Ranking by Reordering

TECHNISCHE UNIVERSITÄT DARMSTADT

Tobias Joppen

Überblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Einleitung
- Rank-differential Methode
  - Idee
  - Problemdefinition
  - Beispiel
  - Vereinfachung des Problems
  - Zusammenfassung und Eigenschaften
- Rating-differential Methode
  - Unterschiede der Methoden

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was hatten die bisherigen Ranking Methoden gemeinsam?

1) Daten Sammeln

Data	**Team 1**	**Team 2**	**…**	**Team n**
Team 1		17-8 4-2	…	6-21
Team 2	3-11 1-3		…	12-10 13-19 9-12
…	…	…		…
Team n	11-10 11-9	8-4 12-8	…	

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was hatten die bisherigen Ranking Methoden gemeinsam?

1) Daten Sammeln
2) Methode anwenden

   z.B. Massey: $Mr = p$

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was hatten die bisherigen Ranking Methoden gemeinsam?

1) Daten Sammeln
2) Methode anwenden
3) Ratingvektor berechnen

-1,4
4,3
0,5
1,2
-3,1
5,3
7,9

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was hatten die bisherigen Ranking Methoden gemeinsam?

1) Daten Sammeln
2) Methode anwenden
3) Ratingvektor berechnen
4) Rankingvektor bestimmen

6
3
5
4
7
2
1

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was hatten die bisherigen Ranking Methoden gemeinsam?

1) Daten Sammeln
2) Methode anwenden
3) Ratingvektor berechnen
4) Rankingvektor bestimmen

Ist das alles notwendig um ein Ranking bestimmen zu können?

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Was hatten die bisherigen Ranking Methoden gemeinsam?

1) Daten Sammeln
2) Methode anwenden
3) Ratingvektor berechnen
4) Rankingvektor bestimmen

Ist das alles notwendig um ein Ranking bestimmen zu können?

-Nein

Rank-differential Methode

TECHNISCHE UNIVERSITÄT DARMSTADT

Wenn

- Aufgabe: Jedes Team hat eine Platzierung

Dann

- Berechnen des Ratingvektors kann vermieden werden!

Weil

- Aus Spieldaten können direkt Platzierungen erzeugt werden

Idee

TECHNISCHE UNIVERSITÄT DARMSTADT

Ziel:

Rankingvektor r bestimmen z.B.:

$$\begin{matrix}\text{Duke}\\ \text{Miami}\\ \text{UNC}\\ \text{UVA}\\ \text{VT}\end{matrix}\begin{pmatrix}5\\1\\4\\3\\2\end{pmatrix}$$

Idee

Ziel:

Rankingvektor r bestimmen z.B.:

$$\begin{matrix} \text{Duke} \\ \text{Miami} \\ \text{UNC} \\ \text{UVA} \\ \text{VT} \end{matrix} \begin{pmatrix} 5 \\ 1 \\ 4 \\ 3 \\ 2 \end{pmatrix}$$

Lässt sich auch umformen in eine Matrix

(Paarweiser Unterschied im positiven)

$$R = \begin{matrix} \\ \text{Duke} \\ \text{Miami} \\ \text{UNC} \\ \text{UVA} \\ \text{VT} \end{matrix} \begin{matrix} \begin{matrix} \text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 3 & 2 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 2 & 1 & 0 \end{pmatrix} \end{matrix}$$

Idee

Jeder Rankingvektor der Länge n erzeugt eine n x n Rangunterschiedsmatrix R, welche eine symmetrische Umsortierung der folgenden fundamentalen Rangunterschiedsmatrix $\hat{R}$ ist.

$$\hat{R} = \begin{array}{c} \\ 1 \\ 2 \\ \dots \\ n-1 \\ n \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \dots & \mathrm{n} \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & \dots & n-1 \\ & 0 & 1 & \dots & n-2 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & 1 \\ & & & & 0 \end{pmatrix} \end{array}$$

$$\hat{r} = \begin{array}{c} \\ 1^{\mathrm{st}} \\ 2^{\mathrm{nd}} \\ 3^{rd} \\ \vdots \\ n^{th} \end{array} \begin{array}{c} \\ \begin{pmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ \mathrm{n} \end{pmatrix} \end{array}$$

Idee

TECHNISCHE UNIVERSITÄT DARMSTADT

Kurz gesagt:

- Ranking-Vektor r erzeugt Rangunterschiedsmatrix R

- Es gilt

$$R = Q^T * \hat{R} * Q$$

mit Q als Permutationsmatrix.

