

# Combinando métodos de seleção de atributos usando agregação de *rankings*

Ronaldo C. Prati

<sup>1</sup>Universidade Federal do ABC - UFABC  
Centro de Matemática Computação e Cognição  
Rua Santa Adélia, 166, Santo André, SP, Brasil

ronaldo.prati@ufabc.edu.br

**Abstract.** *This paper investigates the use of ranking aggregation for combining feature selection methods that produces a ranked list of the features. We investigate the use of Borda, Condorcet and Kemeny methods. Results show that ranking aggregation greatly improves the quality of induced classifiers using a reduced number of features, when compared to the base feature selection methods.*

**Resumo.** *Neste trabalho investigamos o uso de agregação de rankings para combinar métodos de seleção de atributos que produzem uma lista ordenada de atributos. Três métodos de agregação de rankings são investigados: Borda, Condorcet e Kemeny. Resultados experimentais mostram que o uso de agregação de rankings tem um bom desempenho quando comparados com os métodos de seleção de atributos usados na composição, gerando modelos de classificação mais precisos e com menos atributos.*

## 1. Introdução

Seleção de atributos é um tópico de pesquisa muito ativo em áreas como aprendizado de máquina, mineração de dados e bioinformática (Guyon and Elisseeff, 2003). A principal ideia da seleção de atributos é encontrar um subconjunto dos atributos originais, removendo aqueles atributos que são pouco discriminativos para o problema em questão. Em muitos casos, a seleção de atributos pode melhorar significativamente o desempenho de algoritmos que fazem a geração de modelos a partir dos dados, bem como produzir modelos mais compactos e/ou que possam ser mais facilmente interpretáveis por seres humanos. Além disso, em vários domínios, a seleção de atributos é importante por si só pois ajuda a entender melhor quais são os atributos que estão relacionados ao problema. Por exemplo, médicos podem ter uma ideia mais precisa a respeito das causas e consequências de uma certa doença analisando um subconjunto dos atributos disponíveis.

Métodos de seleção de atributo geralmente usam busca heurística para encontrar subconjuntos de atributos. Isso se deve ao fato que uma busca completa é computacionalmente intratável, devido à explosão exponencial das alternativas. Métodos heurísticos podem ocorrer em mínimos locais e não serem capazes de encontrar subconjuntos ótimos. Em aprendizado de máquina, a combinação de métodos (tal como em *ensembles* ou comitês de classificadores) tem sido geralmente utilizada para tentar contornar esse problema. Entretanto, combinação de métodos de seleção de atributos é pouco explorado na literatura.

Em (Prati and Monard, 2006), propusemos a combinação de métodos de seleção de atributos usando agregação de *rankings*. Neste trabalho, esse estudo é substancialmente estendido. Prati and Monard (2006) consideraram apenas um método de agregação de *rankings* (equivalente ao método de Borda, que também é tratado neste trabalho). No presente trabalho outros dois métodos são investigados: Condorcet e Kemeny. Além disso, no estudo anterior, os experimentos foram realizados com um número pré-definido de atributos. Nesse trabalho investigamos o desempenho dos métodos em todo o espectro de atributos, além de incluirmos novos conjuntos de dados (33 no total). Resultados experimentais mostram que a abordagem de agregação de *rankings* produziu resultados muito interessantes, com taxas de erro menores que outros métodos com uma menor proporção de exemplos que os outros métodos.

Esse artigo está organizado da seguinte maneira: na Seção 2 é apresentada uma breve revisão sobre métodos de seleção de atributos. Na Seção 3 é apresentada uma discussão sobre métodos de agregação de *rankings* utilizados neste trabalho para a combinação de métodos de seleção de atributos. Na Seção 4 são apresentados os resultados experimentais obtidos, seguido das conclusões, apresentadas na Seção 5.

## 2. Seleção de subconjuntos de atributos

Os objetivos da seleção de atributos para a aplicação de algoritmos de aprendizado são três (Guyon and Elisseeff, 2003): melhorar o desempenho (geralmente medido em termos da taxa de acerto) dos modelos; criar modelos mais simples e que envolvam menos atributos; e melhorar o entendimento do domínio para os quais os dados foram coletados. Em linhas gerais, na seleção de subconjuntos de atributos estamos interessados em procurar o melhor subconjunto de atributos, de acordo com algum critério de qualidade. Idealmente, o melhor subconjunto contém o menor número de atributos que mais contribuem para construir o modelo. Os outros atributos não são importantes e podem ser descartados. A seleção de um subconjunto de atributos pode ser modelada como um problema de busca, e vários métodos foram desenvolvidos para encontrar o “melhor” subconjunto. Infelizmente, testar todos os possíveis subconjuntos leva a uma explosão combinatorial, a medida que o número de atributos cresce. Duas abordagens são geralmente utilizadas para evitar a busca exaustiva (Vafaie and Jong, 1993). A primeira delas é construir estratégias *ad-hoc* para diminuir o espaço de busca a um tamanho tratável. A segunda utiliza busca heurística (geralmente algoritmos do tipo *hill-climbing* ou busca em feixe) e é usada em domínios nos quais o conhecimento a respeito do domínio é caro ou difícil de se obter. Como o uso de abordagens *ad-hoc* são restritas a domínios específicos, elas são de menor interesse do ponto de vista de métodos gerais de seleção de atributos.

