip-Gram wählen. Haben wir nicht viel Zeit zum Trainieren und seltene Wörter sind nicht so wichtig für unsere Lösung? Dann sollten wir CBOW wählen.

Preprocessing

- Word2vec-Modell und Preprocessing? Gute Frage! • In Literatur keine klare Empfehlung. → Testen
- Bei kleinem Textkorpus kann Fokussierung auf die wichtigen, seltenen Wörter aber schon etwas bringen.

Mögliche Faustregeln

- Tokenisierung: Unverzichtbar – die Basis für die Wortlisten
- Kleinbuchstaben: Vereinheitlichung, reduziert unnötige Vektoren
- Satzzeichen/Sonderzeichen entfernen: Entfernt irrelevante Tokens (z.B. "#", "!", usw.)
- Lemmatization: Sinnvoll, da semantisch verwandte Wörter zusammengeführt werden (go, goes, went ⇒ go)
- Stopwörter entfernen? Optional:
  - Entfernen = Fokus bedeutungstragende Wörter
  - Behalten = besser Modell für Syntax/Satzstruktur
- Stemming? Eher nicht
  - Zu grob, kann Wortbedeutung zerstören
  - Verringert Qualität der Wortvektoren

Word2vec für Prediction-Modelle

- Frage ist, wie können wir unser word2vec-Embedding als Feature für ein Prediction-Modell verwenden?
- In einem Text haben wir das Embedding für verschiedene Wörter.
- Als Feature wird z. B. Mittelwert des Embedding verwendet. Das funktioniert, für mich, überraschend gut.

Vortrainierte Modelle

- Anstelle eigenen Modelles kann man auch ein vortrainiertes Modell verwenden.
- Diese wurden in der Regel mit grossen Textkorpussen (z.B. Wikipedia, Google-News) trainiert.
- Vortrainiert Modell muss nicht immer besser sein. Kann Bias enthalten. Kommt an, wofür Modell entwickelt.

Word2Vec vs. moderne Embeddings

- Word2Vec war Meilenstein im NLP – ideal für Analogie-Rechnungen und explorative Semantik.
- Moderne Modelle (BERT, GPT) gehen weiter: verstehen Wörter im Kontext und satzweise Repräsentation.
- Klassische Embedding heute nur Lehrzwecken oder einfache Anwendung. Modelle kaum weiterentwickeln

Kriterium	Word2vec	BERT / GPT (moderne Modelle)
Repräsentation	Wort = fixer Vektor	Kontextabhängige Satz-/Token-Vektoren
Umgang neue Wörter	Scheitern bei unbekannten Wörtern	Robuster durch Subwords/Tokenisierung
Kontextsensitivität	Kein Satzkontext	Kontext wird vollständig berücksichtigt
Trainierbarkeit	Einfach lokal trainierbar	Nur mit GPU/Cloud oder vortrainiert
Anwendung	Analogie & einfache Semantik	Suche, Klassifikation, QA, Textverständnis
Nutzung in R	read.word2vec()	über reticulate() / Python / API

Transformer Modelle

- haben klassische Embeddings wie Word2Vec abgelöst und bilden heute Standard im modernen NLP.
- Zugrunde liegenden neuronalen Netze sind deutlich komplexer als klassischen Verfahren wie word2vec.
- Zentrum steht Self-Attention-Mechanismus. Erlaubt dem Modell, Wortbedeutung im Kontext gesamten Satzes oder Dokuments zu gewichten. So kann Modell sich auf relevante Textteile konzentrieren.
- Beispiel: In Frage „Wer schrieb Faust?" erkennt Modell, dass «wer» Person meint und «Faust» literarisches Werk ist – obwohl beide Wörter im Satz weit voneinander entfernt stehen. Bekanntesten Vertretern:
  - BERT: Bidirectional Encoder Representations from Transformers
  - GPT: Generative Pre-trained Transformer
- Vortrainierte Modelle sind z.B. Plattform Hugging Face frei verfügbar und lassen mit Python leicht nutzen.

BERT – Entwickelt von Google AI in 2018

- tiefes Lernmodell, ausgelegt Kontext verstehen, indem es Wörterbedeutung im Kontext Gebrauchs erfasst.
- SBERT (Sentence-BERT) Variante von BERT, die speziell für Satzähnlichkeit und Klassifikation entwickelt wurde. Gegensatz BERT liefert es direkt Vektor für ganze Sätze– ideal für Textvergleich oder Clustering.
- **Kernfeatures**
  - Kontextverständnis beide Richtungen: berücksichtigt gesamten Satzkontext – vor und nach Wort.
  - Pre-training und Fine-tuning: wird zuerst auf grossen Textmengen (z.B. Wikipedia) trainiert, um ein allgemeines Sprachverständnis zu entwickeln, und kann dann für spezifische Aufgaben feingetunt werden.
- **Anwendung**: Textklassifikation, Sentiment-Analyse, Frage-Antwort-Systeme und mehr.

GPT – Entwickelt von OpenAI, Erste Version wurde 2018 veröffentlicht

- Modell, das darauf spezialisiert ist, Text zu generieren, der wie menschliche Sprache klingt.
- Wie BERT basiert Transformer-Modell, verwendet andere Technik: autoregressive Sprachmodellierung.
- **Kernfeatures**
  - Autoregressives Modell: Jedes Token wird auf Grundlage der zuvor generierten Tokens vorhergesagt, was eine kohärente Texterzeugung ermöglicht.
  - Skalierbarkeit: sind in verschiedenen Grössen verfügbar, Effektivität steigt mit Anzahl Parameter.
- **Anwendung**: Automatisches Schreiben von Texten, Übersetzung, Zusammenfassung und mehr.

Vergleich von BERT und GPT

- Obwohl BERT und GPT beide auf Transformer-Modell basieren, haben sie unterschiedliche Stärken:
- BERT: Besser für Aufgaben geeignet, bei denen das Verständnis des Kontexts entscheidend ist. Ideal für Klassifikationsaufgaben wie Sentiment-Analyse oder Informationsentnahme.
- GPT: Besser für Aufgaben, bei denen kreative Textgenerierung gefordert ist. Kann als Grundlage für Chatbots oder kreative Schreibwerkzeuge dienen.

Vortrainierte Sprachmodelle

- Statt eigene Modell trainieren, nehmen leistungsstark/vortrainiert Modell. Basieren Transformer-Architektur
- Vorteil: Kein eigenes Training – direkt einsetzbar für Klassifikation, Sentiment, Ähnlichkeit, Textgenerierung.

Modell	Fokus	Eigenschaften
SBERT	Satzähnlichkeit & Klassifikation	Liefert kompakte Satz-Embeddings
GPT	Textgenerierung & Prompt-KI	Erzeugt flüssige Texte, ideal für Chat & Co.
TF-IDF/word2vec	Einfache Baslines	Gut für erste Experimente oder kleine Projekte

SBERT (Sentence-BERT)

- Satz-Embedding mit Transformer-Technologie
- Ideal für:
  - Klassifikation mit dichten Vektoren
  - Semantische Suche
  - Clustering / Ähnlichkeitsanalysen
- Vorteile:
  - Effizient, offline nutzbar
  - Gut geeignet für Random Forest / SVM
  - Multilingual verfügbar

ChatGPT als Klassifikator

Funktioniert über Prompt + Review-Text

- Vorteile:
  - Kein Modelltraining nötig
  - Flexibel & anpassbar
  - Gut bei kleinen Datenmengen
- Einschränkungen:
  - Keine Scores/Schwellenwerte
  - API-gebunden & kostenpflichtig
  - Nicht für maschinelles Modelltraining geeignet
- Typisch Sanity Check oder qualitative Ergänzung

Sentimentanalyse

- geht darum, automatisch zu ermitteln, ob Text positive, negative oder neutrale Meinung ausdrückt.
- Wichtig für Unternehmen um Kundendaten wie Umfrageantworten, Bewertungen, Support-Tickets und Kommentare in sozialen Medien auszuwerten, um z. B. Markenruf in sozialen Medien zu überwachen.
- Insbesondere grossen Datenmengen hat keine Chance, dies manuell auszuwerten. Zudem manuelle Auswertungen häufig nicht konsistent. Vorteile: Skalierbarkeit, Analyse in Echtzeit und konsistente Kriterien

2 verschiedene Ansätze

- Regelbasierte Systeme basierend auf vordefinierten Wortlisten (welche Wörter sind positiv, welche negativ).
- Erstelle Modell für Sentimentklassen anhand Trainingsdate (Bag-of-Word, TF-IDF, word2vec, SBERT/GPT)
- Daneben gibt es auch noch Mischformen.

Sentimentanalyse mit Wortlisten

- Zählt Anzahl positiven und negativen Wörter und schaut, welche Schwergewicht haben.
- Annahme Wörter unabhängig voneinander sind. z. B. "die Benutzeroberfläche ist nicht so toll". Während "nicht so toll" Negativität ausdrückt, erkennen regelbasierte Systeme das Wort "toll" und fügen es der Liste der Positiven hinzu. Auch Ironie wird nicht erkannt.
- Für Beurteilung gibt vordefinierte Wortlisten, je nach Fragestellung sollten Listen angepasst werden.

Topic Modeling

- Topic Modeling: relevante Themen (Topics) automatisiert aus Texten extrahieren. Wie? Ideen?
  - Regelbasierte Ansätze mit Suche nach bestimmten Worten in Texten.
  - Themen-Klassifikation anhand Trainingsdaten.
  - Clustern von Texten und dann Gemeinsamkeiten in Gruppen suchen. Z.B. mittels einer Wortwolke.
- Topic Modeling: stat. Modellierung zur Ermittlung abstrakten Themen, die in Dokumentensammlung sind
- Unsupervised ML Ansatz, ist in Lage Dokumentenreihe scannen, Wort-/Satzmuster erkennen und automatisch Wortgruppen und ähnliche Ausdrücke gruppieren, die Dokumentenreihe besten charakterisieren
- Hofft, dass diese Muster intuitiven Themen entsprechen, welche sich interpretieren lassen.
- Cluster: jedes Dokument ist in einem Cluster vs. Topic Modeling Mischmodell Dokument in mehreren Topics

Annahmen

- Dokumentensammlung haz verschiedene Topics
- Jedes Dokument hat verschiedene Topics
- Topics haben spezifischen Wörtern
- Ein Wort kann in mehreren Topics vorkommen

Ergebnis eines Topic Modeling

- Topics, d.h. wichtigsten Wörter in diesen Topics, welche in diesen Dokumenten auftreten.
- Wahrscheinlichkeiten für verschiedenen Topics in allen Dokumenten.

Achtung:

- Topics bzw. die dazugehörigen Wörter müssen interpretiert werden. Das Topic Modeling alleine ist kein Ersatz für die menschliche Interpretation, sondern gibt nur Hinweise wie verschiedene "latente" Themen zusammenhängen können, indem ihr gemeinsames Auftreten in Dokumenten analysiert wird.
- Blind angewendet, werden nur in den wenigsten Fällen brauchbare Ergebnisse erhalten.
- Enttäuschung kann gross sein, wenn die Ergebnisse uninformative oder sogar missverständlich sind.