Idee

TECHNISCHE UNIVERSITÄT DARMSTADT

Einmal weiterdenken…

- R ist eine Team-gegen-Team Matrix
- Schon einige gesehen im Seminar

Idee

TECHNISCHE UNIVERSITÄT DARMSTADT

Einmal weiterdenken…

- R ist eine Team-gegen-Team Matrix
- Schon einige gesehen im Seminar
- R hat große (inhaltliche) Parallelität mit der Markov-voting-matrix $V_{pointdiff}$ aus der vorletzten Vorlesung

> Zur Erinnerung: Hier wurden aufaddierte Punktunterschiede gespeichert. Ein Eintrag > 0 bedeutet, dass dieses Team dem Gegnerteam um diesen Wert unterlegen ist.

Idee

TECHNISCHE UNIVERSITÄT DARMSTADT

Einmal weiterdenken…

- R ist eine Team-gegen-Team Matrix
- Schon einige gesehen im Seminar
- R hat große (inhaltliche) Parallelität mit der Markov-voting-matrix $V_{pointdiff}$ aus der vorletzten Vorlesung

> Zur Erinnerung: Hier wurden aufaddierte Punktunterschiede gespeichert. Ein Eintrag > 0 bedeutet, dass dieses Team dem Gegnerteam um diesen Wert unterlegen ist.

- Diese Werte sind „anders herum“, daher: Transponieren!
- Ab sofort Datenunterschiedsmatrix: $D = \left(V_{pointdiff}\right)^T$

Idee

TECHNISCHE UNIVERSITÄT DARMSTADT

Wie kommen wir zur Lösung?

- Gegeben: D und $\hat{R} \leftarrow D$ Erzeugt durch Spielergebnisse
- Gesucht: Eine Umsortierung von D zu $\hat{R}$

Idee

Wie kommen wir zur Lösung?

- Gegeben: D und $\hat{R}$ ← D Erzeugt durch Spielergebnisse
- Gesucht: Eine Umsortierung von D zu $\hat{R}$

Das ist natürlich nur selten möglich. Daher:

1) D und $\hat{R}$ normalisieren
2) keine 100% Umsortierung suchen, sondern "nearest matrix problem" lösen. Also eine Umsortierung mit dem kleinsten Fehler $\|(Q^T * D * Q) - R\|$ finden.

Problemdefinition

TECHNISCHE UNIVERSITÄT DARMSTADT

Mathematische Formulierung des Problems:

Finde $\min_{Q} \left\| Q^T D Q - \hat{R} \right\|$

sodass: $Qe = e$

$e^T * Q = e^T$

$q_{ij} \in \{0,1\}$

- Wegen der Norm ist dieses Problem nichtlinear.
- Wir verwenden als Norm die Frobeniusnorm (2er-Norm euklidische-Norm für Matrizen).
- Dadurch wird das Problem quadratisch nichtlinear.

Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

Daten sammeln

$$D = V^T = \begin{array}{c} \\ \text{Duke} \\ \text{Miami} \\ \text{UNC} \\ \text{UVA} \\ \text{VT} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 45 & 0 & 18 & 8 & 20 \\ 3 & 0 & 0 & 2 & 0 \\ 31 & 0 & 0 & 0 & 0 \\ 45 & 0 & 27 & 38 & 0 \end{pmatrix} \end{array}$$

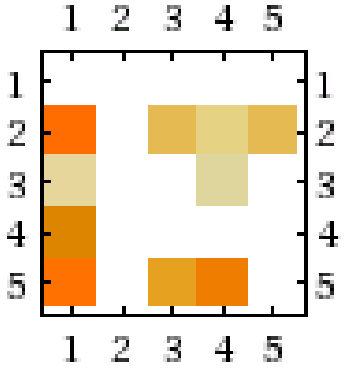
$$\hat{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Beispiel