Algoritmos para a seleção de atributos que utilizam busca heurística podem ser caracterizados como *wrappers*, filtros ou embutidos. A abordagem *wrapper* (Kohavi and John, 1997) utiliza a precisão do modelo induzido com o subconjunto de atributos por algum algoritmo de aprendizado para guiar a busca. Um dado atributo é selecionado se ele melhora a precisão do modelo. Em contrapartida, a abordagem filtro (Duch, 2006) utiliza medidas baseadas em característica do conjunto de dados para selecionar os atributos. A abordagem mais comum é ordenar os atributos utilizando algum critério, e filtrar aqueles que aparecem mais embaixo na lista ordenada. Finalmente, a abordagem embutida (Lal et al., 2006) é um método indireto que seleciona atributos por meio de um algoritmo de aprendizado que implemente internamente algum método de seleção de atributos.

A abordagem embutida é intrínseca a alguns algoritmos de aprendizado e, dessa maneira, somente os algoritmos projetados com essa característica podem ser utilizados. A abordagem filtro é, em geral, computacionalmente menos complexa. A abordagem *wrapper* tende a produzir resultados melhores (Appice et al., 2004), mas ao custo de uma maior complexidade computacional. Além disso, a busca é ajustada principalmente para o algoritmo de aprendizado utilizado como *wrapper*, e o subconjunto encontrado pode não ser o mais apropriado para um outro algoritmo. Filtros, no entanto, são muito flexíveis, uma vez que qualquer algoritmo de aprendizado pode utilizar os atributos selecionados.

### 3. Agregação de *rankings*

Neste trabalho investigamos métodos de agregação de *rankings* dentro do contexto de seleção de atributos. A ideia básica é tentar combinar diferentes métodos de seleção de atributos que geram *rankings* de atributos em um novo *ranking*. A hipótese de pesquisa é que, como encontrar o melhor subconjunto de atributos é computacionalmente intratável e métodos de seleção de atributos mais utilizados são geralmente baseados em heurísticas, uma combinação de diversos métodos possa produzir melhores resultados (modelos mais compactos e com melhor taxa de acerto) do que a utilização dos métodos isoladamente.

Optou-se por estratégias de combinação de *rankings* por dois motivos: os valores associados a cada atributo em diferentes métodos podem estar em escalas diferentes. Por exemplo, um método A atribui uma pontuação para os atributos entre 0 e 1, enquanto que um outro método B atribui uma pontuação entre -1 e 1. Além disso, mesmo que a escala absoluta seja a mesma, *i.e.*, ambas os métodos atribuam pontuação para os atributos entre 0 e 1 por exemplo, os pesos relativos podem ser diferentes. Em outras palavras, uma pontuação 0.8 no método A pode ter um peso diferente de uma pontuação 0.8 no método B. Combinar diretamente esses métodos utilizando os valores atribuídos a cada atributo por cada método deveria levar em consideração esses dois fatores, promovendo um ajuste de escala e ponderando os pesos atribuídos a cada método. Agregação de *rankings* contorna esses dois problemas de uma maneira simples, desconsiderando a escala e peso dos valores e levando em consideração apenas a ordem relativa dada pelos métodos.

Informalmente, o problema de agregação de *rankings* é combinar diferentes ordenações das alternativas dadas por diferentes *rankings* em um único *ranking*, de tal maneira a obter uma “melhor” reordenação dessas alternativas. Agregação de *rankings* tem sido estudada em diferentes áreas, com maior ênfase em ciências sociais, na qual diferentes esquemas de votação foram propostos para se escolher candidatos em eleições.

Alguns desses métodos são bem antigos, tendo sido propostos na segunda metade do século XVIII. Um desses métodos é a “contagem de Borda”, proposto por Jean-Charles de Borda por volta de 1770. A ideia consiste em somar as posições dos *rankings*, e reordenar os candidatos de acordo com essa soma. Para o problema de *ranking* de atributos, considere que para um conjunto de três atributos  $A_1, A_2, A_3$ , um método  $\tau_1$  ordene os atributos na ordem  $A_1, A_2, A_3$  e um outro método  $\tau_2$  ordene os atributos na ordem  $A_3, A_1, A_2$ . Combinando esses dois *rankings* com o método de Borda, teríamos o novo *ranking*  $A_1, A_3, A_2$  pois a soma dos *rankings*  $\tau_1$  e  $\tau_2$  é 3, 5 e 4 para os atributos  $A_1, A_2, A_3$ , respectivamente.

