

# Effiziente Klassifikation und Ranking mit paarweisen Vergleichen

Abschlussbericht

Sang-Hyeun Park

Betreuer: Prof. Fürnkranz  
Technische Universität Darmstadt  
Fachbereich Informatik  
Knowledge Engineering

7. Dezember 2006



# Gliederung

## Kurze Einleitung

## Effiziente Klassifizierung

Dekodierung

Quick Weighted Voting

QWeighted als Suchalgorithmus

## Effizientes Ranking

Ranking

„Schweizer-System“

Erweitertes Schweizer-System

Erweiterte Auswertungen



# Gliederung

## Kurze Einleitung

### Effiziente Klassifizierung

Dekodierung

Quick Weighted Voting

QWeighted als Suchalgorithmus

### Effizientes Ranking

Ranking

„Schweizer-System“

Erweitertes Schweizer-System

Erweiterte Auswertungen



## Bemerkung

- ▶ seit Zwischenbericht (13.07.2006) nicht viele neue Inhalte
- ▶ viele Folien wurden übernommen und sehr komprimiert dargestellt
- ▶ auf die zwei neuen Punkte: QWeighted als Suchalgorithmus, Erweitertertes Schweizer-System wird dafür ausführlicher eingegangen

# Round Robin Binärisierung

## Nachteile

Bei  $n$  Klassen

- ▶ Trainingsphase:  $\frac{n(n-1)}{2}$  Klassifikatoren zu lernen
- ▶ Klassifikationsphase : auszuwerten

Der Gesamtaufwand in der **Trainingsphase** ist trotzdem kleiner als  $n$  Lernprobleme in der *1-vs-rest* Binärisierung.

## Diplomarbeitsziel

Effiziente Klassifikationsphase (Dekodierung) mit RR für Multiklassifikation und Ranking

# Gliederung

Kurze Einleitung

Effiziente Klassifizierung

Dekodierung

Quick Weighted Voting

QWeighted als Suchalgorithmus

Effizientes Ranking

Ranking

„Schweizer-System“

Erweitertes Schweizer-System

Erweiterte Auswertungen



# Dekodierung

## Votingbasiert

### Voting

- ▶ Jeder Klassifikator  $K_{i,j}$  stimmt für eine Klasse.
- ▶ Die Klasse mit den meisten Stimmen "gewinnt".

### Weighted Voting

- ▶ Gewichtung der Stimmen
- ▶ Gewichtung z.B. durch Fehlerabschätzung der Vorhersage

Bei beiden Voting-Arten werden alle Klassifikatoren ( $\frac{n(n-1)}{2}$ ) ausgewertet.



## quick Weighted Voting

Beobachtung: Es gibt Vergleiche, die unnötig für die **Klassifikation** sind.

- ▶ Eigenschaft *uneinholbar*
  - ▶ Klasse  $j$  mit bisher  $n - 2$  Gewinnen kann gegen Klasse  $i$  mit bisher 3 Verlusten nicht verlieren  $\rightarrow$  Klasse  $i$  kommt als *top rank* nicht mehr in Frage.
  - ▶ um oft diese Situation herzustellen  $\rightarrow$  *Reihenfolge* der Vergleiche wichtig



# quick Weighted Voting

## Algorithmus

$K$  sei die Menge der Klassen  $K := \{k_1, k_2, k_3, \dots, k_n\}$  und  
 $limit(x) := played(x) - wins(x)$ .

1. Wähle Klasse  $k_i$  mit minimalen  $limit(k_i)$  und  $k_i \in K$   
(Spieler<sub>1</sub>)
2. Wähle Klasse  $k_j$  mit minimalen  $limit(k_j)$ ,  $k_j \in K \setminus k_i$  und die  
Paarung  $(k_i, k_j)$  wurde noch nicht gespielt (Spieler<sub>2</sub>)
3.
  - 3.1 Existiert keine Klasse, die die Bedingungen in Schritt 2  
erfüllen  $\rightarrow$  winner :=  $k_j$  STOP
  - 3.2 Ansonsten
    - 3.2.1 play( $k_i, k_j$ )
    - 3.2.2 Aktualisiere Statistiken (limits, played, ...)
    - 3.2.3 Gehe zu Schritt 1

# quick Weighted Voting

## Vereinfacht

- ▶ Spiele solange ungespielte Paarungen mit der aktuellen top rank Klasse (bester limit bisher)
  - ▶ bis zur Verifikation vom top rank (limit korrekt)
  - ▶ Eventuell wird nach einem Spiel eine neue top rank Klasse bestimmt.

