CLUSTERING MULTI-CRITÉRIO USANDO DISTÂNCIAS PONDERADAS DE TCHEBYCHEFF PARA DADOS RELACIONAIS

Sergio Queiroz
Centro de Informática – Universidade Federal de Pernambuco (CIn–UFPE)
Av. Jornalista Aníbal Fernandes s/n, Cidade Universitária, CEP 50740-560, Recife-PE
srmq@cin.ufpe.br

Francisco de Carvalho
Centro de Informática – Universidade Federal de Pernambuco (CIn–UFPE)
Av. Jornalista Aníbal Fernandes s/n, Cidade Universitária, CEP 50740-560, Recife-PE
fatc@cin.ufpe.br

Yves Lechevallier
Institut National de Recherche en Informatique et en Automatique (INRIA)
Domaine de Voluceau, Rocquencourt - B.P. 105, 78153 Le Chesnay, France
yves.lechevallier@inria.fr

RESUMO

Apresentamos um novo algoritmo capaz de particionar conjuntos de objetos levando-se em consideração simultaneamente suas descrições relacionais dadas por múltiplas matrizes de dissimilaridade. O algoritmo usa um critério de agregação não-linear, distâncias ponderadas de Tchebycheff, mais adequada do que combinações lineares (tais como médias ponderadas) para a construção de soluções de compromisso. São obtidos uma partição do conjunto de objetos, os protótipos de cada cluster e um vetor de pesos que indica a relevância de cada critério em cada cluster. Como se trata de um algoritmo de clustering para dados relacionais, o algoritmo é compatível com qualquer função de distância usada para medir a dissimilaridade entre os objetos. É mostrada uma aplicação prática, os resultados obtidos foram bastante coerentes com os dados utilizados.

PALAVARAS CHAVE. Análise de agrupamento, Dados Relacionais, Apoio à Decisão Multicritério.

ABSTRACT

We present a new algorithm that is capable of partitioning a set of objects taking into simultaneous consideration their descriptions given by multiple dissimilarity matrices. The algorithm uses a non-linear aggregation criterion, weighted Tchebycheff distances, more appropriate than linear combinations (such as weighted averages) for constructing compromise solutions. We obtain a partition of the set of objects, the prototypes of each cluster and a weight vector that indicates the relevance of each criterion in each cluster. As this is an cluster algorithm based on relational data, it is compatible with any distance function used for measuring the dissimilarity among objects. A practical application is shown, the obtained results were pretty coherent with the data we used.

KEYWORDS. Clustering Analysis, Relational Data, Multicriteria decision support.

1. Introdução

Métodos de agrupamento (ou *clustering*, do inglês) organizam um conjunto de observações (itens/objetos) em subconjuntos (chamados de *clusters*) de tal maneira que itens em um mesmo cluster são mais similares entre si do que itens em clusters diferentes. Tais métodos têm sido aplicados em áreas tão diversas quanto bioinformática, processamento de imagens, mineração de dados e recuperação de informação. Os tipos mais comuns de clustering são os métodos hierárquicos e os métodos de particionamento (Jain, Murty e Flynn, 1999; Xu e Wunsch, 2005).

Métodos hierárquicos produzem uma hierarquia completa, isto é, uma sequência aninhada de partições dos dados de entrada. Os métodos hierárquicos podem ser aglomerativos ou divisivos. No primeiro caso, a sequência de partições aninhadas inicia com o agrupamento trivial, em que cada item é um cluster unitário e termina com o agrupamento em que todos os itens pertencem ao mesmo cluster. Um método divisivo inicia com todos os itens em um cluster único e realiza operações de divisão até que um critério de parada seja atingido (normalmente chegando a uma partição de clusters unitários).

Métodos de particionamento buscam obter uma partição dos dados de entrada em um número fixo de clusters. Esses métodos normalmente buscam uma partição que otimiza (normalmente localmente) uma função objetivo. Para melhorar a qualidade do particionamento obtido, é comum que o algoritmo seja executado inúmeras vezes com diferentes condições iniciais, e a melhor configuração final obtida para as múltiplas execuções é utilizada como resultado do agrupamento. Métodos de partição dividem-se em métodos *hard* e métodos *fuzzy*. Nos métodos *hard* é fornecida uma partição em que cada item do conjunto de entrada é atribuído a exatamente um cluster. Nos métodos *fuzzy* é gerada uma partição em que é fornecido um grau de pertinência de cada item para cada um dos clusters da partição. Isso permite expressar que itens pertencem a mais de um cluster ao mesmo tempo.

Comumente há dois tipos de representação dos itens que podem ser usados para a realização da tarefa de agrupamento: *dados característicos* e *dados relacionais*. Quando cada item é descrito por um vetor de valores quantitativos ou qualitativos, ao conjunto de vetores que descreve os itens é dado o nome de *dados característicos*. Alternativamente, quando é descrita uma relação entre cada par de itens, o conjunto de relações é chamado de *dados relacionais*. O caso mais comum de dados relacionais é quando se tem (uma matriz de) dados de dissimilaridade, por exemplo $D=[d(i,l)]$, onde $d(i,l)$ é a dissimilaridade (normalmente uma distância) entre o par de itens *i* e *l*. Métodos capazes de realizar a tarefa de agrupamento a partir de dados relacionais são extremamente úteis, permitindo o agrupamento de itens que não podem ser descritos por dados característicos, ou quando a medida de distância entre itens não tem uma forma fechada. Por exemplo, se os itens a serem agrupados correspondem a relações de preferência de diferentes indivíduos, é possível agrupar os indivíduos de preferências similares apenas a partir das distâncias entre as preferências de cada par de indivíduos, medida a partir de uma medida de similaridade adequada, tal como a distância probabilística de Ha e Haddawy (2003), que pode mesmo ser usada sob incerteza ou para relações parciais.