Outro método contemporâneo ao de Borda foi proposto por Marie J. A. N. Caritat, Marquês de Condorcet, em 1785. A ideia por trás desse método é que os candidatos nas

primeiras posições do novo *ranking* devem ser aqueles que recebam o maior número de votos, em comparações por pares. No exemplo anterior, o atributo  $A_1$  receberia três votos ( $A_1$  está em melhor posição que  $A_2$  e  $A_3$  em  $\tau_1$  e está em melhor posição que  $A_2$  em  $\tau_2$ ),  $A_2$  receberia apenas um voto ( $A_2$  está em melhor posição que  $A_3$  em  $\tau_1$ ) e  $A_3$  receberia 2 votos ( $A_3$  está em melhor posição que  $A_2$  e  $A_1$  em  $\tau_2$ ). O método de Condorcet tende a encontrar o “melhor compromisso” entre as alternativas. Um alternativa pode ser a melhor no geral sem ser a primeira colocada em nenhum *ranking*, desde que ela não aparece nas piores posições na maioria dos *rankings*. Um problema com o método de Condorcet é que nem sempre é possível encontrar um vencedor, pois havendo mais de três candidatos e três *rankings* podem haver ciclos (mais de um candidato com o mesmo número de votos).

Um método mais recente é conhecido como método de Kemeny. Ele produz um novo *ranking* que minimiza o número de pares de concordância entre os *rankings*. Mais formalmente, assuma que  $\phi$  e  $\sigma$  sejam dois *rankings* que ordenam um conjunto de elementos  $\{1, \dots, n\}$ .  $\phi(i)$  e  $\sigma(j)$  correspondem à posição do elemento  $i$  nos *rankings*  $\phi$  e  $\sigma$ , respectivamente. Existe um par de concordância nos *rankings* para um dado par de elementos  $i, j$  se  $\phi(i) < \phi(j)$  e  $\sigma(i) < \sigma(j)$ , ou seja, o elemento  $i$  deve aparecer em uma melhor posição do que o elemento  $j$  nos dois *rankings*  $\phi(i)$  e  $\sigma(j)$ . O número de pares de concordância é proporcional à distância de Kendall-tau entre os dois *rankings*. Essa distância mede o número de transposições do tipo “*bubble sort*” necessárias para transformar o *ranking*  $\phi$  no *ranking*  $\sigma$ . O *ranking* de Kemeny é aquele que minimiza a soma das distâncias de Kendall-tau entre ele e todos os outros *rankings*. O *ranking* de Kemeny tem uma importante interpretação como sendo o estimador de máxima verossimilhança do *ranking* verdadeiro. Entretanto, computar o *ranking* de Kemeny é um problema NP-Difícil mesmo para problemas com apenas 4 alternativas (Dwork et al., 2001).

Adali et al. (2006) propõe um método baseado no algoritmo PageRank para aproximar o *ranking* de Kemeny. Esse método funciona da seguinte maneira: para cada alternativa (dentro do contexto deste artigo, uma alternativa é um atributo) um nó é inserido em um grafo. Uma aresta é inserida nesse grafo ligando a alternativa  $i$  à alternativa  $j$  sempre que  $i$  aparece antes de  $j$  em um *ranking*. Um peso é atribuído a essas arestas correspondendo à distância entre as duas alternativas no *ranking*<sup>1</sup>. Esses pesos são normalizados de tal maneira que os pesos das arestas que partam de um nó somem 1. O algoritmo PageRank é então aplicado ao grafo, e o *ranking* dos nós do grafo calculado por esse algoritmo é usado como uma aproximação do *ranking* de Kemeny. Essa abordagem é equivalente ao uso de cadeias de Markov para aproximar o *ranking* de Kemeny, como proposto em (Dwork et al., 2001), tendo a vantagem de ser computacionalmente eficiente. Nesse trabalho investigamos essas três estratégias (contagem de Borda, Condorcet e Kemeny) de composição de *rankings* para compor *rankings* de atributos gerados por métodos de seleção de atributos.

#### 4. Avaliação experimental

Para testar os métodos propostos neste trabalho foram utilizados 33 conjuntos de dados, obtidos a partir da página do projeto Weka<sup>2</sup>. Os desenvolvedores do Weka (Witten and Frank, 2005) obtiveram esses conjuntos de dados a partir do repositório de dados da

<sup>1</sup>Se uma aresta ligando  $i$  e  $j$  já existe, somente os pesos são atualizados.

<sup>2</sup><http://prdownloads.sourceforge.net/weka/datasets-UCI.jar>

UCI (Asuncion and Newman, 2007) e disponibilizam esses conjuntos já convertidos para o formato de dados utilizado na ferramenta. Nenhum outro pré-processamento foi aplicado aos conjuntos de dados. Na Tabela 1 é apresentado um resumo sobre as características desses conjuntos de dados. Para cada conjunto de dados é mostrado um número identificador (#); bem como o nome (nome); número de atributos (# atributos), sendo o número de atributos discretos/contínuos mostrado entre parênteses; número de exemplos (# ex.); número de classes (# cl.); e porcentagem de exemplos na classe majoritária (cl. maj.). Como pode ser observado, são conjuntos de dados bem heterogêneos, com diferentes proporções de atributos contínuos/discretos, número de exemplos, classes e porcentagem de exemplos na classe majoritária.