Latent Dirichlet Allocation (LDA) – klassische Methode für Topic Modeling. Prinzip:

- Jedes Dokument ist Mischung aus Topics. D.h. jedes Dokument enthält Wörter aus mehreren Topic. Z.B. Dokument besteht 80 % aus Wörter Topic Service und 20 % Topic Ausblick (Beispiel Restaurantreviews).
- Jedes Topic besteht aus Mischung von Wörtern. Z.B. häufigsten Wörter im Topic Service sind "Kellner", "Freundlichkeit", "Wartezeit" und wichtigen Wörter zum Topic Ausblick sind "Aussicht", "Zürich", "See", usw. Wörter können von mehreren Topics genutzt werden, z.B. "fantastisch" kann in beiden Topics vorkommen.
- LDA ist eine mathematische Methode, um beides, die Topicmischung und die Wortmischung, die mit jedem Topic verbunden ist, gleichzeitig modellieren. → Dirichlet Verteilung

Ziele/Bedingung:

- Dokument so monochromatisch (eindeutig einem Topic zugeordnet) wie möglich.
- Auch Wörter sollen so monochromatisch (eindeutig einem Topic zugewiesen) wie möglich sein.
- Erst durch diese Bedingungen kann Problem lösen, da kein Verständnis zur Bedeutung der Wörter benötigt

Umsetzen mit Gibbs Sampling

- Zufällige Zuteilung der Wörter in den Dokumenten zu einem Topic.
- Anschliessend werden Wörter neu klassifiziert. Annahme: alle anderen Wörter sind richtig zugeteilt.
- Zuteilung Wort w im Dokument d gem. Wahr' für verschiedenen Topics t, welche proportional zur Multiplikation ist: $(p(Topic\ t|Dokument\ d) + \alpha) \cdot (p(Wort\ w|Topic\ t) + \beta)$, Parameter α, β aus Dirichlet-Verteilung
- Zu Beginn ist Zuteilung sehr schlecht. Mehrmaliges iterieren wird Zuteilung besser bis stabiler Zustand

Anzahl Topics

Anzahl Topics ist wichtiger Parameter bei Topic Modeling → muss vorgegeben werden. Wie bestimmen?

- Gute Anzahl Topics hat man gefunden, wenn diese interpretierbar sind und Sinn ergeben.
- Gibt Metriken, welche Hinweis für sinnvolle Anzahl Topics geben. Populär ist **Coherence Mass Cv**.

Coherence Mass Cv

- Metrik misst wie Wörter in Topic miteinander verbunden sind, wobei stat. Unabhängigkeit kontrolliert wird.
- Probabilistische Coherence für jedes Wortpaar $\{a, b\}$ wird berechnet: $P(b|a) - P(b)$, wobei in diesem Topic $\{a\}$ wahrscheinlicher ist als $\{b\}$. Interpretation: Wenn uns auf Dokumente beschränken, die Wort $\{a\}$ enthalten, dann sollte Wort $\{b\}$ in diesen Dokumenten wahrscheinlicher sein als zufälligen Auswahl aus Korpus.
- Coherence Mass für ein Topic, wird Coherence zwischen top M Wörtern des Topics gerechnet & gemittelt.
- Grösserer Wert ist grundsätzlich besser. Dieses Mass ist aber auch nicht unumstritten, der grösste Wert muss nicht zwingend zu den am besten interpretierbaren Topics führen.

Hyperparameter

Neben Anzahl Topic gibt es noch 2 weitere Hyperparameter beim Topic Modeling:

- α: Niedriger Wert ordnet jedem Dokument weniger Topics zu. Bei kurzen Texten sinnvoll. Default 0.1.
- β: Bei niedrigen Wert weniger Wörter einem Topic zugewiesen, während bei hohen Wert mehr Wörter verwendet werden, wodurch die Topics zueinander ähnlicher werden. Default 0.05.

Darüber hinaus sollte auch **Anzahl Iteration** nicht zu klein gewählt werden. Empfehlung **mindestens 500**.

Iterativer Prozess

Topic Modeling ist iterativ. Resultat immer analysiert, Preprocessing angepasst und Parameter korrigiert müsse

- Welches Preprocessing sinnvoll? Welche Stoppwörter entfernen? Häufig inspiziert man Resultate und erweitert Stoppwortliste laufend mit Wörtern, welche man nicht in Topics haben möchte.
- Anstelle von Wörter können auch **n-gramms** analysiert werden.
- Teilweise sinnvoll nur **Nomen** verwenden. Wie diese automatisiert detektieren? **Part Of Speech Tagging**

Allgemein: Begriff Topic Modeling kann irreführen. Grundsätzlich ist Ansatz Werkzeuge zum Lesen und Ergebnisse sollten nicht überinterpretieren. Ausser hat starke theoretische Vorstellung über Anzahl Topics in einem bestimmten Korpus, oder man hat die Ergebnisse sorgfältig validiert

BERTopic

Es gibt auch noch andere Ansätze, z.B. Latent Semantic Analysis (LSA) oder BERTopic.

- BERT-based Topic Modeling kombiniert klassische Methoden (z.B. Clustering, CountVectorizer) mit modernen NLP-Techniken (z.B. Textembedding).
- Eignet sich besonders gut für kurze Texte, wie z.B. Bewertungen, Abstracts, Tweets.
- Erfordert (anders als LDA) keine fixe Anzahl Topics – diese wird durch Clustering automatisch bestimmt.

1. **Preprocessing** (optional, aber empfohlen): Obwohl BERT mit rohem Text umgehen kann, verbessert gezieltes Preprocessing die Qualität der extrahierten Top-Wörter: Entfernen von Stoppwörtern und Filterung seltener Wörter. Wichtig, da Topic-Wörter nicht durch BERT, sondern durch einen CountVectorizer (Zählen der Wörter, welche in den Dokumenten eines Topics besonders häufig vorkommen) bestimmt werden.
2. **Textembedding**: Texte werden mit vortrainierten Transformer-Modell (z.B. dbmdz/bert-base-german-cased) in hochdimensionale Vektoren umgewandelt. → Semantisch ähnliche Texte liegen näher beieinander.
3. **Dimensionsreduktion**: UMAP hochdim. Embedding auf 2–5 Dimensionen projiziert: Clustering ermögliche
4. **Clustering:** Anwendung von HDBSCAN (dichtebasiertes Clustering, Erweiterung DBSCAN). → Dokumente, die keiner dichten Region angehören, werden als Ausreisser (Topic = -1) markiert.
5. **Topic-Identifikation:** Jedes Cluster ein Bag-of-Words (CountVectorizer) für wichtigsten Wörter pro Cluster
6. **Tuning/Visualisierung:** Anzahl/Feinheit Topics kann indirekt mit Para min_cluster_size HBSCAN steuern

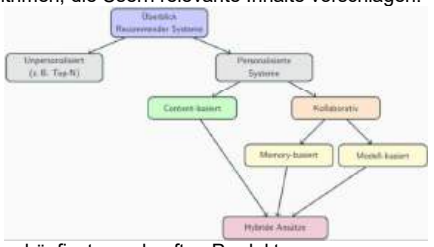
Erweiterungsmöglichkeiten: Labels können durch LLMs wie ChatGPT automatisch generiert werden.

Recommender Systeme (Empfehlungssysteme)

- Sie haben grossen Einfluss auf unser Leben. Empfehlen uns nächste Video oder Netflix-Serien, Songs.
- Personenbezogene Werbung basieren auf ihnen. Algorithmen, die Usern relevante Inhalte vorschlagen.

Recommender Systeme im Marketing

- Im Marketing haben Recommender System insbesondere traditionellen Cross-Selling-Strategien verändert.
- Steigerung Umsatzraten durch automatisierte Identifizierung und Empfehlung interessanter Produkte.
- Auch Kunde profitiert davon, weil er ideale Kombinationen an Produktempfehlungen erhält.



Verschiedenste Arten von Recommender Systemen → →

Unpersonalisierte Recommender Systeme

- Empfehlen Usern z.B. einfach allgemein beliebtesten Artikel, z.B. Top-10-Filme, meistverkaufte Bücher, die am häufigsten gekauften Produkte.
- Sind guter erster Schritt, stossen schnell an Grenzen: keine personalisierte Empfehlung, mögliche Wiederholungen (z.B. Empfehlung bereits gekaufter Artikel) und fehlende Relevanz individuelle Nutzerbedürfnisse.
- Sobald genügend Nutzerdaten vorliegen, sollte personalisierte Empfehlung → Nutzer gezielt ansprechen

Personalisierte Recommender Systeme

- Ziel: Nutzerindividuell Empfehlung basierend Verhalten/Vorliebe/ggf. Profildaten. Merkmal gut Empfehlung:
  - Relevant für den jeweiligen Nutzer
  - Neu – keine Wiederholung bereits konsumierter Artikel
  - Divers – decken diverse Interessen ab
  - Zeitlich passend – verfügbar & kontextsensitiv
- Typen personalisierter Systeme:
  - Content-based Filtering
  - Collaborative Filtering
  - Hybride Ansätze

  → Systeme analysieren frühere Käufe/Bewertung/Nutzungsdaten → liefern massgeschneiderte Vorschläge

Inhaltsbasierte Methoden (Content-based)

- Content-basierte Systeme nutzen Merkmale von Artikeln und/oder Nutzern, um Empfehlungen abzuleiten. Ziel ist es, ein Modell zu erstellen, das beobachtetes Verhalten (z.B. Käufe, Klicks, Bewertungen) erklärt.
- Beispiel: Empfehlung basiert auf inhaltlicher Ähnlichkeit der Artikel – z.B. gleicher Autor, Genre oder Thema.
- Nutzerin liest Buch A → System findet ähnliche Inhalte → Buch B wird empfohlen

Vorteile

- Personalisierung: Hohe Relevanz, da Präferenzen einzelner Nutzer direkt gelernt werden.
- Unabhängigkeit: Keine Interaktionsdaten anderer Nutzer nötig (≠ kollaborativ).
- Transparenz: Empfehlungen lassen sich gut begründen (z.B. weil du A magst, empfehlen wir B).
- Weniger Cold-Start-Probleme: Funktioniert auch neue Nutzer oder neue Artikel, solange Merkmale bekannt.
- Datenschutzfreundlicher: Oft weniger personenbezogene Daten nötig, da Fokus auf Artikelmerkmalen liegt.
- Diversität: Bei vielfältigen Attributen (Genre, Thema) können abwechslungsreiche Empfehlung entstehen.

Herausforderungen

- Merkmalsextraktion: Gute Artikelmerkmale müssen manuell oder durch ML gewonnen werden.
- Über-Spezialisierung: Empfehlungen sind oft sehr ähnlich zu bisherigen Inhalten.
- Filterblase: Nutzer bleiben im «Interessen-Bubble», neue oder kontroverse Inhalte werden kaum angezeigt.

- Begrenzte Serendipität: Überraschende oder unerwartete Empfehlungen sind selten.
- Skalierbarkeit: Verarbeitung grosser Feature-Mengen ist rechenintensiv.
- Beliebtheitsproblem: Artikel können thematisch passen, aber trotzdem unattraktiv / qualitativ schlecht sein.

Kollaborative Filtermethoden

- Kollaborative Methoden stützen sich **nur** auf aufgezeichnete Interaktion zwischen User & Artikeln, um neue Empfehlungen zu erstellen. Interaktionen werden in user-item interaction matrix (User-Artikel) gespeichert.
- Idee: Interaktionen zwischen Usern und Artikeln ist ausreichen, um ähnliche User und/oder ähnliche Produkte zu finden, um darauf aufbauend Empfehlungen zu geben. 2 verschiedene Ansätze:
  - **Memory based:** arbeiten direkt mit Werten aufgezeichneten Interaktionen und gehen von keinem Modell aus. → Suche nach nächsten Nachbarn → Vorschlag der beliebtesten Artikel unter diesen Nachbarn.
    - **User based** (userzentriert)
    - **Item based** (artikelzentriert)
  - **Model based:** geht von zugrundeliegenden generativen Modell aus, das User-Item-Interaktionen erklärt. Man versucht, dieses Modell zu entdecken, um neue Empfehlungen zu machen.