# quick Weighted Voting

## Ergebnisse

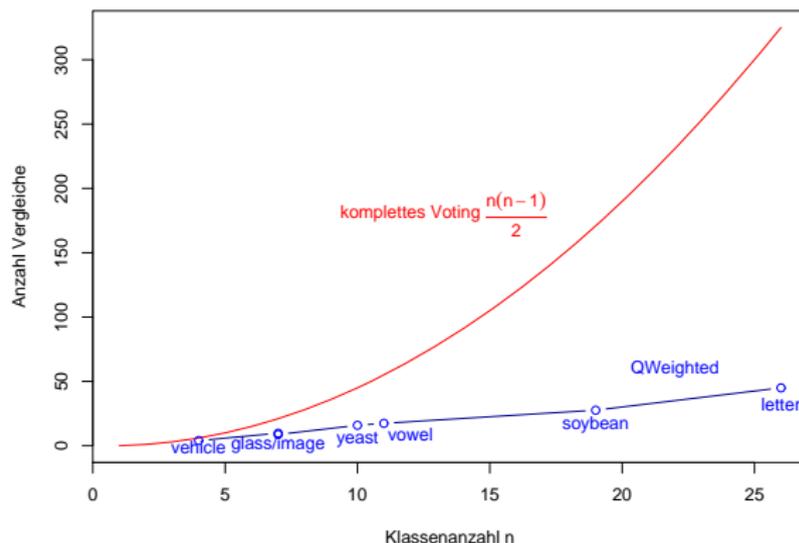
Datensatz	$n$	$\frac{n(n-1)}{2}$	QWeighted
vehicle	4	6	3,9844
glass	6	15	9,7476
image	8	28	8,7459
yeast	10	45	15,8749
vowel	11	55	17,4212
soybean	19	171	27,6510
letter	26	325	45,0053

**Tabelle:**  $n$  ist die Anzahl der Klassen,  $\frac{n(n-1)}{2}$  die Anzahl der Vergleiche von Weighted Voting und QWeighted die durchsch. Anzahl der Vergleiche von Quick Weighted Voting (Ripper, 10CV)



# quick Weighted Voting

Ergebnisse QWeighted im Vergleich zum kompletten Voting



# quick Weighted Voting

## Ergebnisse

Datensatz	JRip	NaiveBayes	C4.5(J48)	SVM	
vehicle	3,9844	4,2660	3,9581	3,6442	6
glass	9,7476	9,5764	9,6896	9,9249	15
image	8,7459	9,0299	8,5472	8,2905	28
yeast	15,8749	15,8608	15,4774	15,5169	45
vowel	17,4212	17,0939	17,1253	15,2808	55
soybean	27,6510	27,6972	29,4460	28,3645	171
letter	45,0053	44,3960	47,7705	42,2618	325

**Tabelle:** Quick Weighted Voting unter verschiedenen Basis-Lerner



# QWeighted als Suchalgorithmus

## Greedy-Algorithmus

- ▶ QWeighted verfährt nach Greedy-Prinzip (verfolge das lokale Optimum)
- ▶ Starke Ähnlichkeit zum Kruskal Algorithmus (bestimmt minimalen Spannbaum eines gewichteten, azyklischen und ungerichteten Graphen)



# Kruskal Algorithmus

---

## Algorithmus 1 : Algorithmus von Kruskal

---

**Input** : Ein (ungerichteter, zusammenhängender, gewichteter) Graph  
 $G_T = (G, E)$  mit den Gewichten  $w$ .

**Output** : ein minimaler Spannbaum von  $G_T$

1 **begin**

2 | Sortiere die Kanten in  $E$  nach aufsteigendem Gewicht: Sei  
|  $E = \{e_1, \dots, e_m\}$  mit  $w(e_1) \leq \dots \leq w(e_m)$ .

3 | Setze  $T := \emptyset$ ,  $k := 1$ .

4 | Falls die Kanten in  $T \cup \{e_k\}$  einen Baum bilden, setze  $T := T \cup \{e_k\}$ .

5 | Falls  $k = m$ , dann STOP. Sonst setze  $k := k + 1$  und weiter mit 4.

6 **end**

---

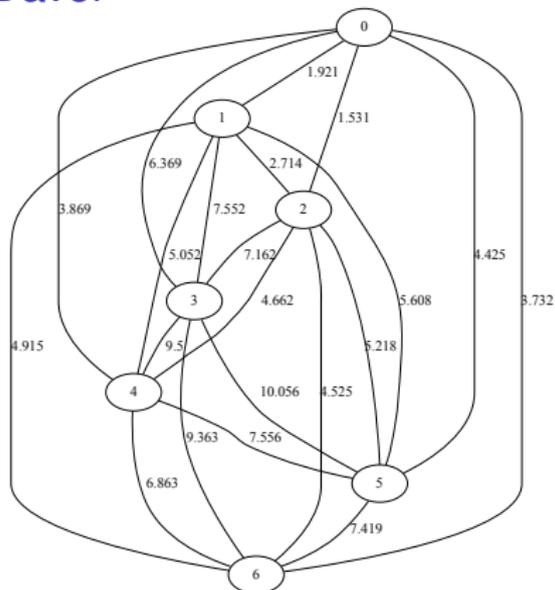
*Beginne mit einem leeren Graphen  $G_E$ . Füge iterativ die minimal gewichteten Kanten von  $G_T$  dem Graphen  $G_E$  hinzu, solange  $G_E$  azyklisch bleibt.*



