

8. Vorlesung

- Das Problem des Zusammenfassen von Sortierungen (Rankings)
- Anwendungen
- Erwünschte Eigenschaften
- Arrow's Satz der Unerfüllbarkeit aller Eigenschaften
- Methoden zum Zusammenfassen von Sortierungen

Problem des Zusammenfassens von Sortierungen

- m Kandidaten (auch “Alternativen” genannt)
 - $M = \{1, \dots, m\}$: Menge der Kandidaten
- n Wähler (auch “Agenten” oder “Richter”)
 - $N = \{1, \dots, n\}$: Menge der Wähler
- Jeder Wähler i hat eine Sortierung π_i von M
 - $\pi_i(a) < \pi_i(b)$ heißt, der i -te Wähler bevorzugt a vor b
 - Sortierung kann total oder partiell sein
- Das Problem des Zusammenfassens von Sortierungen:
Kombiniere π_1, \dots, π_n zu einer Sortierung σ von M , welche die “soziale Wahl” der Wähler repräsentiert.
 - Funktion zum Zusammenfassen der Sortierungen: $f(\pi_1, \dots, \pi_n) = \sigma$
 - σ kann eine totale oder partielle Ordnung sein

Beispiele

- **m klein, n groß**: Wahlen
(Mehrparteienparlament, Universität, Vorstände,...)
- **m mittel, n klein**: Konferenzprogramm-Kommittees, Sport
- **m groß, n klein**: Meta-Suche, Reisepläne, Restaurantauswahl

Anwendungen auf Web-Suche

- Meta-Suche
 - Kombiniere Ergebnisse von verschiedenen Suchmaschinen zu einer besseren Gesamtordnung
- Spam-Bekämpfung
 - Spam hat eine geringe Wahrscheinlichkeit weit vorn in zusammengefaßten Ordnungen aufzutauchen, auch wenn sie bei ein oder zwei Suchmaschinen vorn liegen
- Suche nach mehreren Termen
 - AND: kleine Ausbeute (Recall)
 - OR: geringe Genauigkeit (Precision)
 - Komplexe boolsche Anfrage: zu kompliziert für Durchschnittsanwender
 - Lösung: suche nach kleinen Teilmengen der Terme und fasse Rankings zusammen
- Kombination mehrerer Sortierungsfunktionen
 - Nutze verschiedene Ranking-Funktionen (z.B., VSM, PageRank, HITS, ...) und fasse sie zu einer Funktion zusammen

Anwendungen auf Datenbanken

- Sortiere Tupel in einer Datenbank bezüglich mehrerer Kriterien
 - z.B.: Wähle ein Restaurant mittels Küche, Entfernung, Preis, Qualität, usw.
 - z.B.: Wähle einen Flug nach Preis, Anzahl der Zwischenlandungen, Datum und Zeit, Bonusprogramm, usw.

Erwünschte Eigenschaft: Einmütigkeit

- **Einmütigkeit** (auch Pareto Optimalität):
Wenn alle Wähler Kandidat **a** vor Kandidat **b** bevorzugen ($\pi_i(a) < \pi_i(b)$ für alle i), dann soll auch σ **a** vor **b** bevorzugen ($\sigma(a) < \sigma(b)$).

a	c	a
b	a	c
c	b	b

a vor b = 3:0

Erwünschte Eigenschaft: Condorcet

- **Condorcet Kriterium** [Condorcet, 1785]:
 - Condorcet Gewinner: ein Kandidat a , der von der Mehrheit der Wähler vor allen anderen Kandidaten b bevorzugt wird (für alle b , # der Indizes i mit $\pi_i(a) < \pi_i(b)$ ist mindestens $n/2$).
 - Condorcet Kriterium: Wenn es einen Condorcet Gewinner gibt, soll σ an erste Stelle setzen ($\sigma(a) = 1$).

a	b	c
b	a	a
c	c	b

a vor b = 2:1, a vor c = 2:1

a	b	c
b	c	a
c	a	b

Kein Condorcet Gewinner

Erwünschte Eigenschaft: ECK

- **Erweiteres Condorcet Kriterium (ECK):**
 - Falls die Mehrheit der Wähler Kandidat **a** vor **b** bevorzugen (# der Indizes i mit $\pi_i(a) < \pi_i(b)$ ist mind. $n/2$), dann soll σ **a** vor **b** setzen ($\sigma(a) < \sigma(b)$).
 - Nicht immer realisierbar

a	b	c
b	a	a
c	c	b

$\sigma(a) < \sigma(b) < \sigma(c)$

a	b	c
b	c	a
c	a	b

Nicht realisierbar

ECK und Spam [Dwork et al. 2001]

- **Definition:** eine Seite p ist “spam” für ein Ranking π , falls es eine Seite q gibt, die nach p kommt, welche aber von der Mehrheit der menschlichen Auswerter vor p gesetzt wird.
- **Annahme:** für alle Paare p, q gilt, die Mehrheit der Auswerter stimmt mit der Mehrheit der Suchmaschinen in der Ordnung von p und q überein.
- **Schlußfolgerung:**
 - Spam Seiten sind immer “Condorcet Verlierer”
 - Falls eine Zusammenfassungsmethode ECK erfüllt, eliminiert sie Spam.