Memory and User based (userzentriert)

- Um User Empfehlungen zu geben, werden User mit ähnlichsten Interaktionsprofil (nächste Nachbarn) identifizieren, um Artikel vorzuschlagen, die unter diesen Nachbarn am beliebtesten sind (und für Zieluser neu).
- User wird durch Vektor der Interaktionen mit Produkten (Zeile in Interaktionsmatrix) dargestellt.
- Zwei User mit ähnlichen Interaktionen bei denselben Elementen werden als ähnlich betrachtet.
- Auswahl k-nächsten Nachbarn zum Zieluser.
- Empfehlung Artikel, welche für Referenzuser neu
- Bestimmung beliebtesten Artikel unter Nachbaren.
- Reihenfolge? → Learning to rank

Ähnlichkeitsmasse

- Cosine similarity (Kosinusähnlichkeit): Kosinus eingeschloss Winkels zweier Vektoren a, b: $\frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}}$
- Alternativen: Korrelation, Jaccard Koeffizient (asymmetrische Ähnlichkeit) bzw. euklidische oder Manhattan Distanz (kleinste Distanzen am ähnlichsten)
- Achtung: Anzahl gemeinsame Interaktionen (wie viele Elemente wurden von beiden User bewertet) sollte mitberücksichtigt werden! Vermeiden, jemand nur eine Interaktion mit Zieluser gemeinsam hat, 100%ige Übereinstimmung hat und näher angesehen wird als jemand, 100 gemeinsame Interaktionen mit nur 98%.

Memory and Item based (artikelzentriert)

- User Empfehlung werden Artikel gesucht, die ähnlich sind, mit denen User bereits "positiv" interagiert hat.
- Auswahl Artikel, welche Zieluser besten gefallen haben. Vektor dieser Artikel → Spalte in Interaktionsmatrix.
- Zwei Artikel sind ähnlich, wenn User, die mit beiden interagiert haben, dies auf ähnliche Weise getan haben.
- Ähnlichkeit des Artikels mit allen andern Artikel bestimmen. k-nächste Artikel bestimmen.
- Empfehlung diejenigen Artikel, welche für den Zieluser neu sind.
- Um relevantere Empfehlungen zu erhalten, können wir diese Aufgabe für mehr als nur den «besten» Artikel des Zielusers durchführen und die n bevorzugten Artikel berücksichtigen. In diesem Fall können wir Artikel empfehlen, die in der Nähe mehrerer dieser bevorzugten Artikel liegen.

Artikelzentriert

+ **Stabilität**: Items ändern sich weniger häufig als Nutzerpräferenzen, was zu stabileren Empfehlungen führt.
+ **Robust**: Meist viel User mit Artikel interagier, Nachbarschaftsuche weniger empfindlich einzelne Interaktion
– **Beschränkte Diversität**: Neigt sehr ähnliche Item empfehlen, was Entdeckung neue Kategorie einschränkt.
– **Cold Start neue Items**: Neue Items ohne Bewertungen sind schwer zu empfehlen.

Userentriert

+ **Personalisierung**: Empfehlung nur Interaktion ähnlich Nutzern stammen, erhalt personalisierterte Ergebnis
+ **Dynamik**: Gut geeignet für Systeme, in denen die Nutzerpräferenzen sich schnell ändern.
– **Empfindlich:** Da User meist wenig Artikel interagiert, Methode empfindlich auf aufgezeichnete Interaktion
– **Cold Start neue Nutzer**: Empfehlungen für neue Nutzer ohne vorherige Interaktionsdaten nicht möglich.

Allgemeine Limitierung Memory basierter Methoden

- Gefahr «rich-get-richer» Effekts (System neigt dazu nur noch beliebte Artikel zu empfehlen). Wollen aber den Usern auch Chance geben etwas Neues zu entdecken.
- Mangelnde Skalierbarkeit (neuen Empfehlung für grosse Systeme zeitaufwendig) → Modellbasiert Methode

Modellbasierte kollaborative Filtermethoden

- Geht von latenten Modell aus, das Interaktionen erklärt.
- Hauptannahme: Im niedrigdimensionalen Raum gibt es latente Merkmale, die sowohl User als Artikel so darstellen können, dass wir hochdimensionale Interaktionsmatrix als Skalarprodukt (Dot product) erhalten.
  ⇒ Matrix Faktorisierung (matrix factorisation), welche User-Item Interaktionsmatrix in Produkt zweier rechteckiger Matrizen geringerer Dimensionalität zerlegt.

Matrix Faktorisierung

- Beispiel Modellierung der Interaktionen bei der Filmbewertung:
  - Gibt einige Merkmale, welche Filme beschreiben (und voneinander unterscheiden).
  - Diese Merkmale können auch dazu verwendet werden, die Präferenzen der User zu beschreiben (hohe Werte für Merkmale, die der User mag, ansonsten niedrige Werte).
- Modell nicht explizit vorgeben, stattdessen soll System Merkmale selbst entdecken (Repräsentation finden).
- Da extrahierten Merkmale erlernt und nicht vorgegeben sind, haben sie nicht immer intuitive Interpretation.
- User, welche sich im Bezug der Präferenzen nahe und Artikel, welche sich in Bezug auf die Merkmale ähnlich sind, liegen im latenten Raum nahe beieinander.
- Anwendung Memory basierte Ansätze auch auf U und V (kleiner Dim. → schneller).
- Neue Nutzer/Items, selbst mit minimalen initialen Interaktionen, können in bestehende Matrixfaktorisierungsmodell integrieren, indem ihre latenten Merkmale durch umgekehrte Matrixmultiplikation geschätzt werden. Werden bekannten Faktoren Matrix genutzt, um fehlenden Faktoren berechnen und vorhersagen.

Mathematik hinter der Matrix Faktorisierung

- Wie $U(n \times k)$ und $V(k \times m)$ bestimmen, sodass $X \approx U \times V = \hat{X}$?
- k ist Dimension im latenten Raum. Kann festgelegt werden.
- Bekannter Optimierungsalgorithmen basiert auf Gradientenverfahren (Gradient descent).
- Als Startwerte werden U und V mit Zufallswerten initialisiert.
- Danach Matrixmultiplikation $U \times V$ und Unterschied zur beobachteten Interaktion-Matrix X bestimmt.
- Minimierung Differenzen (Anpassung der Werte von U und V wenn diese gross sind).
  - Quad. Fehler e für jedes Element User-Artikel Interaktion: $e_{ij}^2 = \left(x_{ij} - \hat{x}_{ij}\right)^2 = \left(x_{ij} - \sum_{k=1}^{K} u_{ik} v_{kj}\right)^2$
  - u_{ik}, v_{kj}sollen in Richtung geändert werden, die Fehler minimiert. Fehlergleichung partiell ableiten: $\frac{\delta e_{ij}^2}{\delta u_{ik}} = -2\left(x_{ij} - \hat{x}_{ij}\right)(v_{ki}) = -2e_{ij}v_{ki}$, $\frac{\delta e_{ij}^2}{\delta v_{ki}} = -2\left(x_{ij} - \hat{x}_{ij}\right)(u_{ki}) = -2e_{ij}u_{ki}$
  - Aktualisierung: $u'_{ik} = u_{ik} + \alpha \frac{\delta e_{ij}^2}{\delta u_{ik}} = u_{ik} + 2\alpha e_{ij} v_{ki}$, $v'_{ik} = v_{ik} + \alpha \frac{\delta e_{ij}^2}{\delta v_{ik}} = v_{ik} + 2\alpha e_{ij} u_{ki}$. α ist Lernrate
- Vereinfachung: U und V können auch alternierend aktualisiert werden.

Erweiterungen der Matrix Faktorisierung

- Skalarprodukt ist nicht immer geeignet (z.B. bei booleschen Interaktionen). In diesem Fall kann z.B. logistische Funktion auf Skalarprodukt aufgesetzt werden. D.h. Ergebnis klassischen Matrixfaktorisierung wird durch Sigmoid-Funktion transformiert, um es als Wahrscheinlichkeit interpretieren zu können.
- Moderne Systeme setzen zunehmend auf Deep Learning. Neben erweiterten Matrix-Faktorisierungsverfahren gibt es auch sequenzielle Modelle (z.B. RNNs, Transformers) für Session-based Recommender.

Deep Learning Recommender – Erweiterte Matrixfaktorisierung

- Idee: Statt nur Skalarprodukt (wie bei klassischer Matrixfaktorisierung), werden User- und Item-IDs durch Embedding-Layer in dichte Vektoren überführt. Vektoren kombinieren und durch neuronale Netz verarbeitet

Architektur im Überblick mit Schritt und Beschreibung

- Input — Ein einzelnes User-Item-Paar (z.B. User 17 & Film 204) als One-Hot-Vektoren
- Embedding Layer — Wandelt IDs in kompakte Vektoren um (z.B. Dimension 32)
- Elementwise Multiplikation — Berechnet Interaktion auf Feature-Ebene: Elementweise $u \odot v$
- Dense Layers — Mehrere Schichten mit Aktivierungsfunktionen (z.B. ReLU)
- Output — Vorhersage z.B. einer Interaktion (0/1) oder eines Ratings (1–5)
- Trainingsdaten: User-Item-Paare mit bekannter Interaktion (z.B. Klick = 1, kein Klick = 0)
- Loss: Binär (z.B. Klick / kein Klick): Binary Cross-Entropy, Rating-Vorhersage: Mean Squared Error (MSE)

Vorteile

- Flexibler als klassische Matrixfaktorisierung
- Kann komplexe, nichtlineare Beziehungen modellieren
- Erweiterbar (z.B. mit Text/Bildinformationen oder Kontextdaten)

Daten für User-Item Interaktionsmatrix

Nicht immer ist Erstellen User-Item Interaktionsmatrix einfach, wie in Bsp. Film- oder Artikelbewertungen. Dennoch gibt bei Webinteraktionen verschiedenste Möglichkeiten entsprechende Matrix zu erstellen. Beispiele:

- Besuchte Seiten
- angeklickte Artikel
- Mausbewegungen
- Artikel Warenkorb u/o Bestellvorgang
- On-Page-Indikatoren wie Verweildauer auf Seite
- zeitliche Aspekte (z.B. wie lange liegt Interaktion zurück)

Daten Preprocessing → Normalisierung:

- Preprocessing kann wichtig sein
- Mittelwert-Zentrierung (Mittelwert abziehen)
- Bei Ratings von User müssen wir z.B. mit Verzerrungen rechnen, da nicht jeder User gleiche Baseline hat.
- Z-Transformation (Mittelwert abziehen und durch Standardabweichung teilen).

Bias-Variance Tradeoff der verschiedenen Ansätze

- Bei memorybasierten kollaborativen Methoden wird kein Modell angenommen, Algorithmen arbeitet direkt mit User-Artikel-Interaktionen. → tiefen Bias aber hohe Varianz.
- Bei modellbasierten kollaborativen Methoden wird (ziemlich freies) Modell für Interaktionen zwischen User und Artikel angenommen. → theoretisch höheren Bias (falsche Annahme), aber geringere Varianz.
- Bei inhaltsbasierten Methoden verwendet man stärker eingeschränkte Modelle (Features für User und/oder Artikel vorgegeben). → tendenziell höchster Bias, aber geringste Varianz.