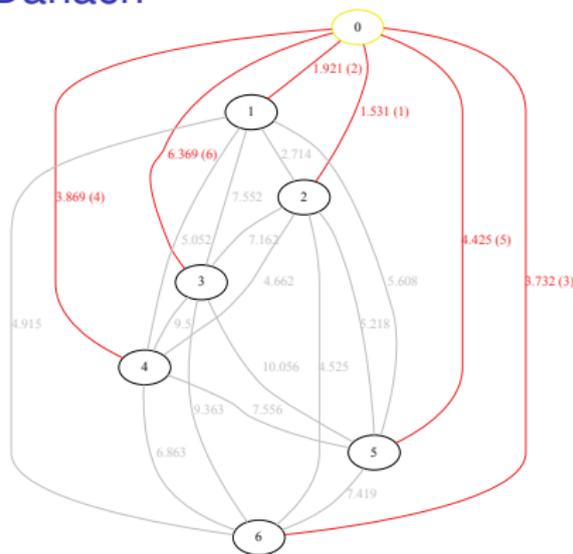
# Kruskal Algorithmus

Ein Beispiel

Davor



Danach



## Projektion des Dekodierungsproblem auf einen Graphen

### Transformationsregeln:

- ▶ Die Knotenmenge  $E$  sei die Klassenmenge  $K$
- ▶ Verbinde jedes Paar von Knoten  $v, w \in E$  durch eine ungerichtete Kante  $(v, w)$
- ▶ gewichte jede Kante  $e_{v,w} = (v, w) \in E$  mit dem Wert  $w(e_{v,w}) = L(k_v) + L(k_w)$ .

### Bedeutung:

- ▶ Knoten stellen die Klassen dar.
- ▶ Kanten stellen die paarweisen Klassifikatoren der inzidenten Knoten bzw. Klassen dar.
- ▶ Gewicht der Kante (Summe von Limits der inzidenten Knoten bzw. Klassen) stellen einen Heuristik-Wert dar.

**Der minimale Spannbaum dieses Baumes beschreibt die minimale Menge an Klassifikatoren zur Bestimmung der *top rank* Klasse**



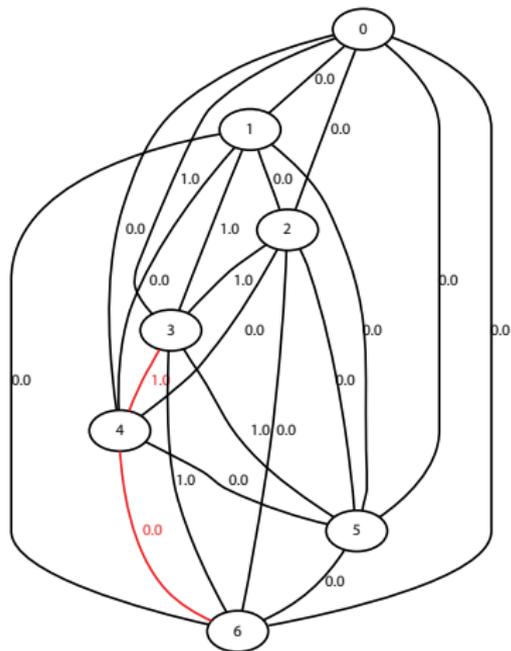
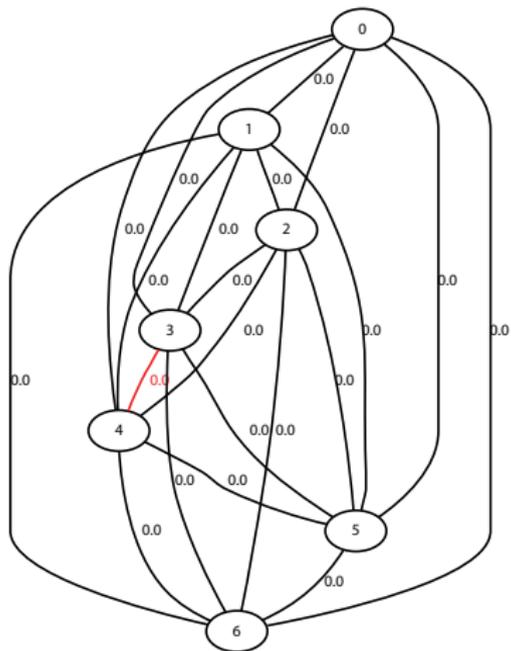
## Projektion des Dekodierungsproblem auf einen Graphen

- ▶ **Kruskal Algorithmus jedoch nicht anwendbar, da Kantengewichte (Limits) nicht vorgegeben**
- ▶ **QWeighted Algorithmus auf den Graphen übertragen**

*Beginne mit einem leeren Graphen  $G_Q$ . Füge die minimale Kante aus  $G_T$  in  $G_Q$  hinzu. (Die Kantengewichte von  $G_T$  werden daraufhin aktualisiert). Wiederhole diesen Schritt, bis ein Knoten existiert, der  $n - 1$  inzidente Kanten besitzt. Dieser Knoten stellt dann die top rank Klasse dar.*