#	nome	# atributos	# ex.	# cl.	cl. maj.	#	nome	# atributos	# ex.	# cl.	cl. maj.
1	anneal	39 (33/6)	898	6	76,17%	2	anneal.orig	39 (33/6)	898	6	76,17%
3	audiology	70 (70/0)	226	24	25,22%	4	autos	26 (11/15)	205	7	32,68%
5	balance-scale	5 (1/4)	625	3	46,08%	6	breast-cancer	10 (10/0)	286	2	70,28%
7	wisconsin(bc)	10 (1/9)	699	2	65,52%	8	horse-colic	23 (16/7)	368	2	63,04%
9	horse-colic.orig	28 (21/7)	368	2	66,30%	10	credit-rating	16 (10/6)	690	2	55,51%
11	german	21 (14/7)	1000	2	70,00%	12	pima	9 (1/8)	768	2	65,10%
13	glass	10 (1/9)	214	7	35,51%	14	cleveland	14 (8/6)	303	5	54,46%
15	heart-statlog	14 (1/13)	270	2	55,56%	16	hepatitis	20 (14/6)	155	2	79,35%
17	ionosphere	35 (1/34)	351	2	64,10%	18	iris	5 (1/4)	150	3	33,33%
19	kr-vs-kp	37 (37/0)	3196	2	52,22%	20	labor	17 (9/8)	57	2	64,91%
21	letter	17 (1/16)	20000	26	4,06%	22	lymphography	19 (16/3)	148	4	54,73%
23	mushroom	23 (23/0)	8124	2	51,80%	24	primary-tumor	18 (18/0)	339	22	24,78%
25	segment	20 (1/19)	2310	7	14,29%	26	sonar	61 (1/60)	208	2	53,37%
27	soybean	36 (36/0)	683	19	13,47%	28	splice	62 (62/0)	3190	3	51,88%
29	vehicle	19 (1/18)	846	4	25,77%	30	vote	17 (17/0)	435	2	61,38%
31	vowel	14 (4/10)	990	11	9,09%	32	waveform	41 (1/40)	5000	3	33,84%
33	zoo	18 (17/1)	101	7	40,59%						

**Tabela 1. Descrição dos conjuntos de dados usados nos experimentos**

Os métodos de seleção de atributos utilizados como *ranking* base estão implementados na ferramenta Weka. Foram utilizados todos os métodos supervisionados que implementam a classe `weka.attributeSelection.AttributeEvaluator` do `weka`. Os métodos utilizados foram:

**Qui-quadrado ( $\chi^2$ )** Avalia a qualidade de um atributo computando a estatística  $\chi^2$  com relação à classe.

**Ganho de informação (IG)** Avalia a qualidade de um atributo computando o ganho de informação com relação à classe. O ganho de informação é obtido subtraindo-se a entropia do atributo classe pela entropia condicional do atributo dada a classe.

**Razão do ganho (GR)** É uma variação do ganho da informação para compensar atributos com muitos valores possíveis. A razão do ganho pondera o ganho de informação, dividindo o ganho de informação pela entropia do atributo.

**Incerteza simétrica (SU)** É uma outra variação do ganho de informação. Ela também pondera o ganho de informação, mas dividindo o ganho de informação pela entropia do atributo somada à entropia da classe. Esse valor é multiplicado por 2 para que os valores estejam entre zero e um.

**ReliefF** É um algoritmo iterativo que a cada iteração atualiza a pontuação para cada atributo (Kononenko, 1994). A cada iteração, um exemplo é aleatoriamente escolhido e  $k$  vizinhos de cada uma das classes são encontrados. A pontuação é então atualizada para cada atributo com base na diferença entre os valores do atributo e os valores dos atributos dos  $k$  vizinhos da mesma classe e das outras classes. Diferença entre valores dos atributos em exemplos da mesma classe são penalizados e diferenças entre valores dos atributos em exemplos de classes diferentes são recompensados.

**Regra Simples** (*OneR*) A qualidade do atributo é proporcional a quão bem ele, isoladamente, classifica os exemplos.

Todos os métodos foram utilizados com os valores padrão dos seus respectivos parâmetros. Os métodos de agregação de *rankings* Borda, Condorcet e Kemeny foram implementados em java, e usavam como entrada os *rankings* gerados pelos métodos de seleção de atributos descritos anteriormente. Para os métodos Condorcet e Borda, empates (caso houvessem) eram quebrados aleatoriamente. Para construir o *ranking* de Kemeny foi utilizada a implementação do PageRank disponível no pacote Jung<sup>3</sup>, com os parâmetros padrão definidos no pacote.