Em diversas situações possuímos não apenas uma única medida de distância entre pares de itens, mas um vetor de distâncias para cada par. Isso pode ser decorrente de problemas *multi-critério*, onde cada distância é obtida segundo um critério diferente; de *decisão coletiva*, onde cada distância corresponde àquela segundo a opinião de um indivíduo diferente; ou sob *incerteza*, onde cada distância corresponde à realização de um cenário diferente. Assim, surge a necessidade de métodos capazes de realizar clustering segundo múltiplas matrizes de dissimilaridade, levadas em consideração simultaneamente. Dada a equivalência entre as três problemáticas, trataremos o problema como agrupamento multi-critério, sem perda de generalidade.

Dessa forma, nos últimos anos, algumas técnicas têm sido propostas na literatura para tratar o problema de agrupamento multi-critério. De Smet e Guzmán (2004) propuseram uma extensão ao popular algoritmo de agrupamento baseado em otimização de um critério local, *k-médias* (k-means), para o caso multi-critério, enquanto Fernandez, Navarro e Bernal, S. (2010) propuseram um novo algoritmo de agrupamento multi-critério utilizando heurísticas. Ambos os

métodos são baseados em funções de distância específicas definidas nos respectivos artigos, faltando-lhes a generalidade dos métodos de agrupamento baseados em dados relacionais, que podem usar qualquer medida de distância entre itens apropriada para o problema em questão.

Considerando métodos de agrupamento baseados em dados relacionais para o problema multi-critério, Frigui, Hwanga e Rhee (2007) propuseram o algoritmo CARD, baseado nos populares algoritmos para agrupamento fuzzy de dados relacionais NERF (Hathaway e Bezdek, 1994) e FANNY (Kaufman e Rousseeuw, 1990). Lechevallier, De Carvalho, Despeyroux e De Melo (2010) propuseram MRDCA, um algoritmo de agrupamento hard, que leva em consideração múltiplas matrizes de dissimilaridade, estendendo os algoritmos dinâmicos para clustering hard de Lechevallier (1974) e De Carvalho, Csernel e Lechevallier (2009). Tanto CARD quanto MRDCA são capazes de calcular um peso para cada matriz de dissimilaridade em cada um dos clusters, simultaneamente, o que potencialmente faz com que os clusters gerados sejam mais significativos (Frigui, Hwanga e Rhee, 2007). No entanto, ambos os métodos utilizam médias ponderadas como função de agregação, o que pode ser em várias situações inadequadas para encontrar soluções de compromisso.

Neste trabalho propomos um novo método de agrupamento hard para o caso multi-critério a partir de dados relacionais, levando-se em consideração múltiplas matrizes de dissimilaridade. O método proposto, denominado WRDCA, modifica o algoritmo MRDCA, utilizando distâncias ponderadas de Tchebycheff ao invés de médias ponderadas. Como é conhecido na literatura de multi-critério, a otimização de uma função baseada em distâncias ponderadas de Tchebycheff é mais adequada para a construção de soluções de compromisso do que médias ponderadas (Steuer e Choo, 1983; Wierzbicki, 1986). No entanto, por ser um critério não-linear, sua otimização quase sempre é não trivial. Neste trabalho, mostramos como o critério pode ser otimizado no nosso método de agrupamento através de programação linear.

O artigo é organizado da seguinte forma: na Seção 2 o problema tratado é descrito em maiores detalhes, bem como sua motivação; a Seção 3 descreve o nosso algoritmo, enquanto que na Seção 4 a aplicação do algoritmo é ilustrada através de um exemplo prático; finalmente, na Seção 5 são apresentados comentários finais.

2. Descrição do problema e motivação

Nessa seção são descritos os inconvenientes de se utilizar combinações lineares (por exemplo, médias ponderadas) como função de agregação na busca de soluções ótimas e o interesse de se usar agregadores não lineares, como distâncias ponderadas de Tchebycheff.

Tanto CARD quanto MRDCA buscam minimizar critérios de adequação para os clusters gerados que levam em consideração a média ponderada das dissimilaridades entre os elementos do cluster. Descreveremos a seguir o caso do algoritmo MRDCA por ser o algoritmo aqui proposto, o WRDCA, uma extensão dele.

Seja $E = \{e_1, \ldots, e_n\}$ um conjunto de n itens a serem particionados em K clusters e sejam p matrizes de dissimilaridade $n \times n$ $(D_1, \ldots, D_p)$, onde $D_j[i, l] = d_j(e_i, e_l)$ fornece a dissimilaridade entre os objetos e_i e e_l segundo a matriz de dissimilaridade D_j $\forall j = 1, \ldots, p$. Suponha que o protótipo g_k de cada cluster C_k é um elemento de E, isto é, $g_k \in E$ $\forall k = 1, \ldots, K$.