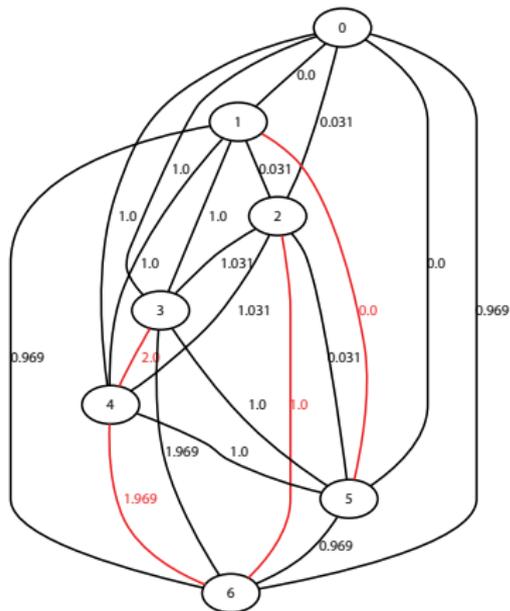
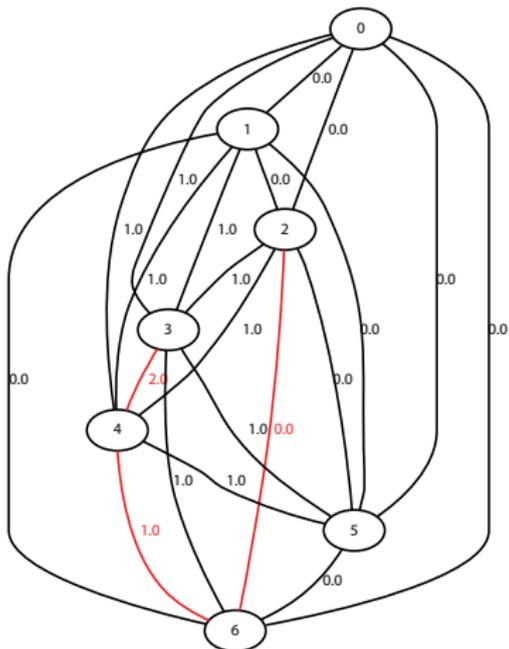
# Projektion des Dekodierungsproblem auf einen Graphen

Ein konkretes Beispiel, Schritte 1-2



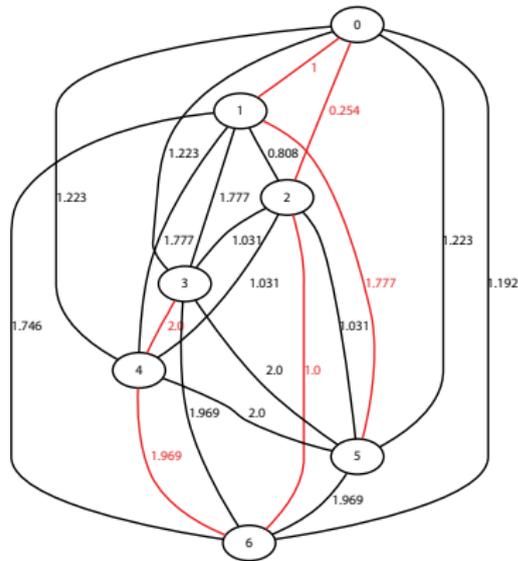
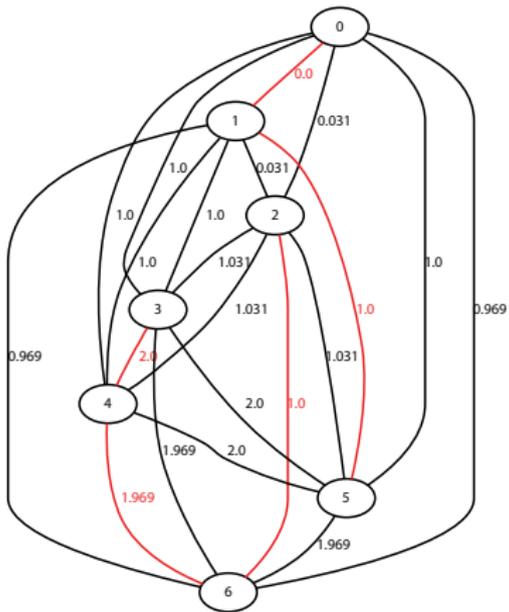
# Projektion des Dekodierungsproblem auf einen Graphen