Kaltstartproblem

- Kollaborative Ansätze verwenden keine Informationen über User oder Artikel.
- Empfehlungen werden genauer, je mehr User mit Artikeln interagieren.
- Da nur vergangene Interaktionen für Empfehlungen berücksichtigen, leidet kollaborative Filterung unter Kaltstartproblem: ist unmöglich, neuen User etwas zu empfehlen oder alten Usern neuen Artikel empfehlen.
- Viele User oder Artikel haben zu wenige Interaktionen, um effizient behandelt zu werden.
- Inhaltsbasierte Methoden haben kein Kaltstartproblem, da auf Eigenschaft User/Artikel Vorschläge machen. Einzig wenn User/Produkte mit neuen Merkmalen dazukommen, braucht Zusatzinformationen.

Zusammenspiel der Ansätze

Workarrounds für Kaltstartproblem bei kollaborativen Ansätzen:

- Zufällige Empfehlung von Artikel an neue User oder von neuen Artikeln an zufällige User (Zufallsstrategie).
- Empfehlung von allgemein beliebten Produkten an neue User oder von neuen Produkten an aktivsten User.
- Kombination mit anderen Methode (z.B. inhaltsbasierte) für frühe Lebensphase neuer User
  ⇒ Grösser System basieren i.d.R. nie auf einzigen Recommender, sondern versch. kombiniert Ansätze ↓

Hybride-Ansätze

- Modelle unabhängig trainieren und Vorschläge kombinieren.
- Bsp: Netflix verwendet Content-Features (Genre, Regisseur) und Nutzungsdate personalisierte Empfehlung

Evaluierung von Recommender Systemen

Auch Recommender Systeme müssen evaluiert werden – je nach Ziel mit unterschiedlichen Metriken:

- **Genauigkeit:** MAE/RMSE: Vorhersage vs. tatsächlich, Precision@k, Recall@k: relevante Treffer in Top-N
- **Ranking-Qualität:** Hit Rate (Top-k mit mindestens einem relevanten Item), ARHR (Average Reciprocal Hit Rate), NDCG (Discounted Cumulative Gain, bewertet Position der relevanten Items)
- **Diversität & Serendipität**: Intra-List-Diversity (Durchschnittlich Unähnlichkeit Items), Serendipitätsmass
- **Neuheit/Abdeckung**: Katalogabdeckung (Anteil Artikel, die min. einmal empfohlen), Item Popularity
- **Erklärbarkeit**: Empfehlungen nachvollziehbar/begründet → qualitative Bewertung, Nutzervertrauen steigt
- **Business-Metrik:** CTR (Click-Through-Rate, Anteil Empfehlung mit Klick), Conversion Rate, Umsatz

Offline vs. Online Evaluierung

- **Offline**: Daten Trainings-/Testsets splitten. Klassisch Metrik möglich, aber keine Garantie Praxistauglichkeit
- **Online**: A/B-Tests mit echten Nutzern. Verhalten & Akzeptanz im realen System messbar. Nur so kann sichergestellt werden, dass der Recommender wirklich nützt
- Ein Recommender, der offline gut aussieht, aber online ignoriert wird, erfüllt sein Ziel nicht.

Was macht eine gute Empfehlung aus?

- Relevanz: passt zur Nutzerpräferenz
- Neuheit: zeigt bisher unbekannte Inhalte
- Diversität: deckt mehrere Interessen ab
- Serendipität: überraschend & interessant
- Erklärbarkeit: nachvollziehbar für den Nutzer
- Vertrauen: Nutzer verlässt sich auf die Vorschläge

Nicht Menge zählt, sondern Qualität. Nicht immer Top-10! Lieber Empfehlung über Relevanz-Schwellenwert.

Generalisierte additive Modelle (GAM)

- GAM erweitert GLM indem einige/alle linearen Terme linearen Prädiktor durch **glatte Funktionen** ersetzen: $\eta_i = \beta_0 + \sum_{k=1}^{m} x_i^{(k)} \beta_k \rightarrow \eta_i = \sum_{k=1}^{m} f_k\langle x_i^{(k)}\rangle$. Analyse grafisch mit geschätzten Funktionen: Glatte Kurve partiellen Residuen-Plot
- Schätzen unbekannte Funktionen **Backfitting-Algorithmus** → Zielvariable Y_i durch Pseudo-ZV ersetzen und gewichtete Glättung wie **IRLS**-Algorithmus durchführen. GAM gut geeignet, da Transformation **nicht-parametrisch** schätzen.
- Hauptzweck für Einsatz von GAM ist **Aufdeckung** geeigneter **Transformationen** bei erklärenden Variablen im GLM.

```
library(gam); baby.gam <- gam(Survival ~ lo(Weight) + lo(Age) + lo(pH), family = binomial, bf.maxit=100, data = baby)
par(mfrow=c(2, 3)) #2*3=Anz.Erkl.Var.; plot(baby.gam, se = T, residuals = T)  ↓ Bessere Darstellung für Variable EXTRP
lo() geht für Faktorvariablen nicht, diese regulär ohne lo() Modell nehmen. plot(b.gam, residuals = T, terms = "lo(EXTRP)")
Faustregel 1: Gerade passt (nicht) zwischen gestrichelte Vertrauensband, gem. Faustregel keine Transformation nötig.
Faustregel 2: summary(baby.gam) #Anova bei Nonparametric: wenn p significant, dann transformieren
baby$tWeight <- ifelse(baby$Weight < 1000, 0, baby$Weight-1000) #bis zu einem gewissen Schwellenwert Prädiktor
baby$tpH <- ifelse(baby$pH < 7.27, 0, baby$pH-7.27) #beibehalten und danach diesen transformieren, Hockey Stick
glm4 <- glm(Survival ~ Weight + tWeight + Age + Apgar1 + Apgar5 + pH + tpH, family = binomial, data = baby)
DaR$HRtr <- ifelse(DaR$HR < 12.5, DaR$HR, 12) #Mehrfacher Hockeystick mit HR → HRtr und HRhigh
DaR$HRhigh <- as.integer(DaR$HR > 19.5); DaR$fF <- as.factor(cut(DaR$F, breaks=c(0,0.1,0.4,0.7,1))) #kat. Var.
DaR.qP <- glm(RDR ~ sVH + sPOP + IAR + HRtr + HRhigh + fF + IND, data = DaR, family = quasipoisson)
```


Advanced Regression Modelling ARM – ZF

- **Inferenz**: Gibt es signifikanten Zusammenhang? Wie sicher sind wir mit geschätzten Koeffizienten?
- **Modellwahl**: Welche Variablen haben grossen Anteil an Erklärung der Zielgrösse?
- **Vorhersagen**: Vorhersagen mit Genauigkeit?
- **Goodness-of-Fit:** Beschreibt Modell Daten adäquat?
- **Residuenanalyse**: Stimmen Modellannahmen?

Zielgrösse Y (ihre Verteilung)	Modell
Stetig mit konstanter Varianz	Multiple lineare Regression
Binär	Logistische Regression
Anzahl	Poisson Regression
Stetig mit proportionaler Varianz	Gamma Regression
Zensierte Wartezeit stetig	Weibull-/Cox-Regression
Zensierte Wartezeit diskret	Grupp. prop. Hazardmodell

Generalisierte lineare und additive Modelle (GLM, GAM) → par(mfrow = c(2, 3)); plot(Y ~ ., data = data)

Logistische Regressionsmodell

- Zielvariable Y ist binär (0/1) und π = Wahrscheinlichkeit, dass der Wert 1 angenommen wird ($\pi = P\langle Y = 1\rangle$) ☞ D.h.: Modellieren Y mittels Bernoulli-Verteilung mit unbekanntem Parameter π. π wird aus Daten geschätzt.
- Idee: bedingte Erwartungswert Zielgrösse, $\mathbb{E}\langle Y_i|x_i\rangle = \pi\langle x_i\rangle$ soll erkl. Var. x_i abhängen. $\mathbb{E}\langle Y_i|x_i\rangle = P\langle Y_i = 1|x_i\rangle$. Ansatz $\pi\langle x_i\rangle = \beta_0 + \beta_1 x_i$ funktioniert nicht, da $\pi\langle x_i\rangle$ Werte [0, 1] muss. ☞ brauchen adäquate Transformation auf $[0, 1]$
- Suche nicht-linearen Funktion G, welche Wertebereich berücksichtigt: $\pi\langle x_i\rangle = G\langle\beta_0 + \beta_1 x_i\rangle$ bzw. $G^{-1}\langle\pi\langle x_k\rangle\rangle = \beta_0 + \beta_1 x_k$
- Wie G aussehen: Funktion G meistens S-Form. Beliebte G sind unten ersichtlich. $\eta = g\langle\pi\rangle$ ist der lineare Prädiktor.

	$G\langle\eta\rangle$	$g\langle\pi\rangle$	Name
Logisitsche Verteilung	$exp\langle\eta\rangle/(1 + exp\langle\eta\rangle)$	$log\langle\pi/\langle 1-\pi\rangle\rangle$	Logit-Modell; log Regression
Normal-Verteilung	$\Phi\langle\eta\rangle$	$\Phi^{-1}\langle\pi\rangle$	Probit-Modell
Extremwert-Verteilung	$1 - exp\langle -exp\langle\eta\rangle\rangle$	$log\langle -log\langle 1-\pi\rangle\rangle$	Komplementäres Log-Log-Modell

- Link-Funktion $g(\cdot)$ ist identisch zur inversen kumulativen Verteilungsfunktion $G^{-1}(\cdot)$.
- Beiden ersten Modelle sind theoretisch sehr schwer auseinander zuhalten ausser in ihren äussersten Extremen
- Dritte Link-Funktion **nicht symmetrisch** um $\pi = 0.5$ ☞ **entscheidend**, was 1 (= **Erfolg**) und 0 (=**Misserfolg**) in **ZV** ist.
- **Logistische** Verteilung (Logit-Modell) wird i. A. **bevorzugt**, weil es eine **einfache Interpretation** der Parameter erlaubt.
- Kann anstelle Ereigniseintretenswahr' **Wettverhältnis** (**odds**) für (Nicht-)Erfolg: $log\langle odds\rangle = log\langle\frac{\pi_i}{\langle 1-\pi_i\rangle}\rangle = \beta_0 + \sum_{j=1}^{m}\beta_j x_i^{(j)}$
- Somit haben Koeffizienten β gleiche Bedeutung bei Beschreibung der log-odds wie bei der multiplen lin. Regression.

Die logistische Regression (**Logit-Modell**) ist durch folgende drei Elemente bestimmt:

1. ZV Y_i Bernoulli mit Erfolgswahrscheinlichkeit π_i.
2. Lineare Prädiktor $\eta_i = \beta_0 + \beta_1 x_i^{(1)} + \cdots + \beta_m x_i^{(m)}$
3. Lineare Prädiktor η_i ist mit Erfolgsparameter über logistische Funktion verlinkt: $\eta_i = log\langle\frac{\pi_i}{\langle 1-\pi_i\rangle}\rangle\ \pi = G\langle\eta\rangle = \frac{exp\langle\eta\rangle}{1+exp\langle\eta\rangle}$

Parameterschätzung: Um geeignete Schätzung herzuleiten ☞ Maximum Likelihood. Verfahren wird für Fall erläutert, dass Zielvariable unabhängig und binomial verteilt, verwenden logistische Link und nur eine erklärende Variable.