Erwünschte Eigenschaft : Unabhängigkeit von irrelevanten Alternativen

- **Unabhängigkeit von irrelevanten Alternativen:**

Relative Ordnung von a und b in σ sollen nur von der relativen Ordnung von a und b in π_1, \dots, π_n abhängen.

- z.B.: falls $\pi_i = (a b c)$ sich in $(a c b)$ ändert, sollte sich die relative Ordnung von a, b in σ nicht ändern.

Erwünschte Eigenschaft: Neutralität und Anonymität

- **Neutralität**

Kein Kandidat soll bevorzugt werden.

- Falls zwei Kandidaten die Positionen in π_1, \dots, π_n tauschen, müssen sie auch in σ die Positionen tauschen.

- **Anonymität**

Kein Wähler soll vor anderen bevorzugt werden.

- Falls zwei Wähler ihre Sortierungen tauschen, soll sich σ nicht ändern.

Erwünschte Eigenschaft: Monotonie und Konsistenz

- **Monotonie**

Fall der Rang eines Kandidaten durch einen Wähler verbessert wird, darf sich dessen Rang in σ auch nur verbessern.

- **Konsistenz**

Falls die Wähler in zwei disjunkte Teilmengen S und T geteilt werden und in beiden Zusammenfassungen von S und T a vor b ist, dann muß auch in der Gesamtzusammenfassung a vor b sein.

Diktatur und Demokratie

- **Diktatur**: $f(\pi_1, \dots, \pi_n) = \pi_i$
- **Demokratie** (auch Mehrheitszusammenfassung):
Nutze erweitertes Condorcet Kriterium um
Kandidaten zu sortieren.
 - Funktioniert immer für $m = 2$.
 - Nicht immer realisierbar für $m \geq 3$.
 - **Satz [May, 1952]**: Für $m = 2$, Demokratie ist die einzige
Zusammenfassungsfunktion, die monoton, neutral und
anonym ist.

Arrow's Nichterfüllbarkeitssatz

[Arrow, 1951]

- **Satz:** Fall $m \geq 3$, dann ist Diktatur die einzige Zusammenfassungsfunktion, die eimütig und unabhängig von irrelevanten Alternativen ist.
 - Gewann Nobelpreis (1972)

Zusammenfassung mittels Position

- **Mehrheit**
 - $\text{score}(a)$ = Anzahl der Wähler, die a als #1wählten
 - σ : sortiere die Kandidaten nach absteigendem Score
- **Top-k Bestätigung**
 - $\text{score}(a)$ = Anzahl der Wähler, die a unter die Top-k ersten wählten
 - σ : sortiere die Kandidaten nach absteigendem Score
- **Borda's rule [Borda, 1781]**
 - $\text{score}(a) = \sum_i \pi_i(a)$
 - σ : sortiere die Kandidaten nach ansteigendem Score
- Alle verletzen die Unabhängigkeit von irrelevanten Alternativen

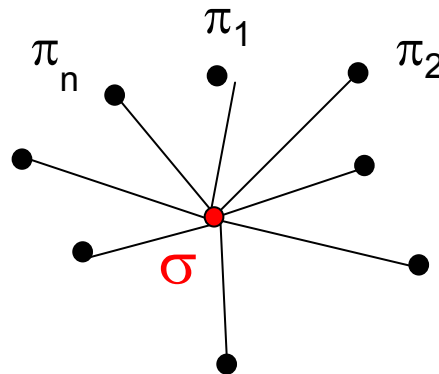
Positionsmethoden: Beispiel

a	a	c	b
b	d	b	d
c	c	d	c
d	b	a	a

	Mehrheit	Top-2 Bestätigung	Borda
a	2	2	$1+1+4+4=10$
b	1	3	$2+4+2+1=9$
c	1	1	$3+3+1+3=10$
d	0	2	$4+2+3+2=11$

Optimales Zusammenfassen

- d : Distanzmaß zwischen Sortierungen
- **Definition:** Die **optimale Zusammenfassung** für π_1, \dots, π_n bezüglich d ist die Sortierung σ , die $\sum_i d(\sigma, \pi_i)$ minimiert.