Ein konkretes Beispiel, Schritte 3-4



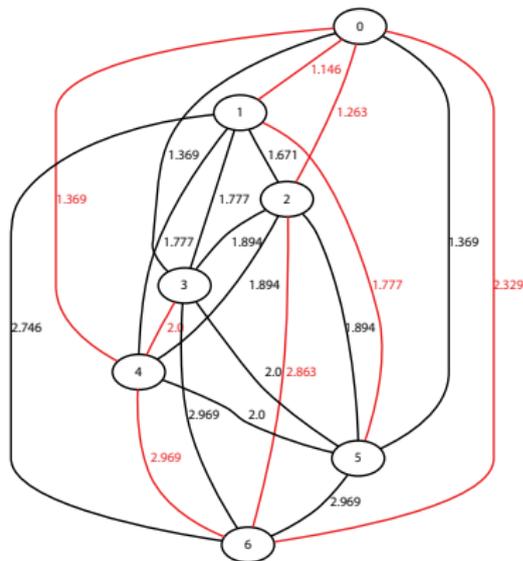
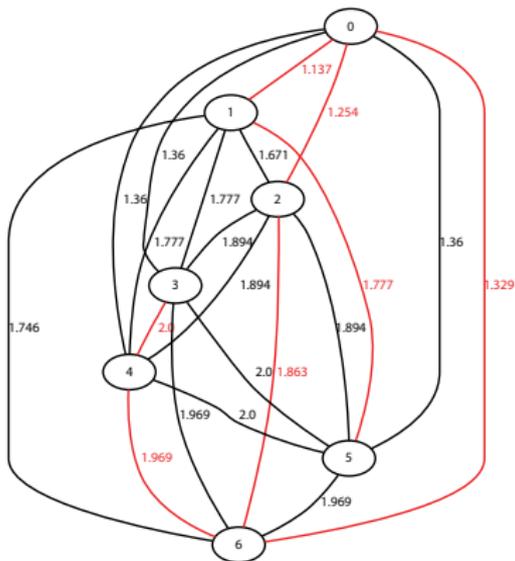
# Projektion des Dekodierungsproblem auf einen Graphen

Ein konkretes Beispiel, Schritte 5-6



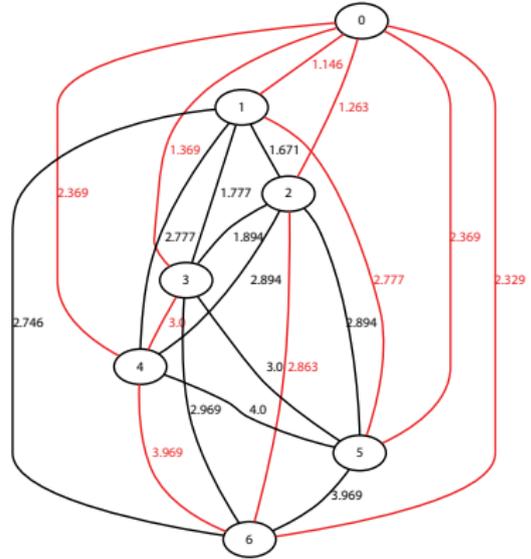
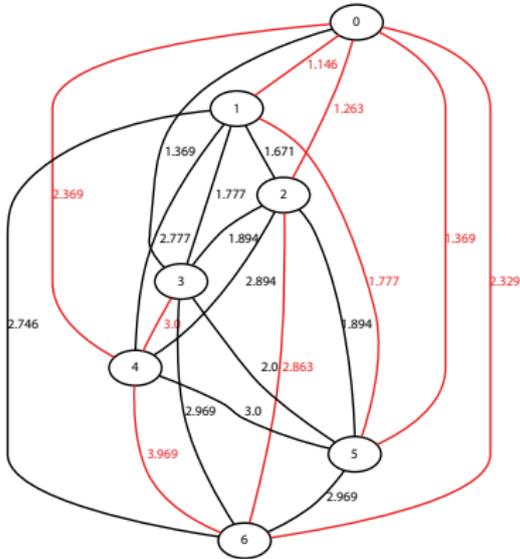
# Projektion des Dekodierungsproblem auf einen Graphen