Os experimentos foram realizados utilizando-se validação cruzada com 10 partições. Além disso, a validação cruzada foi repetida 10 vezes (10x10 *fold cross-validation*). Para avaliar o desempenho dos métodos de seleção de atributos e também os métodos que fazem a agregação de *rankings* utilizamos o seguinte procedimento: para cada *ranking* (tanto os diretamente gerados por um método de seleção de atributos quanto aqueles obtidos por meio da agregação desses *rankings*), o algoritmo Naïve Bayes<sup>4</sup> (também utilizamos o algoritmo implementado no Weka e com parâmetros padrão) foi utilizado para induzir modelos. Primeiramente, o atributo melhor posicionado no *ranking* foi utilizado para induzir um modelo e a taxa de erro no conjunto de teste calculada. A seguir, um novo modelo foi gerado utilizando os dois atributos melhor posicionados no *ranking* e a taxa de erro desse modelo calculada, e assim sucessivamente até que todos os atributos fossem utilizados. Finalmente, a taxa de erro média dessas execuções que contempla modelos gerados com um até o número de atributos foi calculada. Os resultados experimentais dessas médias estão sumarizados na Tabela 2.

Os métodos Borda e Condorcet tiveram um desempenho praticamente idêntico em todos os conjuntos de dados (existe uma pequena diferença apenas no conjunto #2). Uma explicação para esse fato é que, nos experimentos realizados, o critério de Borda foi consistente com o de Condorcet, ou seja, o critério de Borda ordena os atributos da mesma maneira que o de Condorcet.

Em geral, como pode ser observado na tabela, os métodos que fazem a agregação de *rankings* apresentam uma taxa de erro média consideravelmente menor do que a taxa de erro média dos métodos de seleção de atributos isoladamente. Para alguns conjuntos

---

<sup>3</sup><http://jung.sourceforge.net>

<sup>4</sup>Escolhemos o Naïve Bayes pois esse algoritmo leva em consideração todo o conjunto de atributos fornecido a ele. Outros algoritmos (árvore de decisão, por exemplo) fazem internamente uma seleção de atributos ou modificam o espaço de exemplos (*kernels* em SVMs), o que poderia dificultar a análise dos métodos propostos. O desempenho desses algoritmos de aprendizado com os métodos propostos é um interessante tema para pesquisas futuras.

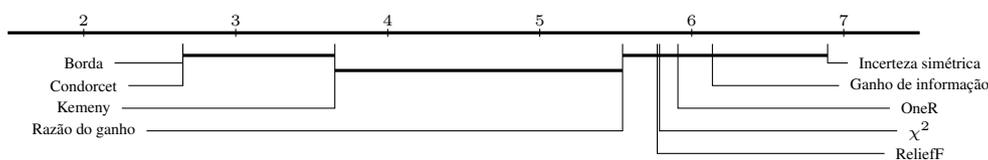
#	Borda	Condorcet	Kemeny	$\chi^2$	IG	GR	SU	Relief	OneR
1	18.96	18.98	21.60	27.19	25.59	24.27	27.14	26.75	28.09
2	27.94	27.57	33.15	26.61	26.09	26.10	26.47	29.81	26.12
3	35.72	35.72	55.48	60.89	65.87	56.49	65.77	65.20	65.68
4	44.19	44.19	54.42	57.92	58.15	53.77	57.73	57.96	54.59
5	23.25	23.25	24.68	20.83	20.71	21.07	20.48	22.34	21.39
6	27.13	27.13	27.09	30.34	29.69	30.96	30.54	26.36	27.43
7	4.19	4.19	6.92	6.79	6.92	5.08	6.78	7.95	6.71
8	21.11	21.11	28.11	34.10	34.05	32.94	33.40	33.81	32.19
9	36.12	36.12	35.70	35.13	35.03	34.29	34.49	29.37	37.86
10	27.17	27.17	26.44	33.87	33.95	31.38	33.27	27.55	33.42
11	27.12	27.12	30.12	30.08	30.05	29.98	30.19	30.04	28.40
12	24.95	24.95	30.57	32.58	32.82	32.35	32.74	32.19	32.63
13	53.13	53.13	61.41	61.77	62.07	62.12	62.32	57.68	61.97
14	24.67	24.67	16.60	29.43	29.45	30.57	30.42	26.23	29.83
15	26.30	26.30	20.31	32.76	32.76	33.16	33.11	30.54	32.88
16	19.57	19.57	19.28	22.40	21.74	21.74	21.58	19.80	20.95
17	13.23	13.23	27.15	23.22	24.19	26.82	28.49	24.53	24.77
18	16.67	16.67	5.33	18.50	19.00	20.00	20.00	19.67	19.33
19	29.68	29.68	21.96	40.27	40.19	38.68	40.11	37.96	40.56
20	15.42	15.42	14.48	21.56	21.35	20.83	21.88	19.17	14.17
21	66.72	66.72	48.09	68.47	68.89	68.27	69.03	69.33	68.57
22	22.59	22.59	20.04	27.37	28.93	27.81	30.63	32.26	32.26
23	7.86	7.86	8.94	22.78	21.96	21.29	23.60	22.04	23.22
24	60.16	60.16	62.28	67.28	67.61	66.02	67.30	67.89	62.21
25	36.92	36.92	25.20	48.77	48.57	48.68	48.87	48.49	49.21
26	33.96	33.96	35.04	33.38	33.42	33.31	33.33	38.25	34.12
27	27.68	27.68	28.60	38.47	40.78	34.17	40.41	41.72	35.83
28	24.00	24.00	24.34	39.27	39.31	39.51	39.32	39.99	33.51
29	54.26	54.26	56.82	54.08	54.47	54.67	54.84	53.39	53.41
30	12.41	12.41	14.93	21.08	21.11	21.16	21.19	17.61	20.16
31	58.48	58.48	58.17	76.05	76.07	75.65	76.36	77.93	78.50
32	25.69	25.69	48.78	50.13	50.12	50.25	50.15	51.61	50.13
33	17.45	17.45	23.27	28.96	29.22	31.61	29.27	27.25	29.93