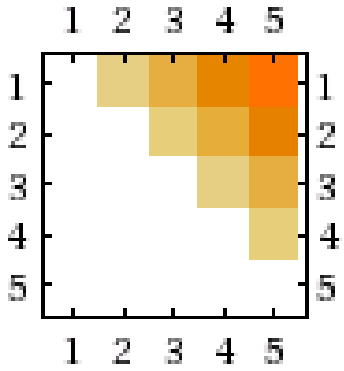
TECHNISCHE UNIVERSITÄT DARMSTADT

Daten normalisieren

$$D = \begin{array}{c} \\ \text{Duke} \\ \text{Miami} \\ \text{UNC} \\ \text{UVA} \\ \text{VT} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ .19 & 0 & .08 & .03 & .08 \\ .01 & 0 & 0 & .01 & 0 \\ .13 & 0 & 0 & 0 & 0 \\ .19 & 0 & .11 & .16 & 0 \end{pmatrix} \end{array}$$



$$\hat{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & .05 & .10 & .15 & .20 \\ 0 & 0 & .05 & .10 & .15 \\ 0 & 0 & 0 & .05 & .10 \\ 0 & 0 & 0 & 0 & .05 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$



Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

Nearest Matrix Problem lösen

$n = 5$

$5! = 120$ ← Anzahl der verschiedenen Permutationen für D

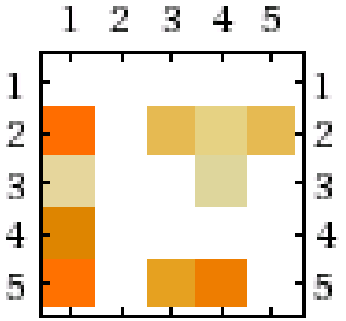
Brute Force ergibt Permutation (5 2 4 3 1),

beziehungsweise:

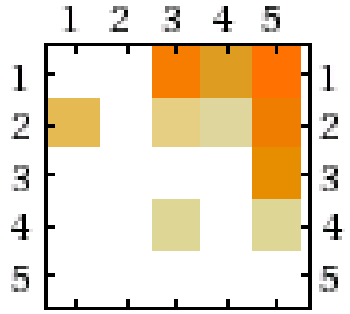
$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad Q^T DQ = \begin{pmatrix} 0 & 0 & .16 & .11 & .19 \\ .08 & 0 & .03 & .01 & .16 \\ 0 & 0 & 0 & 0 & .13 \\ 0 & 0 & .01 & 0 & .01 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Beispiel

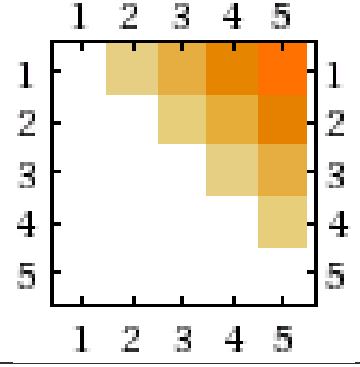
$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ .19 & 0 & .08 & .03 & .08 \\ .01 & 0 & 0 & .01 & 0 \\ .13 & 0 & 0 & 0 & 0 \\ .19 & 0 & .11 & .16 & 0 \end{pmatrix}$$



$$Q^T DQ = \begin{pmatrix} 0 & 0 & .16 & .11 & .19 \\ .08 & 0 & .03 & .01 & .16 \\ 0 & 0 & 0 & 0 & .13 \\ 0 & 0 & .01 & 0 & .01 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



$$\hat{R} = \begin{pmatrix} 0 & .05 & .10 & .15 & .20 \\ 0 & 0 & .05 & .10 & .15 \\ 0 & 0 & 0 & .05 & .10 \\ 0 & 0 & 0 & 0 & .05 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Vereinfachung des Problems

Es gilt dank Frobeniusnorm und der Struktur von $\hat{R}$:

$$\left\|Q^T DQ - \hat{R}\right\|_F^2 = trace\left(D^T D\right) - 2 * trace\left(Q^T DQ\hat{R}\right) + trace\left(\hat{R}^T \hat{R}\right)$$

Vereinfachung des Problems

TECHNISCHE UNIVERSITÄT DARMSTADT

Es gilt dank Frobeniusnorm und der Struktur von $\hat{R}$:

$$\left\|Q^T DQ - \hat{R}\right\|_F^2 = trace\,(D^T D) - 2 * trace\left(Q^T DQ\hat{R}\right) + trace\left(\hat{R}^T \hat{R}\right)$$

$trace\,(D^T D)$ und $trace\left(\hat{R}^T \hat{R}\right)$ sind konstant!