No MRDCA, é encontrada uma partição $P = (C_1, \ldots, C_K)$ de E em K clusters e o protótipo correspondente $g_k \in E$ para o cluster C_k em P tal que o critério de adequação J (função objetivo) é minimizado. O critério J mede a adequação entre os clusters e os seus respectivos protótipos, e é definido como:

$$J = \sum_{k=1}^{K} \sum_{e_i \in C_k} d^{(k)}(e_i, g_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_k^j d_j(e_i, g_k), \text{ onde} \quad (1)$$

$$d^{(k)}(e_i, g_k) = \sum_{j=1}^{p} \lambda_k^j d_j(e_i, g_k) \quad (2)$$

é a dissimilaridade entre um item $e_i \in C_k$ e o protótipo $g_k \in E$ parametrizado pelo vetor de pesos $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^p)$ onde λ_k^j é o peso da matriz de dissimilaridade D_j no cluster Ck, e $d_j(e_i, g_k)$

é a dissimilaridade segundo a matriz j entre o item $e_i \in C_k$ e o protótipo do cluster $g_k \in E \quad \forall j = 1, \ldots, p$. A matriz de relevância λ é composta por K vetores de pesos $\lambda_k = (\lambda_k^1, \ldots, \lambda_k^p)$, e muda a cada iteração (note que os pesos também podem ser diferentes de um cluster para outro), sendo calculada de forma a minimizar J (cf. Lechevallier, De Carvalho, Despeyroux e De Melo, 2010).

Dessa maneira, no MRDCA, a otimização é realizada minimizando-se uma função baseada na combinação linear das dissimilaridades entre os itens do cluster e o protótipo desse. A minimização de uma função que é uma combinação linear da avaliação dos objetos em relação a um conjunto de critérios pode limitar a possibilidade de encontrar soluções de compromisso interessantes. Por exemplo, na Figura 1 são ilustrados objetos avaliados segundo dois critérios a minimizar (eixos horizontal e vertical). Os retângulos representam os objetos não dominados (fronteira de Pareto). A otimização de uma combinação linear dos critérios (por exemplo, através de uma média ponderada da avaliação obtida nos dois critérios) levará sempre à escolha de uma solução no envelope convexo da fronteira (retângulos preenchidos), que nesse caso correspondem a soluções extremamente mal balanceadas (isto é, com uma divergência muito grande de avaliação segundo os dois critérios). Os objetos que seriam opções melhores quanto ao balanceamento entre os critérios (circulados na figura), ou seja, que representam um bom compromisso entre os critérios, estão na região côncava da fronteira, o que impossibilita que sejam escolhidos através da otimização de uma combinação linear.

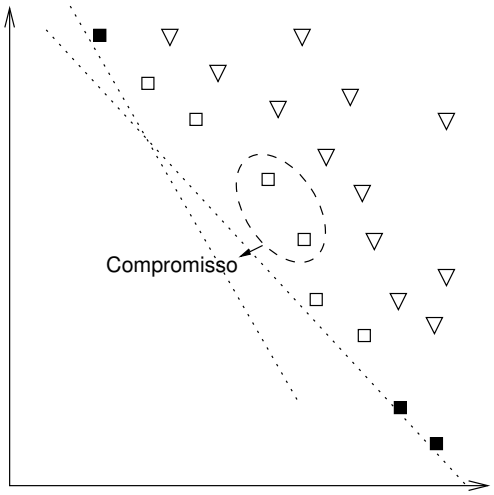

Figura 1. Escolha da melhor alternativa avaliada segundo dois critérios (a minimizar). Os retângulos correspondem a alternativas não-dominadas (isto é, na fronteira de Pareto). A otimização função que é uma uma combinação linear (ex.: média ponderada) dos critérios (linhas pontilhadas na figura) levará à escolha de soluções no envelope convexo (retângulos preenchidos), enquanto boas soluções de compromisso (circuladas na figura) não são atingidas por tal procedimento de otimização.

Quando se busca soluções de compromisso em relação a múltiplos critérios, um método mais flexível do que combinações lineares, utilizado na literatura de otimização multi-objetivo, é o critério de Tchebycheff (Steuer e Choo, 1983; Wiferzbicki, 1986). Para gerar soluções de compromisso segundo o critério de Tchebycheff nós minimizamos a função:

$$f_w(x) = \| w(\bar{u} - u(x)) \|_\infty = \max_{j \in \{1, \ldots, p\}} \{ w_j | \bar{u}^j - u^j(x) | \}$$

onde:

- $\bar{u} = (\bar{u}^1, \ldots, \bar{u}^p)$ representa um ponto ideal (normalmente fictício) que possui simultaneamente a melhor avaliação segundo todos os p critérios. Esse ponto ideal é um limite superior do conjunto de soluções não-dominadas no sentido de Pareto.
- w é um vetor de pesos (positivos).

Note que, ao minimizar a distância segundo a norma infinita (o max), o foco está no

critério em que a diferença da avaliação obtida pelo objeto em questão em relação à melhor avaliação obtida por um objeto é maior. Dessa forma, nós favorecemos os objetos próximos ao ponto de referência $\bar{u}$ em todas as dimensões do espaço de critérios. Isso nos permite encontrar boas soluções de compromisso. As funções $f_w(x)$ possuem duas propriedades importantes (Wierzbicki, 1986):

- **Propriedade 1:** Se $\forall\, j \in \{1, \ldots, p\}$, $w_j > 0$, então toda solução que minimiza $f_w(x)$ no conjunto de objetos *X* é fracamente Pareto-ótima. Além disso, pelo menos uma dessas soluções é Pareto-ótima.
- **Propriedade 2:** Se $\forall\, j \in \{1, \ldots, p\}$, $\bar{u}^j < \inf_{x \in X} u^i(x)$, então para toda solução Pareto-ótima *x*, existe um vetor de pesos *w* tal que *x* é a única solução que minimiza $f_w(x)$ em *X*.