**Tabela 2. Taxa de erro média obtida variando-se o número de atributos usados para construir o modelo usando os *rankings* gerados por cada método**

de dados, a taxa de erro média obtida a partir dos *rankings* agregados está por volta da metade da taxa de erro média dos *rankings* individuais. Isso ocorre, por exemplo, para os conjuntos de dados #3, #23, # 32 quando comparados com o Condorcet e Borda, e para os conjuntos de dados #14, #18, #19 e #23 e #25 quando comparados com o Kemeny. Apenas para o conjunto de dados #5, os métodos de agregação foram superados por todos os *rankings* individuais. Uma possível explicação para esse fato pode ser o reduzido tamanho desse conjunto (apenas 5 atributos e 625 exemplos).

Para testar se existem diferenças significativas com relação à média das taxas de

erros aplicamos e teste de Friedman (Demšar, 2006) nos resultados apresentados na Tabela 2. Como esse teste apontou que existem diferenças significativas com um nível de confiança de 95%, utilizamos o teste *post-hoc* de comparações múltiplas de Nemenyi para encontrar essas diferenças (também com um nível de confiança de 95%). Os resultados desse teste são mostrados graficamente na Figura 1 como um diagrama de diferenças críticas. Nesse diagrama, quanto menor o valor indicado no eixo, melhor o desempenho do método. O teste de Nemenyi detectou três grupos em que não é possível encontrar diferenças significativas entre eles. Esses grupos são indicados por uma linha conectando os métodos em que não há diferenças significativas. O primeiro grupo é composto pelos métodos de composição de *rankings*: Condorcet, Borda e Kemeny. Eles tiveram desempenho melhor que os métodos de seleção de atributos que criam o *ranking* individualmente, e formam o segundo grupo. O terceiro grupo é formado pelo *ranking* de Kemeny e o método da Razão de Ganho. Apesar de existir uma boa diferença entre os dois, não é possível afirmar que exista diferença significativa entre eles (segundo o teste de Nemenyi, essa distância entre os dois métodos no gráfico deve ser superior a 2,09 para que uma diferença significativa seja apontada).



**Figura 1. Diagrama de diferenças críticas calculado usando o teste de Nemenyi nos resultados apresentados na Tabela 2**

Apesar da média da taxa de erro ser um bom indicativo da eficiência dos métodos, ela dá apenas uma ideia geral do comportamento dos métodos, não sendo possível ter uma ideia do desempenho dos classificadores gerados. Na Tabela 3 são apresentadas as menores taxas de erro<sup>5</sup> obtidas com cada um dos métodos de seleção de atributos, incluindo aqueles baseados em *rankings* agregados. Os números entre parênteses indicam qual é a porcentagem de atributos que foi utilizada para a geração dos modelos. A última coluna indica a taxa de erro do modelo construído com todos os atributos. As menores taxas de erro para cada conjunto de dados estão hachuradas em cinza. Caso haja métodos empatados com a menor taxa de erro, aquela(s) obtida(s) com a menor porcentagem de atributos foi hachurada.

Somente para dois conjuntos de dados (#5 e #24), a menor taxa de erro não ficou abaixo daquela obtida usando todos os atributos. Como pode ser observado, existe uma predominância de células hachuradas nas três primeiras colunas, ou seja, a maioria das menores taxas de erro foram obtidas pelos métodos baseados em agregação de *rankings*. Também é interessante notar que, na maioria dos casos, as taxas de erro mínimas dos métodos baseados em agregação de *rankings* foram obtidas com as menores proporções de atributos utilizados para construir o modelo. Esse é um outro indicativo da eficiência dos métodos de agregação de *rankings* para compor os *rankings* gerados pelos métodos de seleção de atributos.

<sup>5</sup>É importante ressaltar que esse procedimento não pode ser usado para selecionar o modelo com menor taxa de acerto, pois os resultados estão calculados no conjunto de teste. Essa abordagem foi adotada apenas para ilustrar o desempenho dos algoritmos.