Das Problem kann also umformuliert werden:

Finde $\max_Q trace(Q^T D\, Q\, \hat{R})$

sodass: $Qe = e$

$e^T * Q = e^T$

$q_{ij} \in \{0,1\}$

Evolutionärer Ansatz

TECHNISCHE UNIVERSITÄT DARMSTADT

Das Problem ist NP-schwer.

Evolutionäre Algorithmen lösen das Problem aber gut.

Evolutionärer Ansatz

TECHNISCHE UNIVERSITÄT DARMSTADT

Es wird keine Permutationsmatrix Q gesucht, sondern ein Permutationsvektor q (äquivalent):

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad q = (5 \quad 2 \quad 4 \quad 3 \quad 1)$$

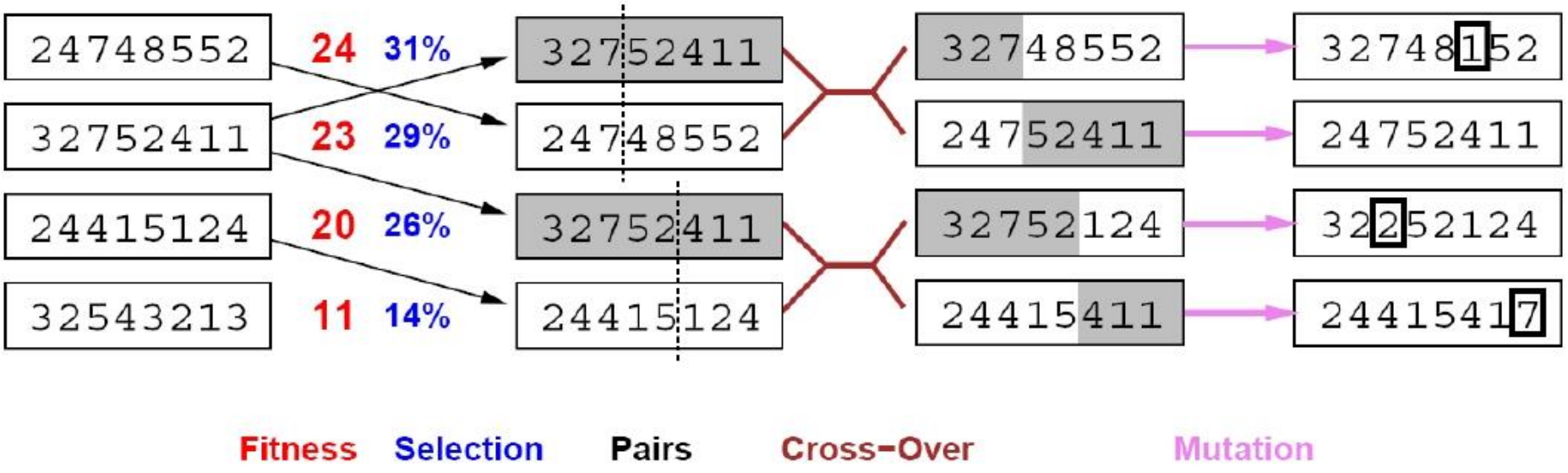
Es gilt die Gleichheit:

$$\min_{Q} \left\| Q^T D Q - \hat{R} \right\| = \min_{q \in p_n} \left\| D(q, q) - \hat{R} \right\|$$

Evolutionärer Ansatz

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mitglieder der Population = Permutationsvektoren
- Fitnessfunktion = Fehler der umsortierten Matrix (oder hillside violations)
- Die fitten Mitglieder werden gekreuzt, der Rest wird mutiert
- Algorithmus stoppt, wenn die Bevölkerung sich kaum mehr ändert



Zusammenfassung Rank-Differential

TECHNISCHE UNIVERSITÄT DARMSTADT

D item-by-item Datenmatrix (paarweise Beziehungen)

$\hat{R}$ fundamentale Rangunterschiedsmatrix

Q ist die Permutationsmatrix

q ist der zu Q korrespondierende Permutationsvektor

$D(q,q)$ ist die umsortierte Datenunterschiedsmatrix $Q^T * D * Q$

Der Algorithmus

1. Löse das Optimierungsproblem

$$\min_Q \left\| Q^T D Q - \hat{R} \right\|$$

2. Sortiere q in absteigender Reihenfolge und speichere die sortierten Indizes als Rankingvektor

Eigenschaften Rank-Differential

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kein Ratingvektor
- Findet die beste Umsortierung von D zu $\hat{R}$
- Das Optimierungsproblem ist NP-Schwer
- Nur für kleines n geeignet
- Beliebige Datenmatrizen können verwendet werden
- Wie bei anderen Methoden auch:

  D kann aus mehreren Matrizen zusammengesetzt werden

Rating Differential Method

TECHNISCHE UNIVERSITÄT DARMSTADT

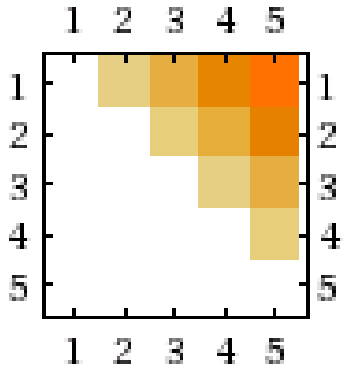
Von:

- Kathryn Pedings
- M.S. Thesis
- College of Charleston

- Für College Basketball

Idee

Betrachtet man $\hat{R}$, so erkennt man eine Struktur:



„hillside Form“

Eine Matrix R ist in hillside Form, wenn

$\forall i \geq j\colon r_{ij} = 0$ (strikte obere Dreiecksmatrix)

$\forall i \ni j \leq k\colon r_{ij} \leq r_{ik}$ (Zeilen in aufsteigender Reihenfolge)

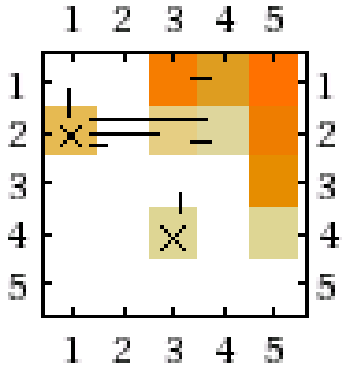
$\forall j \ni i \leq k\colon r_{ij} \geq r_{kj}$ (Spalten in absteigender Reihenfolge)

Idee

Wähle Q so, dass die wenigsten Hillside-Verstöße auftreten

Beispiel mit 9 Verstößen ($Q^T DQ$):

$$\begin{pmatrix} 0 & 0 & .16 & .11 & .19 \\ .08 & 0 & .03 & .01 & .16 \\ 0 & 0 & 0 & 0 & .13 \\ 0 & 0 & .01 & 0 & .01 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



$\forall i \geq j\colon r_{ij} = 0$ (strikte obere Dreiecksmatrix)

$\forall i \ni j \leq k\colon r_{ij} \leq r_{ik}$ (Zeilen in aufsteigender Reihenfolge)

$\forall j \ni i \leq k\colon r_{ij} \geq r_{kj}$ (Spalten in absteigender Reihenfolge)

Unterschiede der Methoden

TECHNISCHE UNIVERSITÄT DARMSTADT

Die Unterschiede wirken sich aus auf:

- Keine Normalisierung notwendig, da
- Kein Vergleich zu $\hat{R}$, sondern Benotung der Form
- Wahl von Q anders bestimmt (Fehlerterm vs. Anzahl der Verstöße)
- Andere Fitnessfunktion des evolutionären Algorithmus

Nachwort zum Lösen des Problems

TECHNISCHE UNIVERSITÄT DARMSTADT

- Evolutionäre Algorithmen lösen das Problem
- Laufzeit aber deutlich höher als z.B. Massey
- In dieser Variante fast ausschließlich für kleines n effizient
- Kann aber besser gelöst werden
  - Transformieren in BILP (binary integer linear program)
  - Vereinfachen in LP (linear program)

Diese Transformation wird im späteren Verlauf des Buches eingeführt (Kapitel 15, rank-aggregation).

Rating-Differential ist ein Spezialfall der Rank-Aggregation.

Verwendungsmöglichkeiten

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ranking von Spielern / Mannschaften

- Online Metasuche
  - Spamfilter
  - Suchwebsite-Rankings

  http://www10.org/cdrom/papers/577/

Ende

TECHNISCHE UNIVERSITÄT DARMSTADT

Vielen Dank für die Aufmerksamkeit!

Noch Fragen?