Distanzmaße

- Kendall Tau Distanz (auch “bubble sort Distanz”)
 - $K(\sigma, \tau)$ = Anzahl der Paare von Kandidaten (a, b) bei den σ und τ nicht übereinstimmen
 - z.B.: $K((a\ b\ c\ d), (a\ d\ c\ b)) = 0 + 2 + 1 = 3$
- Spearman's Distanz
 - $F(\sigma, \tau) = \sum_a |\sigma(a) - \tau(a)|$
 - z.B.: $F((a\ b\ c\ d), (a\ d\ c\ b)) = 0 + 2 + 0 + 2 = 4$

Optimale Zusammenfassung nach Kemeny [Kemeny 1959]

- Optimale Zusammenfassung bezüglich Kendall-Tau Distanz
- Satz [Young & Levenglick, 1978] [Truchon 1998]: Optimale Zusammenfassung nach Kemeny die einzige Zusammenfassungsfunktion, die neutral und konsistent ist und auch das erweiterte Condorcet Kriterium erfüllt.
 - Effektiv um Spam bekämpfen
- Erzeugendes Modell:
 - σ^* ist die “richtige” Sortierung
 - π_1, \dots, π_n sind aus σ durch nicht-deterministischen Austausch aller Paare erzeugt, ein Austausch wird mit Wahrscheinlichkeit $< \frac{1}{2}$ durchgeführt.
 - Dann: optimales Zusammenfassen nach Kemeny hat bei σ die maximale Likelihood bei gegebenen π_1, \dots, π_n . [Young 1988]

Komplexität von optimalem Zusammenfassen nach Kemeny

- NP-hard, auch für $n = 4$ [Dwork et al. 2001]
 - In P, für $n = 2$.
 - Unbekannt für $n = 3$.
- Kann durch Spearman Distanz approximiert werden:
 - Satz [Diaconis-Graham]:
$$K(\sigma, \tau) \leq F(\sigma, \tau) \leq 2 K(\sigma, \tau)$$
- Was ist die Komplexität des optimalen Zusammenfassens nach Spearman?

Optimales Zusammenfassen nach Spearman

- Satz [Dwork et al. 2001]

Optimales Zusammenfassen nach Spearman kann in polynomieller Zeit berechnet werden.

- Beweis

- Finde ein σ , das $\sum_i \sum_a |\sigma(a) - \pi_i(a)|$ minimiert
- Definiere einen gewichteten bipartiten Graphen $G = (L, R, W)$:
 - $L = M$ (Kandidaten)
 - $R = \{1, \dots, m\}$: die möglichen Positionen
 - $W(a, r) = \sum_i |r - \pi_i(a)|$
- Eine Zuordnung in G entspricht einer Sortierung
- Kosten einer Zuordnung: $\sum_i \sum_a |\sigma(a) - \pi_i(a)|$
- Deshalb, finde eine Zuordnung mit minimalen Kosten in dem bipartiten Graphen

Lokale Kemenisierung

[Dwork et al. 2001]

- **Definition:** Eine Sortierung σ ist eine **lokale optimale Zusammenfassung nach Kemeny** für π_1, \dots, π_n , wenn es keine andere Sortierung σ' gibt, mit:
 - σ' kann aus σ durch einen Tausch eines Paares erzeugt werden
 - Erfüllt $\sum_i K(\sigma', \pi_i) < \sum_i K(\sigma, \pi_i)$
- **Eigenschaften:**
 - Jede optimale Zusammenfassung nach Kemeny ist auch lokal optimal nach Kemeny, aber die Umkehrung gilt nicht allgemein.
 - Lokale optimale Zusammenfassung nach Kemeny erfüllt ECK.
 - Lokale optimale Zusammenfassung nach Kemeny kann in $O(n m \log m)$ Zeit berechnet werden.

Markov Ketten Techniken

[Dwork et al. 2001]

- Markov Ketten Zustände = Kandidaten
- Übergänge hängen von den Sortierungen der Wähler ab.
- Grundidee: wechsle probabilistisch zu einem “besseren” Kandidaten
- Ergebnissortierung: erzeugt durch die stationäre Verteilung

Vier MC Methoden

- Aktueller Zustand ist Kandidat **a**.
- **MC1**: Wähle gleichverteilt aus dem Multiset aller Kandidaten, die bei einem Wähler mindestens genauso weit vorn wie **a** liegen.
 - Wahrscheinlichkeit bei **a** zu bleiben: \sim durchschnittliche Position von **a**.
- **MC2**: Wähle einen Wähler **i** zufällig gleichverteilt und picke zufällig gleichverteilt einen Kandidaten unter aus denjenigen, welche beim **i**-ten Wähler mind. soweit vorn wie **a** liegen.
- **MC3**: Wähle einen Wähler **i** zufällig gleichverteilt und picke zufällig gleichverteilt einen Kandidaten **b**. Falls der **i**-te Wähler **b** vor **a** hat, gehe zu **b**. Sonst bleibe bei **a**.
- **MC4**: Wähle einen Kandidaten **b** zufällig gleichverteilt. Falls die Mehrheit der Wähler **b** vor **a** haben, gehe zu **b**. Sonst bleibe bei **a**.
 - Position von **a** \sim Anzahl der “Zweikämpfe” die **a** gewinnt.