A propriedade 1 nos mostra que a minimização de $f_w(x)$ obtém pelo menos uma solução Pareto-ótima. A propriedade 2 nos mostra que, ao contrário de funções de agregação lineares, toda solução Pareto-ótima pode ser atingida pela minimização de $f_w(x)$ utilizando-se um vetor de pesos apropriado. Isso explica porque a minimização de $f_w(x)$ é preferível a somas ponderadas no caso de otimização multiobjetivo em conjuntos não-convexos (Steuer e Choo, 1983; Wierzbicki, 1986). A Figura 2 mostra as distâncias para um ponto ideal segundo a norma infinita para o exemplo da Figura 1, utilizando o vetor de pesos identidade para *w*. Note que, ao modificarmos o vetor de pesos, nós deformamos o hipercubo de forma a alcançar outras regiões de interesse.

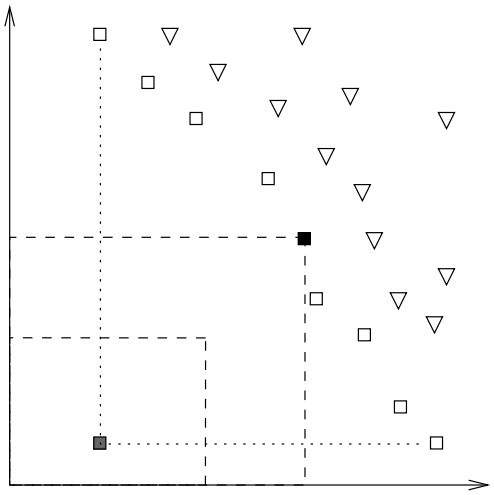

Figura 2: Distância segundo a norma infinita para um ponto ideal (fictício, no canto inferior esquerdo da figura), para o exemplo ilustrado na Figura 1 (utilizando o vetor de pesos identidade). O ponto ótimo encontrado (preenchido) é um dos identificado como “compromisso” na Figura 1.

3. Clustering multi-critério usando distâncias ponderadas de Tchebycheff para dados relacionais

Nessa seção, é apresentado o algoritmo WRDCA, uma extensão ao algoritmo MRDCA de agrupamento hard para dados relacionais com múltiplas matrizes de dissimilaridade (Lechevallier, De Carvalho, Despeyroux e De Melo, 2010), baseado em distâncias ponderadas de Tchebycheff.

Assim como utilizado na Seção 2, seja $E = \{e_1, \ldots, e_n\}$ um conjunto de *n* itens a serem particionados em *K* clusters e sejam *p* matrizes de dissimilaridade $n \times n$ ($D_1, \ldots, D_p$), onde $D_j[i, l] = d_j(e_i, e_l)$ fornece a dissimilaridade entre os objetos e_i e e_l segundo a matriz de dissimilaridade D_j $\forall j = 1, \ldots, p$. Suponha que o protótipo g_k de cada cluster C_k é um elemento de *E*, isto é, $g_k \in E$ $\forall k = 1, \ldots, K$.

No WRDCA, é encontrada uma partição $P = (C_1, \ldots, C_K)$ de *E* em *K* clusters e o protótipo correspondente $g_k \in E$ para o cluster C_k em *P* tal que o critério de adequação *J* (função

objetivo) é minimizado. O critério *J* mede a adequação entre os clusters e os seus respectivos protótipos, e é definido como:

$$J=\sum_{k=1}^{K}\sum_{e_i\in C_k} d^{(k)}(e_i,g_k)=\sum_{k=1}^{K}\sum_{e_i\in C_k}\max_{j=1}^{p}\ \lambda_k^j d_j(e_i,g_k)\text{ , onde} \tag{3}$$

$$d^{(k)}(e_i,g_k)=\max_{j=1}^{p}\ \lambda_k^j d_j(e_i,g_k) \tag{4}$$

é a dissimilaridade entre um item $e_i\in C_k$ e o protótipo $g_k\in E$ parametrizado pelo vetor de pesos $\lambda_k=(\lambda_k^1,\ldots,\lambda_k^p)$ onde λ_k^j é o peso da matriz de dissimilaridade D_j no cluster Ck, e $d_j(e_i,g_k)$ é a dissimilaridade segundo a matriz *j* entre o item $e_i\in C_k$ e o protótipo do cluster $g_k\in E$ $\forall j =$ 1, …, *p*. A matriz de relevância λ é composta por *K* vetores de pesos $\lambda_k=(\lambda_k^1,\ldots,\lambda_k^p)$, e muda a cada iteração (note que os pesos também podem ser diferentes de um cluster para outro). O algoritmo WRDCA alterna as três etapas seguintes, até que converge para um valor de J que corresponde a um mínimo local:

- **Etapa 1: Definição dos melhores protótipos**
  Nessa etapa, a partição $P = (C_1, \ldots, C_K)$ de *E* em *K* clusters e a matriz de relevância λ são mantidos fixados.
  Proposição 1. O protótipo $g_k=e_l\in E$ do cluster C_k que minimiza o critério *J* é computado de acordo com:

$$l=\arg\min_{h=1}^{n}\sum_{e_i\in C_k}\max_{j=1}^{p}\lambda_k^j d_j(e_i,e_h) \tag{5}$$