Ein konkretes Beispiel, Schritte 7-8



# Projektion des Dekodierungsproblem auf einen Graphen

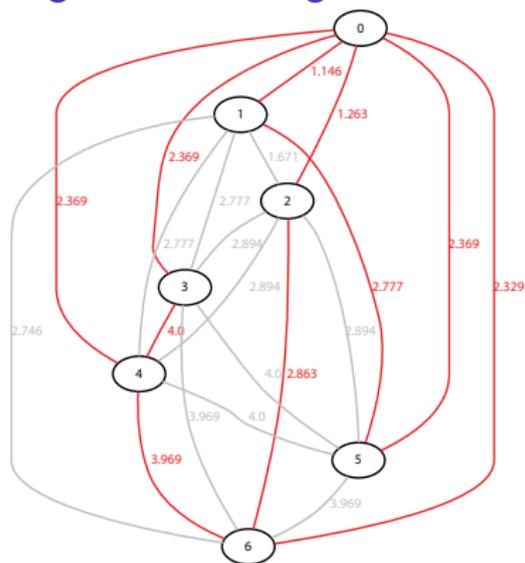
Ein konkretes Beispiel, Schritte 9-10



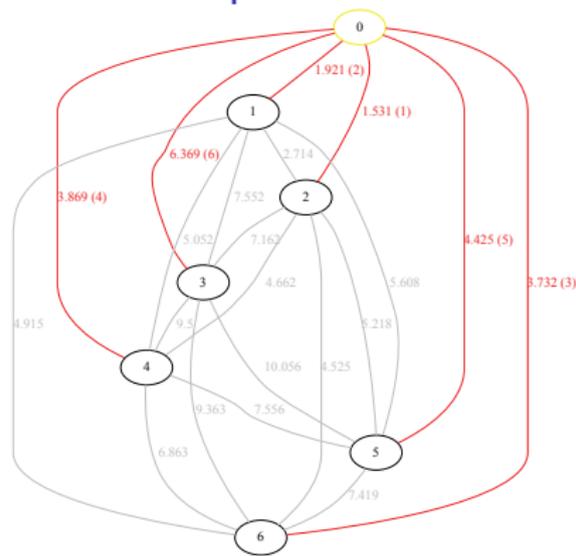
# Projektion des Dekodierungsproblem auf einen Graphen

Ein konkretes Beispiel, Schritt 11

## Ergebnis QWeighted



## Minimaler Spannbaum



## QWeighted als Suchalgorithmus

- ▶ Der minimale Spannbaum eines solch transformierten Dekodierungsproblem ist immer sternförmig (top rank Knoten besitzt  $n - 1$  inzidente Kanten)
- ▶ QWeighted bestimmt eine möglichst kleine Obermenge des minimalen Spannbaumes (markierte Kante bedeutet Klassifikationsauswertung)
- ▶ Dabei sind die Kantengewichte nicht vorgegeben sondern werden iterativ angenähert
- ▶ Durch Modifikation Bestimmung des exakten minimalen Spannbaumes möglich



# Quick Weighted Voting

## Eigenschaften

- ▶ Klassifikationsergebnis stimmt immer überein mit *Weighted Voting* (Bis auf mehrdeutige top ranks)
- ▶ Übernimmt somit die Eigenschaften (gute Genauigkeit, keine neuer empirischer Vergleich mit anderen Dekodierungsmethoden nötig)
- ▶ Reduzierung der Vergleiche nahezu unabhängig vom Basis-Lerner
- ▶ im worst-case Anzahl der Vergleiche identisch mit *Weighted Voting*
- ▶ im average-case  $n \log(n)$  und best-case  $n - 1$
- ▶ Ähnlichkeit zum A\* Algorithmus, spezielle Version des Kruskal Algorithmus

# Gliederung

## Kurze Einleitung

## Effiziente Klassifizierung

Dekodierung

Quick Weighted Voting

QWeighted als Suchalgorithmus

## Effizientes Ranking

Ranking

„Schweizer-System“

Erweitertes Schweizer-System

Erweiterte Auswertungen



# Ranking

## Gegeben:

gleiche Ausgangsstellung wie bei Klassifikation

## Gesucht:

Funktion, die jedem Testobjekt eine totale **Ordnung** der Klassen zuordnet

- ▶ Verallgemeinerung der Klassifikation

# Schweizer-System

- ▶ ist eine Turnierform, die bei Schach- und ähnlichen Sportarten benutzt wird
- ▶ entstand aus der Notwendigkeit, dass bei hoher Spieleranzahl unmöglich jeder gegen jeden spielen kann
- ▶ Idee:
  - ▶ Es spielen immer soweit möglich gleichstarke Spieler in jeder Runde. ( $\lfloor \frac{n}{2} \rfloor$  Paarungen)
  - ▶ Rundenzahl ist begrenzt. (z.B.  $\log(n)$ )
- ▶ Dadurch Anzahl der Spiele drastisch reduziert
- ▶ Anders als bei Systemen mit KO-Runden komplettes Ranking möglich



# Ergebnisse

- ▶ Ranking-Error
  - ▶ Das Verhältnis von Genauigkeitsverlust und Vergleichsreduzierung ist nahezu konstant.
  - ▶ Vermutlich alle Vergleiche nötig für korrektes Ranking
  - ▶ Das Schweizer-System ist einer zufälligen oder (Klassen-)geordneten Reihenfolge überlegen
- ▶ Position-Error
  - ▶ Für die Bestimmung des top rank sind nicht alle Vergleiche notwendig
  - ▶ Mit zunehmender Klassenanzahl konvergiert der Position-Error schneller gegen den Nullwert.