- Um Maximum-Likelihood-Schätzung von β_0, β_1 zu erhalten, nehmen wir partiellen Ableitungen von $l\langle\underline{\beta}|\underline{y}^*\rangle$ bezüglich β_0 und β_1 durch Anwendung der Kettenregel. ML-Schätzung erhalten wird durch Gleichung $\sum_k(y_k^* - m_k\hat{\pi}_k)x_k^{(j)} = 0,\ j = 0,1$
- Da $\pi_k = G\langle\underline{x}_k^T\underline{\hat{\beta}}\rangle$ eine nichtlineare Funktion der unbekannten Parameter $\underline{\beta}$ ist, ist das **Gleichungssystem nicht linear** → iterativ Algorithmen (Gradientenverfahren) wie z.B. der IRLS-Algorithmus müssen verwendet werden.
- Kommt **nicht darauf** an, **ob** wir **binären Daten oder gruppierten** für Anpassung verwenden. Schätzwerte sind gleich
- Anpassung mit binären Zielvariablen: MAC.glm1 <- glm(**Y** ~ **ac**, family = binomial, **MAC**) #Bei Default wird Logit-Modell
- Anpassung mit gruppierten Daten: MAC.glmG1 <- glm(cbind(**Y, m-Y**) ~ **ac**, family = binomial, data = **MAC1**) #angepasst
- **Achtung**: Anpassung ist Resultat nichtlinearen Gleichungssystems ☞ kann vorkommen, dass folgende erscheint: Warning message: glm.fit: algorithm did not converge. Algorithmus hat nicht konvergiert ☞ **kann Schätzwerten nicht trauen**
- Leider nicht möglich Konvergenz Algorithmus sicherzustellen, ausser max. Anzahl Schritte erhöhen glm(maxit = 1000).

Binäre Regression – Besonderheiten

- Punkte liegen auf zwei Kurven wegen binären Werten. Hat nichts mit Verletzungen Modellannahmen zu tun
- fast unmöglich, Ausreisser zu erkennen
- Situation kann durch Datenaggregation erheblich verbessert werden
- Bei binären ZV ist Normal-QQ-Plot oft bedeutungslos. Selbst wenn Residuen nicht normalverteilt, sieht Plot gut aus.
- Leverage-Werte oft sehr klein. Können nicht Faustregel beurteilen, da zu viele Beob. verglichen Anz. unbek. Parameter
- **Beispiel:** Geschätzte Modellgleichung für Erwartungswert binomial Logit-Modell mit einer erkl. Variable weight:
- $\hat{\eta}_i = -4{,}561 + 0{,}0051 \cdot weight_i \rightarrow \hat{\pi}_i = \hat{\mu}_i = \frac{exp\langle\hat{\eta}_i\rangle}{1+exp\langle\hat{\eta}_i\rangle}$, Werte $-4{,}561, 0{,}0051$ sind β_0, β_1 Koeffizient aus Summary Output

Koeffizient log. Regression = **Änderung log. Wettverhältnis** → exp(0.0051)–1 #0.005 ← Wenn Erhöhung weight um 1 Einheit, führt dazu dass das Log. Odds Wettverhältnis für zutreffen bspw. Infekt (Wert 1) sich um 0.5 % (0.005*100) erhöht.

sunflowerplot(x = **birth$weight**, y = **birth$Y/birth$m**, number = **birth$m**) | Verteilungsannahme: $(Y_i|X_i) \rightarrow (infect_i|wcc_i)$
x = erkl. Variable, y = Y Anz. zutreffende Argumente, m = Total Anz. | ZV unabhä. bin/bern: Y~Bin(m=**6**, $\pi_i = P(infect_i = 1)$
x <- seq(min(**birth$weight**), max(**birth$weight**), length=50) #Prognose | $\mathbb{E}\langle Y_i\rangle = m_i\pi_i \rightarrow \mathbb{E}\langle\frac{\mu_i}{m_i}\rangle = \pi_i \equiv \mu_i|\eta_i = log\langle\frac{\pi_i}{1-\pi_i}\rangle = \beta_0 + \beta_1 \cdot wcc_i$
mu.p <- predict(**birth.glm1**, newdata = data.frame(**weight** = x), | Ab welche Wert erkl. Var. wcc_i haben 80% Erfolg
type = "response"); lines(x, mu.p, col = 'blue', lwd = 4, lty = 6) | $wcc_i = \frac{1}{\beta_1} \cdot \left[log\langle\frac{\pi_i}{1-\pi_i}\rangle - \beta_0\right], \pi_i = 0.8, \beta_k$ =summary

Asymptotisch Normalverteilt (Wald Asymptotik): Theorie ML-Schätzer: Jeder ML-Schätzer asymptotisch normalverteilt.

- asymptotisch im Sinne Zentralen Grenzwertsatzes. D.h. Approximation umso besser, je grösser Anzahl Beobachtungen.
- Können deshalb Summary-Output gleich interpretieren wie bei Kleinsten-Quadrate. Testresultate, VI gelten **nur approx**.

Das umfassende Modell – Die einfache Exponentialfamilie

Sowohl das lineare wie auch das logistische Regressionsmodell haben folgende Gemeinsamkeiten:

- Erklärende Variablen $x^{(1)}, x^{(2)}, \dots, x^{(m)}$
- eine Zielvariable Y mit Erwartungswert $E(Y) = \mu$
- eine Link-Funktion g, sodass $g\langle\mu\rangle = \underline{x}^T\underline{\beta}$
- eine Verteilung für Variabilität in Zielvariablen Y.

- Klassische lin. Reg. Identität $g\langle\mu\rangle = \mu$ als Link und Normalverteilung mit $\mathbb{E}\langle Y_i\rangle = \mu_i$ und konstanter Varianz σ^2 verwendet.

Zufallsvariable Y mit Erwartungswert $\mu = \mathbb{E}\langle Y\rangle$ und Dispersion ϕ hat Verteilung in einfachen Exponentialfamilie, falls

- Dichte oder Wahrscheinlichkeitsfunktion von Y geschrieben werden kann als $f\langle y_i; \mu_i, \phi\rangle = \exp\left\{\frac{y_i b(\mu_i) - c(\mu_i)}{\phi/w_i} + d\langle y_i; \phi, w_i\rangle\right\}$
- $b\langle\,\rangle$, $c\langle\,\rangle$ **spez. Verteilung** aus **Exp.familie**
- $d\langle\,\rangle$ **Normalisierung** Dichte/Wahrscheinlichkeitsfunktion auf **1**
- Dispersionsparameter ϕ Teil der Varianz von Y
- «Variable» w_i ist für jede Beobachtung i eine bekannte Zahl. w_i kann unter den versch. Beobachtungen unterschiedlich sein. Interpretation als Gewicht. Wird in diesem Modul**teil** verwendet, um Binomialverteilung mit $m_k > 1$ abzudecken.

- Kann zeigen, dass für Erwartungswert von Y gilt $\mu_i = E\langle Y_i\rangle = \frac{c'\langle\mu_i\rangle}{b'\langle\mu_i\rangle}$, wobei $c'\langle\mu_i\rangle, b'\langle\mu_i\rangle$ Ableitung nach μ sind
- Varianz ist Produkt positiven Funktion $V\langle\mu_i\rangle = \frac{1}{b'\langle\mu_i\rangle}$ von Erwartungswert, Dispersion ϕ und Gewicht w_i: $var\langle Y_i\rangle = \frac{\phi}{w_i} V\langle\mu_i\rangle$

Gamma-Verteilung (Verallgemeinerung Exponentialverteilung)

- Zeit zwischen zwei konsekutiven Ereignissen, die Poisson verteilt sind, ist exponentialverteilt.
- beide sind geeignet, um Betragsgrössen wie (Überlebens-)zeiten, Schadenshöhen, Verluste oder Kosten zu modellieren
- Dichte gammaverteilten ZV Y_i ist $f\langle y, \alpha, \beta\rangle = e^{-\beta \cdot y} \cdot y^{\alpha-1} \cdot \frac{\beta^\alpha}{\Gamma\langle\alpha\rangle}$ mit $\alpha, \beta > 0$ auf Definitionsbereich $y \in [0, \infty)$
- Der Parameter α ist **Formparameter** und β ist die **Rate** ($1/\beta$ ein **Skalenparameter**).
- $\Gamma\langle\,\rangle$ ist die Gamma-Funktion, eine Verallgemeinerung der Fakultätsfunktion auf die reellen Zahlen.
- $\alpha = 1$, $\beta = \lambda$ → **Exp.verteilung** mit λ: $T \sim Exp\langle\lambda\rangle$
- $\alpha = \frac{m}{2}$, $\beta = \frac{1}{2}$ → χ^2**-Verteilung** mit m Freiheitsgraden.
- m positive ganze Zahl, dann wird Gamma-Verteilung mit $\alpha = m$ und $\beta = \lambda$ auch **Erlang-Verteilung** genannt.
- **Summe von zwei exponentialverteilten Z.V. ist gammaverteilt**. Allgemein: Falls $T_i, i = 1, 2 \dots, m$ unabhängig $\sim Exp\langle\lambda\rangle$ dann ist Summe $T_1 + T_2 + \dots + T_m \sim \Gamma\langle\alpha = m, \beta = \lambda\rangle$. Wenn wir annehmen, dass Wartezeit in einer Warteschlange unabhängig exponentialverteilt mit Parameter λ ist, dann ist Zeit zur Bedienung von m Kunden gammaverteilt.

Zusammenstellung wichtigsten Grössen für einige der populärsten Verteilungen aus der einfachen Exponential-Familie. → →

Verteilung	Wertebereich Y	$\mathbb{E}\langle Y\rangle = \mu$	$var\langle Y\rangle$	$b\langle\mu\rangle$	$V\langle\mu\rangle$	ϕ	w
Gauss $\langle\boldsymbol{\mu}, \boldsymbol{\sigma^2}\rangle$	$(-\infty, +\infty)$	μ	σ^2	μ	1	σ^2	1
Binomial $\langle\boldsymbol{m}, \boldsymbol{\pi}\rangle$	$\frac{(0,1,2,\dots,m)}{m}$	π	$\frac{\pi(1-\pi)}{m}$	$\log\langle\frac{\mu}{1-\mu}\rangle$	$\mu(1-\mu)$	1	m
Poisson $\langle\boldsymbol{\lambda}\rangle$	0,1,2, …	λ	λ	$\log\langle\mu\rangle$	μ	1	1
Gamma $\langle\boldsymbol{\alpha}, \boldsymbol{\beta}\rangle$	$(0, +\infty)$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$	$-\frac{1}{\mu}$	μ^2	$\frac{1}{\alpha}$	1
Invers Gauss	$(0, +\infty)$	μ	$\frac{\mu^3}{\lambda}$	$-\frac{1}{\mu^2}$	μ^3	$\frac{1}{\lambda}$	1

Das generalisierte lineare Modell – GLM

- **1 Verteilungselement:** Verteilung Zielvariable Y_i, gegeben erklärenden Variablen $\underline{x}_i$, ist Mitglied der einfachen Exponentialfamilie mit Erwartungswert $\mathbb{E}\langle Y_i|\underline{x}_i\rangle = \mu_i$ und Dispersion ϕ. Zielvariable $Y_i, i = 1, \dots n$, ist unabhängig verteilt.
- **2 Strukturelement:** Erwartungswert μ_i wird zum linearen Prädiktor $\eta_i := \underline{x}_i^T\underline{\beta}$ der erklärenden Variablen mittels einer oft nichtlinearen Funktion $g\langle\,\rangle$ in Beziehung gebracht: $g\langle\mu_i\rangle = \eta_i = \underline{x}_i^T\underline{\beta}$. Funktion $g\langle\,\rangle$ wird Link-Funktion genannt.