- **Etapa 2: Definição da melhor matriz de relevância**
  Nessa etapa, a partição $P = (C_1, \ldots, C_K)$ de *E* em *K* clusters e o vetor de protótipos $g=(g_1,\ldots,g_K)$ são mantidos fixados.
  Proposição 2. O vetor $\lambda_k=(\lambda_k^1,\ldots,\lambda_k^p)$ ótimo de cada cluster *k* pode ser calculado resolvendo-se o problema de programação linear min-max (mmLP):

$$\text{Minimizar}\ \sum_{e_i\in C_k}\max_{j=1}^{p}\ \lambda_k^j d_j(e_i,g_k)\ \text{, sujeito às restrições:}$$
$$0\le\lambda_k^j\le 1\quad\forall\ j=1,\ldots,p \tag{6}$$
$$\sum_{j=1}^{p}\lambda_k^j=1$$

Um problema de mmLP pode ser resolvido por *branch-and-bound* e como mostra Burks e Sakallah (1993), também pode ser transformado em um problema de programação linear inteira mista (MILP), permitindo a sua resolução por *solvers* de uso geral, o que é vantajoso dada a disponibilidade de implementações extremamente eficientes desses últimos. Para tal, basta notar que:

1. Uma restrição max de duas variáveis do tipo: $x_i=\max(x_j,x_k)$ pode ser substituída pelas restrições seguintes:

$$x_i\ge x_j,\ \ x_i\ge x_k \tag{7}$$

$$x_i-x_j\le c_i M,\ \ x_i-x_k\le(1-c_i)M \tag{8}$$

onde c_i é uma variável inteira 0-1 e *M* é uma constante positiva suficientemente grande.

2. O max de aridade *p* pode ser transformado em max binários, sabendo-se que:

$$\max_{j=1}^{p}\lambda_k^j d_j(e_i,g_k)=\max(\lambda_k^1 d_1(e_i,g_k),\mathrm{m}(2))\text{ , com} \tag{9}$$

$$\begin{cases} m(y) = \lambda_k^p d_p(e_i, g_k) & \text{se } y = p \\ m(y) = \max(\lambda_k^y d_y(e_i, g_k), m(y+1)) & \text{se } y \neq p \end{cases}$$

- **Etapa 3: Definição da melhor partição**
  Nessa etapa, o vetor de protótipos $g = (g_{1,}..., g_K)$ e a matriz de relevância λ são mantidos fixados.
  Proposição 3. A partição $P = (C_1, \ldots, C_K)$ que minimiza o critério J é atualizada de acordo com a seguinte regra de alocação:

$$C_k = \{e_i \in E : d^{(k)}(e_i, g_k) < d^{(h)}(e_i, g_h) \forall h \neq k\} \quad \forall k = 1,..., K \qquad (10)$$

se o mínimo não for único, e_i será alocado ao cluster com o menor índice.

É fácil demonstrar que cada etapa anteriormente descrita reduz o critério J. O algoritmo WRDCA é iniciado definindo-se uma partição inicial e executa as três etapas iterativamente, até a convergência, quando o critério $J(P, \lambda, g)$ alcança um valor estável, caracterizando um mínimo local. O algoritmo é resumido abaixo.

Algoritmo multi-critério de clustering hard para dados relacionais usando distâncias ponderadas de Tchebycheff

1. **Inicialização**
   Fixar o número K de clusters;
   Selecionar aleatoriamente K objetos distintos $g_k \in E$;
   Definir a matriz de relevância λ , onde $\lambda_k = (\lambda_k^1, ..., \lambda_k^p) = (1, ..., 1)$;
   Alocar cada objeto e_i ao protótipo mais próximo, a fim de obter a partição $P = (C_1, \ldots, C_K)$, onde cada C_k é construído conforme (10).
2. **Etapa 1: definição dos melhores protótipos.**
   A partição $P = (C_1, \ldots, C_K)$ e a matriz de relevância λ permanecem fixas.
   Computar os protótipos $g_k \in E$ de cada cluster C_k de acordo com a equação (5).
3. **Etapa 2: definição da melhor matriz de relevância**
   A partição $P = (C_1, \ldots, C_K)$ e o vetor de protótipos g permanecem fixos.
   Para cada k computar o vetor λ_k de acordo com o problema mmLP (6). Para resolvê-lo, transformamo-lo em um problema MILP usando(7), (8) e (9) e aplicamos um *solver*.
4. **Etapa 3: definição da melhor partição**
   O vetor de protótipos g e a matriz de relevância λ permanecem fixos.
   Construir a nova partição $P' = (C'_1, \ldots, C'_K)$ de acordo com (10) e verificar a convergência:
   convergência ← true;
   para $i = 1$ até n faça
   se e_i pertencia ao cluster C_m em P e agora pertence em P' ao cluster C'_k com $k \neq m$ então *convergência* ← *false;*
   $P \leftarrow P'$;
5. **Critério de parada**
   Se convergência = true então PARE. Se não, ir para 2. (Etapa 1).

4. Aplicação

Para ilustrar a utilidade prática do algoritmo proposto, nós o utilizaremos para agrupar categorias de população (itens) quanto à similaridade de opinião sobre um conjunto de 10 diferentes questões (critérios).