# Erweitertes Schweizer-System

## Sonneborn-Berger und Buchholz Feinwertung

- ▶ Sonneborn-Berger und Buchholz Feinwertung versuchen nicht eindeutige Zwischenrankings mithilfe von Spielinformationen weiter zu differenzieren
- ▶ Die SoBerg Wertung eines Spielers ist bestimmt durch die Summe der vollen Punktzahlen aller Gegner, gegen die er gewonnen hatte, sowie die halbe Punktzahl von allen Gegnern, gegen die er Remis gespielt hat. (Semantik: gewisse Transitivität)
- ▶ Die Buchholz Wertung eines Spielers ist bestimmt durch die Summe der Punktzahlen aller bisher gespielten Gegner, gleichgültig, ob Remis Gewinn oder Niederlage das Ergebnis war. (Semantik: Aussage über die Spielstärke der bisher gespielten Gegner)



# Erweitertes Schweizer-System

## Auswertungen

Datensatz	Ratio SoBerg 1	Ratio SoBerg 2
vehicle	1 ( $\pm 0$ )	1 ( $\pm 0$ )
glass	1.000164612 ( $\pm 0.000506713$ )	0.948780129 ( $\pm 0.036785133$ )
image	1.00016504 ( $\pm 0.00460769$ )	1.119807177 ( $\pm 0.109182789$ )
yeast	1.012155144 ( $\pm 0.027819882$ )	1.001444946 ( $\pm 0.042759935$ )
vowel	0.99837131 ( $\pm 0.004819358$ )	1.014541148 ( $\pm 0.029203963$ )
soybean	1.002016211 ( $\pm 0.079115186$ )	1.020374465 ( $\pm 0.087806017$ )
letter	1.0000142 ( $\pm 0.00012756$ )	1.002840891 ( $\pm 0.007778173$ )

Datensatz	Ratio Buchholz 1	Ratio Buchholz 2
vehicle	1 ( $\pm 0$ )	1 ( $\pm 0$ )
glass	1.000066755 ( $\pm 0.000694399$ )	0.948682274 ( $\pm 0.036850418$ )
image	0.99769493 ( $\pm 0.00648815$ )	1.119982024 ( $\pm 0.111612066$ )
yeast	1.012163307 ( $\pm 0.027793654$ )	1.000566501 ( $\pm 0.043557822$ )
vowel	0.997842851 ( $\pm 0.005085225$ )	1.015615179 ( $\pm 0.030460505$ )
soybean	0.953833147 ( $\pm 0.036558897$ )	1.007701388 ( $\pm 0.044429992$ )
letter	0.999988525 ( $\pm 6.69568E-05$ )	1.002311609 ( $\pm 0.007300357$ )

**Tabelle:** Auswertungen der Feinwertungen. Die zweite und dritte Spalte gibt den Mittelwert des Verhältnisses zwischen SoBerg 1/2 bzw. Buchholz 1/2 Ranking-Error zu Schweizer-System Ranking-Error wieder



## Erweiterte Auswertungen

- ▶ Auswirkungen bei beiden Versionen so gering, dass sie nahezu vernachlässigbar sind.
- ▶ Zur Bestimmung der Gründe wurden weitere Auswertungen durchgeführt.
- ▶ Ergebnis: Zusammenspiel mehrerer Faktoren für das Versagen der plausiblen Feinwertungen



## Erweiterte Auswertungen

- ▶ Feinwertungen kommen sehr selten zum Einsatz
  - ▶ zu geringe Anzahl an Gleichstandssituationen aufgrund weighted Klassifikatoren
  - ▶ dadurch geringer Einfluss auf Ranking-Error
- ▶ Tendenz neigt zu Verbesserungen, jedoch gleichen sich die Änderungsbeträge aus
- ▶ Informationsbasis gering → vage Vorhersage



# Erweiterte Auswertungen

## 1. Faktor: geringe Anzahl an Gleichständen

	SoBerg 1		SoBerg 2	
	G	GM	G	GM
vehicle	5 (0.099%)	17 (0.335%)	5 (0.099%)	17 (0.335%)
glass	35 (0.779%)	35 (0.779%)	41 (0.912%)	41 (0.912%)
image	4494 (9.264%)	7143 (14.725%)	6251 (12.886%)	6388 (13.168%)
yeast	89 (0.133%)	89 (0.133%)	85 (0.127%)	85 (0.127%)
vowel	4482 (8.231%)	4838 (8.885%)	4360 (8.007%)	4702(8.635%)
soybean	106976 (91.595%)	259298 (222%)	106538 (91.22%)	252172 (215.9%)
letter	39105 (3.008%)	40257 (3.097%)	39089 (3.007%)	40241 (3.095%)

	Buchholz 1		Buchholz 2	
	G	GM	G	GM
vehicle	5 (0.099%)	17 (0.335%)	5 (0.099%)	17 (0.335%)
glass	35 (0.779%)	35 (0.779%)	41 (0.912%)	41 (0.912%)
image	4494 (9.264%)	7143 (14.725%)	6076 (12.525%)	6215 (12.812%)
yeast	89 (0.133%)	89 (0.133%)	85 (0.127%)	175 (0.262%)
vowel	4482 (8.231%)	4838 (8.885%)	4373 (8.031%)	4717 (8.663%)
soybean	106976 (91.595%)	259298 (222%)	106443 (91.138%)	253453 (217%)
letter	39105 (3.008%)	40257 (3.097%)	39044 (3.003%)	40196 (3.092%)