Link-Funktion

Ihre **Wahl** ist zu einem **gewissen Teil recht willkürlich**, sofern sie folgende grundlegende Eigenschaft besitzt:

- Inverse Link-Funktion $g^{-1}\langle\,\rangle$ soll Werte linearen Prädiktor $\eta = \underline{x}^T\underline{\beta}$ auf Raum der möglichen Werte abbilden, die Erwartungswert von Y annehmen kann. Funktion $g^{-1}\langle\,\rangle$ **muss unmögliche Werte vermeiden**. Link-Funktionen:
  - $g\langle\mu\rangle = \mu$, wenn $\mu = \mathbb{E}\langle Y\rangle$ keinen Einschränkungen unterliegt,
  - $g\langle\mu\rangle = \log\langle\mu\rangle$, wenn $\mu = \mathbb{E}\langle Y\rangle > 0$ sein muss,
  - $g\langle\mu\rangle = \text{logit}\langle\mu\rangle := \log\langle\mu/(1-\mu)\rangle$, falls $\mu = \mathbb{E}\langle Y\rangle$ zwischen 0 und 1 liegen muss.

Kanonische Link-Funktion

- Für jede Verteilung aus einfachen Exponentialfamilie gibt spezielle **Link-Funktion**, die im gewissen Sinn **natürlich** ist.
- Führen Erwartungswert μ in kanonischen Parameter $\theta := b\langle\mu\rangle$ über, bzw. wird angenommen, dass Effekte linearen Prädiktors linear auf kanonischen Parameter wirken, $\theta = \eta$. Diese Link-Funktionen nennt man kanonische Link-Funktionen.
- Also entspricht $b\langle\mu\rangle$ gerade dem kanonischen Link für die jeweilige Verteilung. Siehe Tabelle oben Spalte $b\langle\mu\rangle$
- ACHTUNG: kanonische Link für Gammaverteilung erfüllt Anforderungen nicht. Genau Anpassungsresultat überprüfen!

Link-Funktionen, die in R implementiert sind

- Kanonische Link wird mit D bezeichnet: (Spaltenname entspricht Verteilung (= family) in R, Zeilenname dem Namen der Link-Funktion
- z.B. glm(**Y~X**, family=**poisson**(link = **identity**))

	binomial	gaussian	Gamma	inverse.gauss	poisson	quasi
logit	D					●
probit	●					●
cauchit	●					
cloglog	●					●
identity		D	●	●	●	●
inverse		●	!D!	●		●
log	●	●	●	●	D	●
$\mathbf{1}/\boldsymbol{mu^2}$				D		●
sqrt					●	●

Poisson-Verteilung/Regression

- Bei Poisson-Verteilung ist Erwartungswert gleich Varianz → Normalverteilung kann nicht angenommen werden, da mit höheren $\mathbb{E}(X)$ dann auch eine höhere Varianz vorhanden ist.
- Bei Schätzungen sollte man ganze natürliche Zahlen nehmen, da die Zielgrösse immer ganze Zahlen (Anzahlen) hat.

- Zielgrösse: $Hurricanes_i = Anzahl\ Wirbelstürme\ im\ Jahr\ i$, $Hurricanes_i \sim \boldsymbol{Pois(\lambda_i)}$, unabhängig mit $\mathbb{E}(Y_i) = \mu_i$
- Struktur: $log(\lambda_i) = \beta_0 + \beta_1 \cdot SOI_i + \beta_2 \cdot NAO_i + \beta_3 \cdot SST_i$, Link: log = kanonischer Link, linearer Prädiktor $log(\mu_i) = \eta_i$
- coef(**pois.log.glm**)["**SOI**"] #0.0618, → exp(coef(**pois.log.glm**)["**SOI**"]) #1.063 → 1.063-1 = 0.063 → Wenn SOI um eine Einheit erhöht wird, dann vergrössert sich die erwartete Anzahl Wirbelstürme um 6.3 %.

Poisson-Regression: Annahme: lineare Prädiktor wirkt gemäss einem Potenzgesetz auf Erwartungswert λ ein.

- $0 < \beta$ → Je grösser der Wert in der Variable, desto grösser ist der Faktor (Faktor = x^β)
  - $0 < \beta < 1$ → monoton steigende, konvexe Funktion
  - $1 < \beta$ → monoton steigende, konkave Funktion
- $\beta < 0$ → Je grösser Wert in Variable, desto kleiner (d.h. nahe bei 0) ist Faktor → monoton fallende, konvexe Funktion

Gamma-Regression $\boldsymbol{exp(a+b+c) = exp(a) \cdot exp(b) \cdot exp(c)}$ | $\boldsymbol{(exp(a))^b = exp(a \cdot b)}$

- Zielvariable Y_i ist $Y_i \sim Gamma(\mu, \phi)$ verteilt → $\mathbf{ClmAmt_i} \sim Gamma(\mu, \phi)$, wobei $log(\mu_i) = \beta_0 + \beta_1 \cdot \boldsymbol{Bluebook_i} + \dots + \beta_m \cdot \boldsymbol{DensityUrbans_i}$, log-Link bildet multiplikative Skala ab und stellt sicher, dass $\mathbb{E}(X)$ nur positive Werte annehmen kann
- $\mu_i = e^{\beta_0} \cdot e^{\beta_1 \cdot \boldsymbol{Bluebook_i}} \cdot \dots \cdot e^{\beta_m \cdot \boldsymbol{DensityUrbans_i}}$ coef(**gamma.log.glm**) #0.0618, → exp(coef(**gamma.log.glm**)) #1.063 → 1.063-1 = 0.063 → Wenn Bluebook um eine Einheit erhöht wird, dann vergrössert sich die erwartete ClmAmt um 6.3 %.

qchisq(c(0.**025**, 0.**975**), **51**) #[33, 72] Gamma ist exp., wenn $\phi = 1$ → Res. deviance (18.05 on **51** df) im Annahmebereich

Statistische Inferenz und Variablenselektion

- p ML-Schätzgleichungen $\underline{\beta}$ werden durch Differenzierung Log-Likelihood-Funktion bzgl. p-Parameter hergeleitet. (Ableitungen **score** Funktion). Gleichungssystem wird mit iterativen gewichteten Kleinste-Quadrate-Verfahren (iteratively reweighted least-squares; **IRLS**) gelöst. Gewichte sind abhängig von Lösung, was zu einem iterativen Vorgehen führt.
- Mit $\underline{\hat{\beta}}$ erhält angepasste Werte $\hat{\mu}_i$. (mult. Reg. $\hat{y}_i$) → **schätzen Erwartungswert** und **nicht Responsewerte** predicten

Schätzung des Dispersionsparameters (I. A. muss der Dispersionsparameter ϕ auch geschätzt werden)

- Kann durch Maximierung Likelihood geschehen. Oder wie bei multiplen lin. Regr. durch erwartungstreuen Schätzer.
- Im Falle der Binomial- und der Poisson-Verteilung ist $\phi = 1.$, Für den Gamma-Fall: siehe später

Asymptotische Verteilung und Wald-Typ-Inferenz

- Approx. besser, je grösser Anz. Beob. n ist, weil Approx. nach ZGWS. Stichprobenumfang hängt Datensatz/Modell ab.
- In Praxis heisst das für GLM $\underline{\hat{\beta}} \overset{a}{\sim} \mathcal{N}_p\langle\underline{\beta}, \phi^*(\boldsymbol{X^TWX})^{-1}\rangle$ mit $\boldsymbol{W}_{ii} = \frac{f w_i}{V\langle\hat{\mu}_i\rangle(g'\langle\hat{\mu}_i\rangle)^2}$ wobei $\boldsymbol{W}$ Diagonalmatrix und $\phi^* = 1$ bei Binomial- und Poissonverteilung, sonst $\phi^* = \hat{\phi}$ → Inferenz basierend auf diesem Verteilungsresultat = Wald-Typ-Inferenz
- VI **approx**: $\underline{\hat{\beta}} \pm q \cdot se\langle\underline{\hat{\beta}}\rangle$, $se\langle\underline{\hat{\beta}}\rangle = diag(\sqrt{\phi^*(\boldsymbol{X^TWX})^{-1}})$. Falls $\phi^* = 1$ → q Quantil Normalverteilung, bei $\phi^* = \hat{\phi}$ t-Verteilung
- Vertrauensint. Wald: confint.default(**dat.glm**); coef(**dat.glm**)[1]+c(-1,1)*qnorm(**0.975**, 0, 1)*summary(**dat.glm**)$coef[1,2]
- family(**USH.glm**)$linkinv(coef(**USH.glm**)) #Retour transformieren Koeffizienten mit Link Funktion | ↖ von Hand Wald VI

Vergleich von geschachtelten Modellen

Mult. lin. Regression F-Test, um statistische Signifikanz einer oder mehrerer erkl. Var. Abzuklären:

- $H_0: \beta_{j1} = 0, \beta_{j2} = 0, \dots, \beta_{jq} = 0$, Alternative H_A: mindestens ein $\beta_{jk}, k = 1, \dots, q$, ist ungleich 0. Teststatistik $F := \frac{(SS_E^* - SS_E)/q}{SS_E/(n-p)}$
- $SS_E^* := \sum_{i=1}^n R_i^2$, r_i aus «kleinen» Modell, $(p-q)$ Koeffizienten | $SS_E := \sum_{i=1}^n R_i^2$, r_i aus «grossen» Modell, p Koeffizienten.

Residuen-Devianz – In multiplen linearen Regression werden

- Differenzen zwischen SS_E von versch. Modellen zum Vergleich. SS_E kann auch Schätzung Skalenparameters σ benutzen
- SS_E misst die Genauigkeit des Modells und ist zugleich Ausdruck, der in Methode Kleinsten Quadrate minimiert wird.
- Summe lässt sich auch mit Hilfe der Log-Likelihood ausdrücken und umformulieren, sodass Term nicht von ϕ abhängt.
- Der Term wird Residuen-Devianz genannt: $D\langle\underline{Y}; \underline{\hat{\beta}}\rangle := 2\phi\left(\ell\langle\underline{Y}; \phi\rangle - \ell\langle\underline{\hat{\mu}}; \phi\rangle\right)$

Devianz-Test: GLM-Generalisierung F-Test mit Ersetzung Quadrat-Summen durch Residuen-Devianz. Teststatistik $T_D := \frac{D\langle\underline{Y}; \underline{\hat{\beta}^r}\rangle - D\langle\underline{Y}; \underline{\hat{\beta}}\rangle}{\phi^*}$, $D\langle\underline{Y}; \underline{\hat{\beta}^r}\rangle$ Devianz reduziertes Modell, $D\langle\underline{Y}; \underline{\hat{\beta}}\rangle$ Devianz volles Modell und ϕ^* 1 oder Schätzung v. ϕ

- I. A. ist Verteilung von T_D unter Nullhypothese zu schwierig herzuleiten. **Asymptotisch** ist Teststatistik T_D unter H_0 χ_q^2-verteilt, q ist Anzahl Koeffizienten, die in H_0 0 gesetzt werden. Näherung ist je besser je grösser Stichprobengrösse n.