Em maio de 2008, o jornal estadunidense Los Angeles Times juntamente com a emissora de televisão KTLA realizaram uma pesquisa de opinião a fim de aferir a opinião dos adultos da Califórnia sobre questões ligadas ao casamento entre pessoas do mesmo sexo. Foram entrevistados 834 adultos. Os resultados foram divulgados não somente considerando o conjunto da população, mas também por subgrupos específicos, listados na Tabela 1. Para cada uma das 10

questões aqui consideradas (Tabela 2), as respostas de cada um dos 22 subgrupos foi divulgada, isto é, quantas pessoas do subgrupo escolheram cada uma das modalidades possíveis (Tabela 3) para a pergunta. Para medir a dissimilaridade relativa entre dois grupos e_i e e_j da população em relação a uma questão q, nós usamos o coeficiente de afinidade de Bacelar-Nicolay (2000), o que significa considerar os vetores de m valores (m é o número de modalidades da questão q) correspondente às frequências de cada uma das modalidades de q. A dissimilaridade de e_i e e_j segundo q é dada então por $d_q(e_i,e_j)=1-\sum_{l=1}^{m}\sqrt{\frac{f_l^q(e_i)}{f^q(e_i)}\times\frac{f_l^q(e_j)}{f^q(e_j)}}$, onde $f_l^q()$ é a frequência da modalidade l da questão q para o grupo, enquanto que $f_q()=\sum_{l=1}^{m} f_q^l()$.

O algoritmo proposto foi executado para obter partições com K = 1,..., 10. Para cada k foram feitas 100 execuções e o melhor resultado de acordo com o critério de adequação J foi escolhido. A tabela 4 mostra os resultados obtidos para K = 1, …, 5. É interessante observar cada um dos resultados, pois ele nos indica o que aconteceria se agrupássemos os dados em k grupos, quais subgrupos ficariam no mesmo cluster, e é possível observar que opiniões esses elementos têm mais em comum. Por exemplo, com um único cluster, vê-se maior concordância entre as respostas das questões Q8 e Q14, essas são justamente aquelas que pessoas com sentimentos diferentes sobre o tema podem responder da mesma forma. Outros dados interessantes de se observar: para todos os K = 1, …, 5, os grupos CONS, AF/REP e EVAN ficaram sempre juntos. Vimos que isso se deu porque eles são muitos similares quanto às respostas à questão Q18, e são consideravelmente diferentes dos outros subgrupos. Isso também era esperado dada à natureza do tema da pesquisa. Se usarmos o método de Da Silva (2009) para determinar o número mais adequado de *clusters*, que consiste em escolher os picos do gráfico das "diferenças de segunda ordem" do critério de adequação J: $J^{(K-1)}+J^{(K+1)}-2J^{(K)}$, $K=2,...,9$, obtemos que o número de clusters mais apropriado para o conjunto de dados é 5.

Tabela 1: Subgrupos da população

Nome do grupo	**Descrição**
ALL	Todos os entrevistados
REG	Entrevistados registrados para votar
18-34, 35-44, etc.	Grupos por idade
WHITE, NON/WHT	Brancos não-hispânicos, todos os outros incluindo Latino/Hispânico
MALE, FEMALE	Homens, Mulheres
EVAN, N/EVAN	Auto-denominado evangélico, não-evangélico
AF/DEM, AF/IND, AF/REP	Filiado aos Democratas, Independentes, e Republicanos
LIB, MOD, CONS	Auto-denominado liberal, moderado, conservador
<COL, DEG+	Sem formação universitária, superior completo ou mais
DK/GL, KN/GL	Não conhece qualquer gay ou lésbica, ou conhece alguém que é gay ou lésbica

Tabela 2. Questões consideradas

Id	**Pergunta**	**Tipo de resposta**
Q8	How closely have you been following the news about the California Supreme Court's decision on same-sex marriage? Have you been following it very closely, not too closely, or not closely at all?	A
Q9	As you may know, last week the California Supreme Court ruled that the California	B

	Constitution requires that same-sex couples be given the same right to marry that opposite-sex couples have. Based on what you know, do you approve or disapprove of the Court's decision to allow same-sex marriage in California? Do you strongly or only somewhat (approve/disapprove)	
Q10	As you may also know, a proposed amendment to the state's constitution may appear on the November ballot which would reverse the Supreme Court's decision and reinstate a ban on same-sex marriage. The amendment would state that marriage is only between a man and a woman. If the November election were held today, would you vote for or against the amendment to make marriage only between a man and a woman?	C
Q11	As you may know, Governor Schwarzenegger said he will respect the Supreme Court's decision on same-sex marriage. He also said he will not support a ballot measure to amend the constitution to define marriage as only between a man and a woman. Do you agree or disagree with the governor's decision to respect the Supreme Court ruling and not support a ballot initiative? Do you strongly or only somewhat (agree/disagree) with the governor's decision?	D
Q12	Do you think the debate about same-sex marriage is the most important issue facing the state, or an important issue but not the most important one, or is it not an important issue at all?	E
Q14	Regardless of your opinion about same-sex marriage, do you think legal recognition of it is inevitable, or not?	F
Q15	Do you personally believe that same-sex relationships between consenting adults are morally wrong or is that not a moral issue?	G
Q16	"As long as two people are in love and are committed to each other it doesn't matter if they are a same-sex couple or a heterosexual couple." Do you agree or disagree with this statement? Do you (agree/disagree) strongly or only somewhat?	D
Q17	"If gays are allowed to marry, the institution of marriage will be degraded." Do you agree or disagree with this statement? Do you (agree/disagree) strongly or only somewhat?	D
Q18	Do you have a friend, family member or co-worker who you know is gay or lesbian?	H