#	Borda	Condorcet	Kemeny	$\chi^2$	IG	GR	SU	Relief	OneR	Todos
1	13.56(94.74)	13.56(94.74)	11.11(76.32)	10.44(92.11)	13.56(100.00)	11.44(71.05)	12.00(94.74)	13.56(100.00)	12.00(92.11)	13.56
2	16.22(7.89)	16.22(7.89)	21.78(84.21)	18.56(92.11)	22.67(63.16)	20.22(94.74)	22.67(63.16)	22.89(60.53)	23.00(68.42)	24.56
3	26.52(86.96)	26.52(86.96)	29.57(100.00)	27.39(98.55)	29.57(100.00)	27.39(92.75)	29.57(100.00)	29.57(100.00)	29.57(100.00)	29.57
4	36.67(60.00)	36.67(60.00)	41.90(100.00)	40.48(92.00)	40.48(96.00)	40.95(96.00)	40.00(96.00)	40.95(96.00)	40.95(96.00)	41.90
5	8.73(100.00)	8.73(100.00)	8.73(100.00)	8.73(100.00)	8.73(100.00)	8.73(100.00)	8.73(100.00)	8.73(100.00)	8.73(100.00)	8.73
6	23.45(66.67)	23.45(66.67)	22.76(66.67)	25.17(100.00)	25.17(88.89)	25.17(100.00)	25.17(100.00)	24.48(22.22)	25.17(100.00)	25.17
7	2.57(66.67)	2.57(66.67)	2.71(100.00)	2.71(88.89)	2.71(88.89)	2.71(100.00)	2.71(88.89)	2.71(100.00)	2.57(88.89)	2.71
8	17.03(4.55)	17.03(4.55)	22.97(27.27)	23.51(100.00)	23.51(100.00)	23.51(100.00)	23.51(100.00)	23.51(100.00)	23.51(100.00)	23.51
9	32.97(3.70)	32.97(3.70)	30.00(25.93)	31.62(88.89)	31.62(88.89)	32.16(85.19)	31.62(88.89)	25.14(11.11)	32.97(3.70)	35.95
10	14.49(66.67)	14.49(66.67)	20.00(86.67)	20.58(100.00)	20.58(100.00)	20.58(100.00)	20.58(100.00)	20.58(100.00)	20.58(100.00)	20.58
11	25.60(55.00)	25.60(55.00)	27.20(100.00)	27.20(100.00)	27.20(100.00)	27.20(100.00)	27.20(100.00)	27.20(100.00)	26.60(65.00)	27.20
12	23.12(75.00)	23.12(75.00)	24.03(87.50)	24.16(100.00)	24.16(100.00)	24.16(100.00)	24.16(100.00)	24.16(100.00)	24.16(100.00)	24.16
13	47.27(77.78)	47.27(77.78)	49.55(100.00)	49.55(100.00)	49.55(100.00)	47.73(77.78)	49.55(100.00)	49.55(100.00)	49.55(100.00)	49.55
14	14.84(100.00)	14.84(100.00)	12.90(92.31)	14.84(100.00)	14.84(100.00)	14.84(100.00)	14.84(100.00)	14.84(100.00)	14.84(100.00)	14.84
15	17.78(100.00)	17.78(100.00)	16.30(92.31)	17.78(100.00)	17.78(100.00)	17.78(100.00)	17.78(100.00)	17.78(100.00)	17.78(100.00)	17.78
16	16.25(94.74)	16.25(94.74)	15.62(15.79)	17.50(89.47)	16.88(73.68)	16.25(94.74)	16.25(63.16)	16.88(94.74)	16.25(94.74)	18.75
17	9.01(17.65)	9.01(17.65)	16.34(94.12)	16.34(100.00)	16.34(100.00)	16.34(100.00)	16.34(100.00)	16.33(97.06)	15.78(97.06)	16.34
18	4.67(100.00)	4.67(100.00)	4.67(25.00)	4.67(100.00)	4.67(100.00)	4.67(100.00)	4.67(100.00)	4.67(100.00)	4.67(100.00)	4.67
19	12.31(94.44)	12.31(94.44)	8.31(75.00)	12.50(100.00)	12.50(100.00)	12.50(100.00)	12.50(100.00)	12.50(100.00)	12.50(100.00)	12.50
20	8.33(87.50)	8.33(87.50)	6.67(93.75)	8.33(93.75)	8.33(93.75)	8.33(93.75)	8.33(93.75)	5.00(93.75)	5.00(56.25)	10.00
21	36.03(100.00)	36.03(100.00)	34.31(68.75)	35.98(93.75)	36.03(100.00)	36.03(100.00)	36.03(100.00)	36.03(100.00)	36.03(100.00)	36.03
22	12.67(88.89)	12.67(88.89)	14.67(94.44)	14.00(88.89)	16.00(100.00)	14.67(88.89)	16.00(100.00)	16.00(100.00)	16.00(100.00)	16.00
23	0.93(81.82)	0.93(81.82)	2.57(81.82)	4.32(100.00)	4.32(100.00)	4.32(100.00)	4.32(100.00)	4.32(100.00)	4.32(100.00)	4.32
24	50.88(100.00)	50.88(100.00)	50.88(100.00)	50.88(100.00)	50.88(100.00)	50.88(100.00)	50.88(100.00)	50.88(100.00)	50.88(100.00)	50.88
25	19.87(100.00)	19.87(100.00)	19.13(47.37)	19.87(94.74)	19.78(89.47)	19.87(100.00)	19.87(100.00)	19.87(100.00)	19.87(94.74)	19.87
26	26.19(61.67)	26.19(61.67)	27.14(83.33)	28.57(48.33)	28.57(48.33)	28.57(48.33)	28.57(48.33)	31.90(80.00)	30.95(48.33)	33.81
27	6.38(100.00)	6.38(100.00)	6.38(100.00)	6.38(94.29)	6.38(100.00)	6.38(97.14)	6.38(100.00)	6.38(100.00)	6.09(97.14)	6.38
28	3.45(75.41)	3.45(75.41)	4.23(72.13)	4.67(98.36)	4.67(98.36)	4.73(100.00)	4.73(100.00)	4.73(100.00)	4.73(100.00)	4.73
29	49.29(33.33)	49.29(33.33)	49.76(50.00)	47.06(77.78)	48.35(77.78)	49.06(77.78)	48.94(61.11)	47.65(66.67)	47.18(77.78)	53.76
30	5.45(31.25)	5.45(31.25)	9.32(81.25)	10.00(100.00)	10.00(100.00)	10.00(100.00)	10.00(100.00)	10.00(100.00)	10.00(100.00)	10.00
31	36.36(100.00)	36.36(100.00)	32.02(76.92)	36.36(100.00)	36.36(100.00)	36.36(100.00)	36.36(100.00)	36.36(100.00)	36.36(100.00)	36.36
32	19.08(42.50)	19.08(42.50)	19.30(80.00)	18.02(95.00)	18.00(92.50)	18.76(97.50)	18.12(92.50)	19.30(100.00)	18.28(92.50)	19.30
33	5.55(76.47)	5.55(76.47)	5.64(82.35)	5.55(88.24)	5.55(88.24)	6.64(94.12)	5.55(94.12)	6.64(100.00)	5.55(94.12)	6.64