Normalverteilte Zielgrössen: Devianz-Teststatistik $T_D := \frac{S\langle\hat{\beta}^r\rangle - S\langle\hat{\beta}\rangle}{S\langle\hat{\beta}\rangle/(n-p)}$ wobei erwartungstreue Schätzer $\phi = \sigma^2$ anstelle ML-

- Nicht identisch mit F-Test (Faktor $1/q$ fehlt), P-Wert stimmen grossen Stichproben überein, weil $\mathcal{F}_{q,n-p} \to \frac{1}{q}\chi_q^2$ falls $n \to \infty$. → Praxis sinnvoll, normalverteilten Zielgrösse F-Test anstelle Deviianztest verwenden, da Verteilung F bekannt ist.

anova(**grossesModell**, **reduziertesModell**, test = "Chisq") #Wenn p-Wert kleiner als Sig.niveau, grosses Modell besser. Auf 5% Niveau besteht zwischen beiden Modellen kein signifikanter Unterschied, da P-Wert von 0.0697 grösser als 5% ist. Man kann davon ausgehen, dass kleinere Modell Daten ebenso gut beschreibt. Nehmen kl. Modell: Prinzip Occam's Rasor

Null-Devianz: Null Deviance bezieht sich auf Residuen-Devianz des Modells, das nur aus Achsenabschnitt besteht.

- Muss Differenz zwischen null deviance und residual deviance bilden und durch $\hat{\phi}$ teilen, sofern ϕ unbekannt ist.

- **Beispiel Narkos:** Null Deviance: 15.4334 on 5 degrees of freedom, Residual Deviance: 1.7321 on 4 degrees of freedom
- 1- pchisq((**15.4334 - 1.7321**)/1, df = **5 - 4**) #0.0002143061 #durch 1 teilen, wenn Pois/Binom, sonst durch $\hat{\phi}$
- Da P-Wert kleiner 0.05, muss Null-Hypothese, dass alle Koeffizienten ausser Achsenabschnitt 0 sind, verworfen werden. Folglich hat mindestens eine der erklärenden Variablen signifikanten Einfluss auf Erwartungswert der Zielvariable.

Notwendigkeit für Devianztest: Probleme Wald-Test: hat erheblichen Nachteil in **binären Regression**, wenn wahre Parameterwerte sehr weit von Null entfernt. Standardfehler werden sehr (zu) gross. Je weiter $\hat{\beta}_k$ von 0 entfernt, desto grösser Wald-Teststatistik für $H_0 : \beta_k = 0$. Ab gewissen Punkt wird sie wieder kleiner und geht gegen 0, wenn $\hat{\beta}_k$ unendlich gross wird. Folglich Null-Hypothese nicht signifikant. → **Hauck-Donner-Phänomen**

- Nehmen Devianztest anova(test = "Chisq") oder drop1() Hypothese $H_0 : \beta_k = 0$ oder «profile» Vertrauensintervalle.

Schätzung des Dispersionsparameters – Overdispersion (übermässige Streuung)

- Gem. Modell ist Residuen-Devianz ungefähr χ^2-verteilt. Somit gilt, dass mittleren Residuen-Devianz $\mathbb{E}\langle D/(n-p)\rangle \approx \phi$ und folglich können wir mit $D/(n-p)$ den Parameter ϕ schätzen. Dies ist **keine Maximum-Likelihood-Schätzung**

Im Fall von **gammaverteilten** Zielvariablen:

- können auch $D/(n-p)$ zur Schätzung von ϕ verwenden. Schätzer reagiert empfindlich auf angepasste Werte nahe 0!
- Stabileres Verhalten zeigt Momentenschätzer $CV := \frac{1}{n-p}\sum_{i=1}^{n}\left(\frac{Y_i-\hat{\mu}_i}{\hat{\mu}_i}\right)^2$ = empirisch Varianz Pearson-Residuen.

Modelle mit $\phi = 1$ (Binomial-, Poisson- und Exponential-Modellen ist $\phi = 1$ → braucht keine Schätzung!)

- Mit $D/(n-p)$ Modellunstimmigkeiten bzgl. Streuung aufdecken, die sich in ungewöhnlichen Werten von D zeigen.
  - wissen: $D \overset{a}{\sim} \chi^2$-verteilt mit $(n-p)$ df, **ausser bei binären Zielvariablen**. Testen auf Nullhypothese H_0: $\phi = 1$
  - Plausibilisierungsverfahren mit statistischem Test: Falls $D > q_{0.95}^{\chi^2_{n-p}}$ dann Overdispersion vorhanden auf 5 % Niveau.
- **Binäre** Zielvariabel gilt asymptotische Theorie nicht → Wenn immer möglich, binäre Daten aggregieren zu binomial ZV!

Bsp Narkos: Residual dev. 1.73 on 4 df, qchisq(0.95, df=4) #9.49, D kleiner als q, H_0 nicht abgelehnt, 1-pchisq(**1.73, 4**)
Da P-Wert von 0.324 > 0.05, gibt (k)eine Evidenz gegen Null-Hypothese "phi=1" und damit (k)eine Evidenz Overdispersion
Overdispersion: gibt mehr Variabilität in Daten, als (**Poisson)**-Verteilung zulässt. Dispersionsparameter sollte 1 sein. Bei Overdispersion ist er (deutlich) grösser als 1. Unter Nullhypothese, dass keine Overdispersion gibt, hat Devianz χ^2 Distr.

Mögliche Ursachen für Overdispersion

- **Falsche Strukturform** für Modell. **Nicht richtigen erklärenden Variablen** einbezogen oder haben **nicht transformiert** oder **nicht in der richtigen Weise kombiniert**.
- (Kleine Anzahl) **Ausreisser**
- **Zufallskomponente** Modell ist **unzureichend**. Binomialverteilung für Antwort Y ergibt sich, wenn **Erfolgswahr' π unabhängig** und für jeden Versuch **innerhalb Gruppe gleich ist**. Wenn Annahmen verletzt → kann Overdispersion sein

Berechnen von Vertrauensintervallen

- Beste Weg formale Schlussfolgerungen darzustellen und Beziehungen in Variablen aufzuzeigen, geschieht mit Konfint.
- «Standard»-Inferenzangaben basiert auf Tatsache, dass geschätzten Koeffizienten $\hat{\underline{\beta}}$ genähert Gauss-verteilt sind
- **Approx.** $100 \cdot (1-\alpha)$ % KI für Linearkombination $\psi = \underline{u}^T\underline{\beta}$ Koeffizienten bestimmen: $\underline{u}^T\hat{\underline{\beta}} \pm q \cdot se\langle \underline{u}^T\hat{\underline{\beta}}\rangle$ mit $se\langle \underline{u}^T\hat{\underline{\beta}}\rangle := \sqrt{\hat{\phi}^* \underline{u}^T(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\underline{u}}$, q Quantil aus $\mathcal{N}$-/t-Vert. und $u = p$ dim. Vektor festen Komponenten. confint.default(**glm**) #Wald-Typ

Vertrauensintervall erwartete Response-Wert bei $\underline{x}_o$: Schätzung: $\hat{\mu}_o = g^{-1}\langle \underline{x}_o^T\hat{\underline{\beta}}\rangle$. Standardansatz, beruht auf Fehlerfortpflanzungsgesetz. $g^{-1}\langle \underline{x}_o^T\hat{\underline{\beta}}\rangle \pm q \cdot se\langle\hat{\mu}_o\rangle$ mit $se\langle\hat{\mu}_o\rangle := \sqrt{\hat{\phi}^*\left(G'\langle \underline{x}_o^T\hat{\underline{\beta}}\rangle\right)^2 \underline{x}_o^T(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\underline{x}_o}$, q Quantil $\mathcal{N}$-/t-Verteilung

- **Approximation kann** sehr **unzuverlässig** sein → Vertrauensintervall teilweise ausserhalb Definitionsbereich von μ
- Resp <- predict(**fit.glm1**, se.fit = TRUE, type = "response"); Resp$fit + qnorm(**0.975**)*cbind(-Resp$se.fit, Resp$se.fit)
- Überzeugende praktische **Alternative** ist Vertrauensintervall für $\eta_o = \underline{x}_o^T\underline{\beta}$ zu bestimmen und dessen Endpunkte dann mit inversen Link g^{-1} rücktransformieren: $\underline{x}_o^T\hat{\underline{\beta}} \pm q \cdot se\langle\hat{\eta}_o\rangle$ mit $se\langle\hat{\eta}_o\rangle := \sqrt{\hat{\phi}^*\underline{x}_o^T(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\underline{x}_o}$
- Transformation Endpunkte mit g^{-1} erhalten Vertrauensintervall für $\mu_o = g^{-1}\langle\eta_o\rangle = g^{-1}\langle \underline{x}_o^T\underline{\beta}\rangle$: $g^{-1}\langle \underline{x}_o^T\hat{\underline{\beta}} \pm q \cdot se\langle\hat{\eta}_o\rangle\rangle$
- pLink <- predict(**fit.glm1**, se.fit = TRUE, type = "link"); h <- pLink$fit + qnorm(**0.975**)*cbind(-pLink$se.fit, pLink$se.fit);
- 1/(1+exp(-h)) #Rücktransformation; family(**MAC1.glm1**)$linkinv(pLink$fit + qnorm(**0.975**) * outer(pLink$se.fit, c(-1, 1)))
- pLink <- predict(**Chal.glm**, newdata = data.frame(**Temp = 31**), type="link", se=T) #Rücktransformation wie oben

Devianz basierte Vertrauensintervalle (VI) – profiling

- Vorgestellten VI beruhen auf Wald-Asymptotik. Diese Näherungen sind nicht immer gut genug. **Besser Devianz-Test:**
- Ein VI für β_k ist verknüpft mit Test $\beta_k = \tilde{\beta}_k$. Sei $\hat{\underline{\beta}}$ die ML-Schätzung von $\underline{\beta}$ und $\hat{\underline{\beta}}^{*k}$ jene von $\underline{\beta}$, wobei β_k auf den Wert $\tilde{\beta}_k$ fixiert ist. Devianz-Teststatistik für $\beta_k = \tilde{\beta}_k$ ist dann: $T_D = \frac{D(\underline{y};\hat{\underline{\beta}}^{*k}) - D(\underline{y};\hat{\underline{\beta}})}{\hat{\phi}} \overset{a}{\sim} \chi_1^2 \rightarrow \tilde{\beta}_k | T_D = \frac{D(\underline{y};\hat{\underline{\beta}}^{*k}) - D(\underline{y};\hat{\underline{\beta}})}{\hat{\phi}} \leq q_{1-\alpha}^{\chi_1^2}$ ist KI
- Falls ϕ tatsächlich geschätzt werden muss, wird in Praxis χ^2-Verteilung durch entsprechende F-Verteilung ersetzt.
- Nachteil: Berechnung i.A. aufwändiger als Waldtyp VI. Vorteil: Devianz oft gut, auch kleine Stichprobengrösse n
- confint(**fit.glm**) #Bemerkung im Output: Waiting for profiling to be done... → glm(…, maxit = **100**) #falls nicht konvergiert

Binäre Regression – Prognose? Zielvariable kann nur Werte 0 und 1 annehmen!

- Schätzung/Prognose für μ: $\hat{\mu}_o = g^{-1}\langle \underline{x}_o^T\hat{\underline{\beta}}\rangle$, approx VI für erwarteten Responsewert $\underline{x}_o$: $g^{-1}\langle \underline{x}_o^T\hat{\underline{\beta}} \pm q_{1-\alpha/2}^{\mathcal{N}} \cdot se\langle\hat{\eta}_o; \phi = 1\rangle\rangle$
- Weiteres Problem **class imbalance**, macht Over-/Undersamling. Oversamling sind Gift für Schätzung Wahrscheinlichkeiten μ, weil sie systematisch verzogenen Schätzungen führt. Vielfach reicht bei imbalance klassische ML-Schätzung.
- Falls Stichprobengrösse n sehr klein und imbalance extrem ist, dann alternative Schätzung verwenden

Variablenselektion

- Basiert Informationstheorie, Akaike Information Criterion: $AIC = -2(\max LogLikelihood) + 2 \cdot \#geschätzter\ Parameter$
- GLM AIC durch Devianzen ausdrücken: $AIC = -2 \cdot \ell(\hat{\mu}, \hat{\phi}) + 2 \cdot \#geschätzter\ Parameter$, $-2 \cdot \ell(\hat{\mu}, \hat{\phi}) = \frac{D}{\hat{\phi}} - 2 \cdot \ell\langle \underline{y}, \hat{\phi}\rangle$
- AIC/BIC? führen ähnlichen Entscheidungen. BIC bestraft jedoch grössere Modelle härter → **nicht** zwingend **Optimum**
- BIC: wenn verstehen will, welche Prädiktoren Beitrag leisten und wenn schlankes, gut interpretierbares Modell will.
- AIC: wenn Modell für Vorhersage von Erw.werten eingesetzt werden soll, verwendeten Prädiktoren weniger zentral sind.
- step(**grosses.glm**, *scope = list(lower = ~1, upper = formula(**grosses.glm**))*) #Default=both|AIC, kleinster Wert = bestes
- step(**grosses.glm**, k = log(nrow(**Datensatz**)), *direction = "backward"*) #für BIC
- Vorwärts: M0 <- glm(**Y ~ 1**, data = **DF**); step(M0, scope = list(lower = ~ 1, upper = ~ **x1+x2+x3+x4**), direction = "forward")
- Rückwärts: MF <- glm(**Y ~ x1 + x2 + x3 + x4**, data = **DF**); step(MF, direction = "backward")
- unter AIC Wert steht finale Modell in R Notation. Variablen mit Minus davor sind enthalten #direction = "both"

Prüfen Modellgüte von GLMs: Sollten Modellannahmen überprüfen bevor GLM als aussagekräftig akzeptieren und es verwenden!

- Überprüfung beruhen in erster Linie auf angepassten Werten und Residuen. **Ziel Residuenanalyse? Aufdecken systematische Abweichung Daten vom Modell und Identifikation ungenügend beschriebener Bereiche / Ausreisser**
- 4 Definition für GLM Residuum, die alle im Fall Normalverteilung identisch mit uns bekannten Definition Residuums sind.
- Einfachste und naheliebendste Definition Residuums ist **Response Residuen**: $R_i := Y_i - \hat{\mu}_i$, R_i sind i.A. nicht normalverteilt und ihre Varianz ist in erster Näherung $\phi V\langle\mu_i\rangle(1-H_{ii})/w_i$, Varianzfunktion $V\langle\mu_i\rangle$ ist i.A. nicht konstant
- Somit Definition sinnvoll **Pearson-Residuen**: $R_i^{(P)} := R_i \cdot w_i/\sqrt{V\langle\hat{\mu}_i\rangle}$ → ungefähr konstant Varianz, i.A. nicht normalvertlt.
- Basierend auf IRLS: **Working Residuen**: $R_i^{(W)} := R_i g'\langle\hat{\mu}_i\rangle$. → gewichtete Residuen; $\sqrt{w_i}R_i^{(W)} = \sqrt{\frac{w_i}{V\langle\hat{\mu}_i\rangle(g'\langle\hat{\mu}_i\rangle)^2}}(Y_i - \hat{\mu}_i)g'\langle\hat{\mu}_i\rangle \equiv R_i^{(P)}$
- Eigenschaft klassisch Residuen: **Unterschiede zwischen beobachteten und angepassten Responsewerten** messen
- Summe quadrierten Residuen zur Schätzung σ^2 verwenden und Unterschiede Summen quadrierten Residuen versch. Modellen zu Modellvergleich einsetzen → $R_i^{(D)} := sign\langle y_i - \hat{\mu}_i\rangle\sqrt{d_i}$, d_i =Summanden Residuen-Devianz.
- Wie Erfahrung zeigt, sind Unterschiede zwischen Devianz- und Pearson-Residuen üblicherweise sehr klein.

Tukey-Anscombe-Diagramm (Residuen-Plot): Pearson-Residuen $R_i^{(P)}$ gegen angepassten lineare Prädiktorwerte $\hat{\eta}_i$

- Wird verwendet, um auf **nicht erwartete Nichtlinearitäten** im linearen Prädiktor (oder allenfalls auf **unpassende Link-Funktionen**) und nebenbei auf **Ausreisser** aufmerksam zu werden.
- Abweichungen sichtbarer machen, Glätter hineinlegen, der bei passendem Modell um horizontale Null-Achse schwankt
- Beurteilung kann herausfordern, da bei nicht normalverteilten Zielvariablen irritierende Artefakte auftreten können.

Scale-Location-Diagramm: Wurzel absoluten Werte standardisierten Residuen gegen angepassten linearen Prädiktor.

- Untersucht **konstante Varianz** entlang angepassten Werte und relevanten Strukturen mit Glätter herausfiltern.

Half-Normal QQ-Plot: Indirekt zu prüfen, ob Verteilungsannahme zutrifft

- Falls **Verteilungsannahme** für Daten zutrifft, müssen Devianz-Residuen genähert normalverteilt → um Gerade streuen
- Fokus Abweichungen (Ausreisser rechts im Plot). Deshalb wird half-normal QQ-Plots verwendet.
- Half-normal QQ-Plot werden geordneten absoluten Devianz-Residuen gegen entsprechenden Quantile der Standardnormalverteilung aufgetragen, d.h. gegen die $\frac{n+i+\frac{1}{8}}{2 \cdot n+\frac{9}{8}}$-Quantile der Standardnormalverteilung $(i = 1,2,\ldots,n)$
- Devianz-Residuen nur genähert normvertlt. Möglich, trotz Verteilungsannahme «korrekt», keine lin. Struktur sichtbar ist!
- Darstellung nützlich um festzustellen, ob allfällige Overdispersion auf kleine Anzahl Ausreisser zurückzuführen ist.
- Bei binären ZV ist Normal-QQ-Plot oft bedeutungslos. Selbst wenn Residuen nicht normalverteilt, sieht Plot gut aus.
- source("ARM/RFn_Plot-glmSim.R"); par(mfrow = c(2, 4)); plot(**turb.glm**); plot.glmSim(**turb.glm**, *SEED = 184*)

Bootstap-Simulation: Idee: Erzeugen Beobachtungen y_i^* gem. Modell unter Verwendung neuer Zufallszahlen. Erzeugt n Zufallszahlen y_i^* entsprechend der verwendeten Verteilungsfamilie mit Erwartungswert $\hat{\mu}_i$ und Streuung $\hat{\phi}$.

- Berechnen GLM-Anpassung mit erkl. Var. aus Datensatz und neu generierten Responswerten y_i^*. Anschliessend wird glatte Linie TA-Diagramm bestimmt und graue Linie eingezeichnet. Wiederholen diese beiden Schritte $n_{rep} = 19$ mal
- Gibt Fälle, wo rote Glätter starke nichtlineare Strukturen zeigt, aber noch innerhalb grauen Bootstrap-Simulationen liegt.
- Andererseits überdecken Bootstrap-Simulationen jedoch die horizontale Nulllinie nicht! Was heisst das jetzt?
- Können daraus schliessen, dass es **keine** Evidenz gibt, dass Missspezifikation im linearen Prädiktor vorliegt.

Residual gegen Leverage — zu einflussreiche Beobachtungen → Ausreisser und/oder Hebelpunkt

- Messen Hebelwirkung einer Beobachtung mittels der Diagonalelemente H_{ii} der GLM-Hutmatrix
- Beobachtung wird als Leverage Punkt bezeichnet, wenn $H_{ii} > 2p/n$ ist, p Anzahl der Koeffizienten ist.
- Cook's distance misst wieder Einfluss einzelner Beobachtungen und ist definiert als $d_i^{\odot} := \frac{1}{p}\frac{R_i^2}{\hat{\phi}V(\hat{\mu}_i)(1-H_{ii})}\frac{H_{ii}}{1-H_{ii}}$, $1 \leq i \leq n$
- **Faustregel**: Beobachtungen mit einer Cook's Distance $d_i^{\odot}$ **grösser als 1** gelten als zu einflussreich

(Zeitliche) Korrelation in den Residuen: Residuen z. B. gegen Reihenfolge oder räumliche Anordnung Versuchseinheiten
Alternative: Streudiagramm Paare $(R_{i-1}^{(P)}, R_i^{(P)})$ plot(**fit**, type = "h") #Falls Beob. unab. streuen Punkte wild (weiss Rauschen)

Partielle Residuen-Plots (term plot). Ziel: Entdecken Nichtlinearitäten in erklärenden Grössen

- Auf x-Achse erkl. Variable $x_i^{(k)}$, auf y-Achse Pseudo-Zielvariable minus «Effekt» lin. Prädiktors ohne Komponente $x_i^{(k)}$:
- $Z_i - \left(\underline{x}_i^T\hat{\underline{\beta}} - \hat{\beta}_k x_i^{(k)}\right) = R_i^{(W)} + \hat{\beta}_k x_i^{(k)}$ → Falls Modell richtig, sollten dargestellten Punkte um Gerade streuen.

- **Residual Plot:** Da der rote Glätter klar inner-/ausserhalb stochastischen Fluktuation (graue Spaghetti) liegt, gibt es keine/starke Evidenz, dass Einfluss der erklärenden Variablen auf die Erwartungswerte falsch spezifiziert wurde.
- **Half-Normal QQ-Plot:** Da schwarzen Punkte inner-/ausserhalb stochastischen Fluktuation (graue Punkte) liegen, gibt es keine/klare Evidenz, dass die Verteilung inadäquat spezifiziert ist und auch noch Hinweise auf zwei Ausreisser Wegen Overdispersion haben grauen Punkte eine andere Steigung als schwarzen.
  Scale-Location-Plot: Da rote Glätter vollständig inner-/ausserhalb der stochastischen Fluktuation liegt (graue Spaghetti), gibt es keine/klare Evidenz, dass Varianz inadäquat spezifiziert wurde. Wegen Overdispersion liegen grauen Spaghettis auch viel tiefer als rote Glätter!
- **Residuen vs. Leverage:** Keine/Mehrere Beobachtungen haben Cook's Distance grösser als 1 und gibt/sind deshalb (keine) zu einflussreich(en Beoba). Bei den einen ist es bedingt durch Ausreisser, bei den anderen durch Hebelarme.
- **Fazit:** Alle Plots zeigen (keine)klare Evidenz, dass Modell Daten (nicht) adäquat beschreiben kann (Inferenz wertlos)

a <- 2*length(coef(**glm**))/nrow(**data**); abline(v=a, col=3)
Y: Ausreiser, X: Hebelpunkte

Behandlung von Unzulänglichkeiten: J.W. Tukey nannte sie First Aid Transformations

- Logarithmus-Transformation für Konzentrationen/Beträge,
- Wurzeltransformation für ZählDaten
- Logit für Anteile (Prozentzahlen/100): $\hat{y} = \boldsymbol{log}\langle\frac{y+0.005}{1.01-y}\rangle$
- Arcus-Sinus-Wurzel $\hat{y} = \boldsymbol{arcsin}\langle\sqrt{y}\rangle$
- Nicht anwenden Zeitvariablen/zu wenig Streuung
- First-Aid/Log Faktor 10