Tabela 3. Modalidades possíveis para cada tipo de resposta

Tipo de resposta	**Opções**
A	Very closely; Somewhat closely; Not too closely; Not closely at all; D/Know
B	Strongly approve; Somewhat approve; Somewhat disapprove; Strongly disapprove; D/Know
C	Vote yes; Lean yes; Lean no; Vote no; Wouldn't vote; D/Know
D	Strongly agree; Somewhat agree; Somewhat disagree; Strongly disagree; D/Know
E	Most important issue; Important, not most; Not important; D/Know
F	Yes, inevitable; No, not inevitable; Don't know; Refused
G	Morally wrong; Not a moral issue; Don't know
H	Don't know anyone; Know some one; Don't know

Tabela 4. Clusters formados (K = 1,..., 5)

K	Clusters	J	Weights
1	0: {ALL, 18-34, 35-44, 45-64, 65, WHITE, NON/WHT, MALE, FEMALE, EVAN, N/EVAN, REG, AF/DEM, AF/IND, AF/REP, LIB, CONS, MOD, DEG+, <COL KN/GL, DK/GL}	0.0416857	0: **(Q8, 0.1813894)** (Q9, 0.0683023) (Q10, 0.0852353) (Q11, 0.0994427) (Q12, 0.0616341)

			(Q14, 0.2396502) (Q15, 0.0621767) (Q16, 0.0852170) (Q17, 0.1109299) (Q18, 0.0060282)
2	0: {CONS, WHITE, EVAN, AF/IND, AF/REP} 1: {ALL, 18-34, 35-44, 45-64, 65, NON/WHT, MALE, FEMALE, N/EVAN, REG, AF/DEM, LIB, MOD, DEG+, <COL, KN/GL, DK/GL}	0.0183624	0: (Q8, 0.0333435) (Q9, 0.0111429) (Q10, 0.0078180) (Q11, 0.0137559) (Q12, 0.0367437) (Q14, 0.0421773) (Q15, 0.0126376) (Q16, 0.0252103) (Q17, 0.0160186) **(Q18, 0.8011522)** 1: **(Q8, 0.17455943)** (Q9, 0.06924683) (Q10, 0.0585095) **(Q11, 0.1520772)** (Q12, 0.0589263) **(Q14, 0.1645182)** **(Q15, 0.1588072)** (Q16, 0.06917268) (Q17, 0.090044394) (Q18, 0.0041383)
3	0: {MOD, DEG+, 35-44, KN/GL, AF/IND} 1: {18-34, 45-64, WHITE, FEMALE, EVAN, REG, AF/REP, LIB, CONS} 2: {ALL, <COL, 65, DK/GL, NON/WHT, MALE, N/EVAN, AF/DEM}	0.0183624	0: (Q8, 0.0126625) (Q9, 0.0158274) (Q10, 0.0210686) (Q11, 0.0145157) (Q12, 0.0177137) (Q14, 0.0547759) **(Q15, 0.7786444)** (Q16, 0.0544917) (Q17, 0.0178451) (Q18, 0.0124551) 1: (Q8, 0.0967453) (Q9, 0.0284488) (Q10, 0.0403842) (Q11, 0.0468929) (Q12, 0.0694556) (Q14, 0.0807277) (Q15, 0.0280232) (Q16, 0.0478342) (Q17, 0.0500132) **(Q18, 0.5114748)** 2: (Q8, 0.1684585) (Q9, 0.0607710) (Q10, 0.0233045) (Q11, 0.146762) (Q12, 0.2148528) (Q14, 0.1443811) (Q15, 0.1393691) (Q16, 0.0387845) (Q17, 0.0596847) (Q18, 0.0036318)
4	0: {ALL, 45-64, 65, WHITE, NON/WHT, MALE, N/EVAN, REG, AF/DEM, LIB} 1: {18-34, <COL, DK/GL, FEMALE} 2: {CONS, EVAN, AF/REP} 3: {MOD, DEG+, 35-44, KN/GL, AF/IND}	0.0127481	0: (Q8, 0.0850434) (Q9, 0.0446241) (Q10, 0.0295330) (Q11, 0.0772770) **(Q12, 0.2056236)** **(Q14, 0.1565712)** (Q15, 0.0921626) (Q16, 0.0640870) (Q17, 0.0986221) **(Q18, 0.1464561)** 1: (Q8, 0.0398146) (Q9, 0.0567014) (Q10, 0.0376267)

			(Q11, 0.0862839) (Q12, 0.0730960) **(Q14, 0.2176137)** **(Q15, 0.4468275)** (Q16, 0.0172722) (Q17, 0.0219152) (Q18, 0.0028487) 2: (Q8, 0.0303960) (Q9, 0.0312662) (Q10, 0.0071269) (Q11, 0.0148062) (Q12, 0.0637396) (Q14, 0.0500755) (Q15, 0.0115205) (Q16, 0.0461355) (Q17, 0.0146026) **(Q18, 0.7303312)** 3: (Q8, 0.0126625) (Q9, 0.0158274) (Q10, 0.0210686) (Q11, 0.0145157) (Q12, 0.0177137) (Q14, 0.0547758) **(Q15, 0.7786444)** (Q16, 0.0544917) (Q17, 0.0178451) (Q18, 0.0124551)
5	0: {DEG+, 35-44, KN/GL, LIB} 1: {ALL, 45-64, <COL, WHITE, NON/WHT, MALE, FEMALE, REG, AF/IND} 2: {MOD, 18-34, N/EVAN, AF/DEM} 3: {CONS, EVAN, AF/REP} 4: {65, DK/GL}	0.0102792	0: (Q8, 0.0487481) (Q9, 0.0392412) (Q10, 0.0655921) (Q11, 0.0558824) (Q12, 0.0681942) **(Q14, 0.2108756)** **(Q15, 0.2117056)** **(Q16, 0.1936396)** (Q17, 0.0941577) (Q18, 0.0119635) 1: (Q8, 0.0372563) (Q9, 0.0866998) (Q10, 0.0368745) (Q11, 0.0516218) **(Q12, 0.3891302)** (Q14, 0.0769031) (Q15, 0.0715126) (Q16, 0.0642533) (Q17, 0.0879141) (Q18, 0.0978342) 2: (Q8, 0.0114129) (Q9, 0.0058326) (Q10, 0.0129910) (Q11, 0.0123604) (Q12, 0.0051066) (Q14, 0.0359857) (Q15, 0.0167416) (Q16, 0.0057521) (Q17, 0.0052644) **(Q18, 0.8885527)** 3: (Q8, 0.0303960) (Q9, 0.0312662) (Q10, 0.0071269) (Q11, 0.0148062) (Q12, 0.0637396) (Q14, 0.0500755) (Q15, 0.0115205) (Q16, 0.0461355) (Q17, 0.0146026) **(Q18, 0.7303312)** 4: (Q8, 0.0488996) (Q9, 0.0744447)

			(Q10, 0.0275139) **(Q11, 0.4206682)** (Q12, 0.0708031) (Q14, 0.0776714) **(Q15, 0.1656099)** (Q16, 0.0268216) (Q17, 0.0850468) (Q18, 0.0025206)

5. Conclusões

Apresentamos um novo algoritmo capaz de particionar conjuntos de objetos levando-se em consideração simultaneamente suas descrições relacionais dadas por múltiplas matrizes de dissimilaridade. O algoritmo usa um critério de agregação não-linear, distâncias ponderadas de Tchebycheff, mais rico que combinações lineares (tais como médias ponderadas). Foi mostrada uma aplicação prática, onde foram consideradas 10 matrizes de dissimilaridade simultâneas (questões de uma pesquisa de opinião), e os resultados obtidos foram bastante coerentes com os dados utilizados.

Referências

Bacelar-Nicolay, H. (2000), The affinity Coefficient, In: Bock, H. H e Diday, E. (Eds.): *Analysis of Symbolic Data*, Springer, Heidelberg, 160-165.

Burks, T. M. e Sakallah, K. A. (1993), Min-max linear programming and the timing analysis of digital circuits, *Proceedings of the ICCAD'93*, 152-155.

Da Silva, A. (2009), Analyse de données évolutives: application aux données d'usage Web, *Thèse de Doctorat*, Université Paris-IX Dauphine.

De Carvalho, F. A. T., Csernel, M. e Lechevallier, Y. (2009), Clustering constrained symbolic data, *Pattern Recognition Letters* 30(11), 1037-1045.

De Smet, Y. e Guzmán, L. M. (2004), Towards multicriteria clustering: An extension of the k-means algorithm, *European Journal of Operational Research*, 158, 390-398.

Fernandez, E., Navarro, J. e Bernal, S. (2010), Handling multicriteria preferences in cluster analysis, *European Journal of Operational Research*, 202(3), 819-827.

Frigui, H., Hwanga, C. e Rhee, F. C.-H. (2007), Clustering and aggregation of relational data with applications to image database categorization, *Pattern Recognition*, 40(11), 3053-3068.

Ha, V. e Haddawy, P. (2003), Similarity of personal preferences: Theoretical foundations and empirical analysis, *Artificial Intelligence* 146, 149-173.

Hathaway, R. J., Bezdek, J. C. (1994), Nerf c-means: non-Euclidean relational fuzzy clustering, *Pattern Recognition* 27 (3), 429-437.

Jain, A. K., Murty, M. N. e Flynn, P. J. (1999), Data Clustering: A Review, *ACM Computing Surveys*, 31(3), 264–323 .

Kaufman, L. e Rousseeuw, P. J., *Finding Groups in Data*, Wiley, New York, 1990.

Lechevallier, Y., Optimisation de quelques critères en classification automatique et application a l'étude des modifications des protéines sériques en pathologie clinique, *Thèse de 3ème cycle*, Université Paris-VI, 1974.

Lechevallier, Y., De Carvalho, F. A. T., Despeyroux, T. e De Melo, F. M. (2010), Clustering of Multiple Dissimilarity Data Tables for Documents Categorization, *Proceedings of COMPSTAT'2010 19th International Conference on Computational Statistics*, 1263-1270 .

Los Angeles Times e KTLA, California Same Sex Marriage Issues Survey, *Field Poll*, May 20-21, 2008. Disponível online em http://latimes.com/timespoll

Steuer, R. e Choo, E.-U. (1983), An interactive weighted Tchebycheff procedure for multiple objective programming, *Math. Prog.* 26, 326–344.

Wierzbicki, A. (1986), On the completeness and constructiveness of parametric characterizations to vector optimization problems, *OR Spektrum*, 8:73–87.

Xu, R. e Wunsch, D. (2005), Survey of Clustering Algorithms, *IEEE Transactions on Neural Networks* 16(3), 645–678 .