**Tabelle:** Erweiterte Auswertung: Anzahl der Gleichstände für die Feinwertungen (G - Gleichstände, GM - Gleichstandsmengen)



# Erweiterte Auswertungen

## 2. Faktor: nahezu ausgleichende Änderungsbeträge

	SoBerg 1			SoBerg 2		
	S	VB	VS	S	VB	VS
vehicle	2	0	0	2	0	0
glass	8	0	2	8	0	2
image	2145	545	510	1869	557	280
yeast	43	10	1	39	10	1
vowel	1707	604	452	1678	630	453
soybean	236100	81056	77814	232180	84770	71600
letter	7275	2357	2610	7257	2343	2622

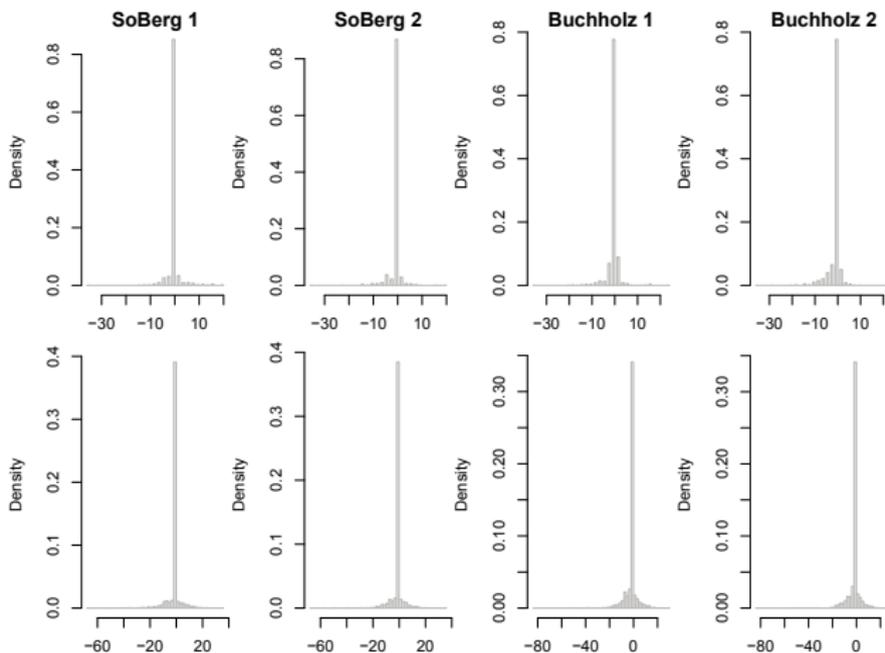
	Buchholz 1			Buchholz 2		
	S	VB	VS	S	VB	VS
vehicle	2	0	0	2	0	0
glass	14	1	2	20	1	2
image	4507	812	772	3321	992	391
yeast	83	39	4	123	54	5
vowel	2541	975	564	2587	929	569
soybean	238109	99891	56817	233654	100277	54144
letter	8854	3025	2509	8793	2968	2509

**Tabelle:** Erweiterte Auswertung: Auswirkungen der Feinwertungen (S-Sortierung möglich, VB-Verbesserung, VS-Verschlechterung)



# Erweiterte Auswertungen

2. Faktor: nahezu ausgleichende Änderungsbeträge (oben: vowel, unten:image)



# Erweiterte Auswertungen

## 3. Faktor: geringe Informationsbasis

	SoBerg 1		SoBerg 2	
	durchschn. SPA	durchschn. KL	durchschn. SPA	durchschn. KL
vehicle	2	2 (33.3%)	2	2 (33.3%)
glass	2	2.75 (13.1%)	2	2.75 (13.1%)
image	2.119	3.373 (16.1%)	2.097	3.212 (15.3%)
yeast	2.022	2.465 (5.5%)	2.024	2.093 (4.7%)
vowel	2.204	3.098 (5.6%)	2.205	2.917 (5.3%)
soybean	2.912	50.833 (29.7%)	2.945	51.44 (30.1%)
letter	2.671	2.247 (0.7%)	2.671	2.240 (0.7%)

	Buchholz 1		Buchholz 2	
	durchschn. SPA	durchschn. KL	durchschn. SPA	durchschn. KL
vehicle	2	2 (33.3%)	2	2 (33.3%)
glass	2	5.571 (26.5%)	2	4.35 (20.7%)
image	2.119	8.766 (41.7%)	2.099	7.58 (36.1%)
yeast	2.022	5.795 (12.9%)	2.023	5.642 (12.5%)
vowel	2.204	6.203 (11.3%)	2.205	5.551 (10.1%)
soybean	2.912	72.865 (42.6%)	2.927	72.629 (42.5%)
letter	2.671	3.340 (1%)	2.672	3.304 (1%)

**Tabelle:** Erweiterte Auswertung: Einflussbereich der Feinwertungen (SPA - Anzahl Spieler einer Gleichstandsmenge, KL - Anzahl Spiele / Klassifikationsauswertungen)