Tabela 3. Menores taxas de erros obtidas por cada um dos métodos de seleção de atributos. Os números entre parênteses indicam a porcentagem de atributos utilizada para construir o modelo em que aquela taxa de erro foi obtida.

## 5. Considerações finais

Neste trabalho foram investigados métodos de composição de *rankings* para combinar métodos de seleção de atributos que geram uma lista ordenada de atributos em um novo *ranking*. Para avaliar o desempenho dos métodos propostos o algoritmo Naïve Bayes foi utilizado para gerar modelos de classificação de uma maneira iterativa, em que a cada iteração um novo atributo era adicionado à lista de atributos utilizados para gerar o modelo de acordo com cada *ranking* gerado por cada método. Resultados experimentais em 33 conjuntos de dados da UCI mostraram a eficácia do método proposto. No geral, a taxa de acerto média dos métodos que usam agregação de *rankings* foram consideravelmente menores que a dos outros métodos individualmente. Além disso, os métodos baseados em agregação obtiveram, na maioria dos casos, a menor taxa de erro para os conjuntos de dados presentes no estudo usando as menores porcentagens de atributos do que todos os atributo dos conjuntos de dados.

## Agradecimentos

Este trabalho foi realizado com auxílio financeiro do CNPq. Parte deste trabalho foi desenvolvido quando o autor era pós-doutorando no ICMC/USP São Carlos.

## Referências

- Adali, S., Hill, B., and Magdon-Ismail, M. (2006). “The Impact of Ranker Quality on Rank Aggregation Algorithms: Information vs. Robustness”. In *II Workshop on Challenges In Web Information Retrieval and Integration*. IEEE Computer.
- Appice, A., Ceci, M., and Flach, S. R. P. A. (2004). “Redundant Feature Elimination for Multi-Class Problems”. In *Int. Conference on Machine Learning (ICML'2004)*.
- Asuncion, A. and Newman, D. (2007). UCI machine learning repository. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.
- Duch, W. (2006). Filter methods. In Guyon et al. (2006), chapter 3.
- Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001). “Rank aggregation methods for the Web”. In *International World Wide Web Conference*, pages 613–622.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.
- Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L., editors (2006). Springer.
- Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.
- Kononenko, I. (1994). “Estimating Attributes: Analysis and Extensions of RELIEF”. In *European Conf. on Machine Learning*, volume 784 of *LNCS*, pages 171–182. Springer.
- Lal, N., Chapelle, O., Weston, J., and Elisseeff, A. (2006). Embedded methods. In Guyon et al. (2006), chapter 5, pages 139–167.
- Prati, R. C. and Monard, M. C. (2006). “Combinando métodos de seleção de subconjuntos de atributos baseados na abordagem filtro”. In *I Workshop on Computational Intelligence (SBIA/SBRN/IBeramia'2006)*. publicado em CD-ROM.
- Vafaie, H. and Jong, K. D. (1993). “Robust Feature Selection Algorithms”. In *IEEE Int. Conf. on Tools with AI*, pages 356–363.
- Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition.