Concepção e implementação de algoritmos baseados em espaços de formulações para o agrupamento de dados

Lucas Lyon de Azevedo
Èverton Santi

24 de Maio de 2018

Resumo

Este trabalho apresenta um algoritmo baseado em espaços de formulações como método alternativo para o Problema de *Clustering* Heterogêno (PCH). O Problema de *Clustering* heterogêneo considera eliminar o processo de agregação para os casos em que múltiplas matrizes de dissimilaridades são consideradas no processo de agrupamento. O *Formulation Space Search* (FSS) leva em consideração que problemas de otimização podem ter mais de uma formulação, ou seja, uma solução pode ser um ponto estacionário em uma formulação, mas pode vir a não ser em outra. Os resultados obtidos por meio do algoritmo proposto são comparados com o estado da arte. A análise destes resultados sugere que o algoritmo é competitivo, por ter um grande diferencial de tempo.
PALAVRAS CHAVE : *Clustering*. Espaços de Formulações. Otimização.

Abstract

This paper presents an algorithm based on formulation spaces as an alternative method for the Heterogeneous Clustering Problem. The Heterogeneous Clustering Problem (PCH) considers the cases where multiple matrices of dissimilarities are considered in the clustering process. The Formulation space search (FSS) takes into account that optimization problems may have

more than one formulation, a solution may be a stationary point in a formulation, but may not be in other. The results obtained by means of the proposed algorithm are compared with the state of the art. The analysis of these results suggests that the algorithm is competitive because it has a great time differential.
Keywords : Clustering. Formulation space . Optimization.

1 Introdução

O Problema de *Clustering* Heterogêneo (PCH) foi proposto por (SANTI et al., 2016) como uma alternativa aos modelos tradicionais de *clustering*, no intuito de reduzir a perda de informação gerada pelo procedimento de agregação. A eliminação deste processo intermediário consiste no seu maior diferencial com relação aos métodos tradicionais (ex.: *k-means,single-linkage,* etc). O problema tem como objetivo identificar G segmentos, grupos de indivíduos, cuja as matrizes de dissimilaridades representam uma solução similar em relação ao agrupamento de um conjunto de objetos. Sendo assim, para cada segmento identificado pelo PCH, gera-se uma partição com base nos n objetos. O modelo proposto por (SANTI et al., 2016) é baseado no conceito de medianas, isto implica que cada objeto é relacionado ao item que melhor representa os elementos de seu *cluster* (SANTI et al., 2016) .

De maneira a formular o PCH, considera-se que m indivíduos devem julgar n objetos, afim de se obter uma matriz de dissimilaridades $D^k = (d^k_{ij})$ para todo $k \in \{1, ..., m\}$ e para todo $i, j \in \{1, ..., n\}$, e c^k, para $k \in \{1, ..., m\}$, o número de clusters esperado por cada individuo k em relação a estes objetos.

A partir das definições apresentadas, a formulação do PCH é apresentad no ANEXO A, com base neste tem-se que (1) minimiza a soma das dissimilaridades para cada associação de objetos i e sua respectiva mediana j, condicionada à pertinência de cada indivıduo k em relação aos G grupos. O conjunto de restrições definido em (2) impõe que cada objeto i seja associado a exatamente uma mediana j dentro de um grupo g. As restrições em (3) garantem que um objeto i somente será associado ao outro objeto j em um grupo g se o objeto j for uma mediana dentro daquele grupo. As restrições em (4) impõem que um indivıduo k deve ser designado a exatamente um grupo g, enquanto que as restrições em (5) garantem que não haverá grupos vazios.

As restrições em (6) impõem que o número total de medianas (clusters ou categorias) para cada grupo g deverá ser igual ao piso do número médio de medianas esperado pelos indivıduo alocados a este grupo. Por fim, as restrições em (7) e (8) são restrições de integralidade sobre os valores das

variáveis de decisão do problema.

O PCH foi resolvido por meio de uma metaheurística *Variable Neighborhood Search* (VNS) em (SANTI et al., 2016) , e uma alternativa para resolver o PCH, é o *Formulation space search* (FSS) que foi primeiramente introduzido por (Mladenovic et al., 2005) em *"Reformulation descent applied to circle packing problems"*, e em seguida foi extendido para o Problemas de Programação Não-Linear Inteira Mista (MINPL) por (López e Beasley, 2014) . O FSS leva em consideração que problema de otimização podem ter mais de uma formulação, e que uma solução pode ser um ponto estacionário em uma formulação, mas pode vir a não ser em outra. Pensando nisso o FSS se baseia em alternar entre formulações para escapar justamente desses ótimos locais.

Sendo o FSS uma nova ideia na literatura, este trabalho tem como objetivo propor um novo algoritmo para o PCH baseado no conceito do FSS.

O restante deste texto está organizado como segue: na Seção 2 os métodos e algoritmo proposto são apresentado, na Seção 3 os resultados são demontrados, na Seção 4 está a conclusão do artigo e na Seção 5 os anexos.

2 Método

Todos os modelos e algoritmos FSS considerados neste trabalho foram implementados na linguagem AMPL (A mathematical programming language), escolhiida por ser de fácil manipulação matemática. Primeiramente, testou-se os algoritmo FSS em (López e Beasley, 2014) e em (López e Beasley, 2016) afim de resolver o PCH, não foi utilizado o método proposto em (Mladenovic et al., 2005) pois este é para problemas continuos, entretanto o PCH é um problema discreto.

Para os testes foi utilizado um computador com um processador Intel(R) Xeon(R) CPU X5650 2.67GHz, uma memória de 6GB e com o sistema CentOS Linux 7.

Dado que os algoritmos FSS considerados neste trabalho fazem uso de resolvedores, utilizou-se o SNOPT e o CPLEX, pois estes apresentaram maior eficiência quando comparados a outros produtos disponíveis. O SNOPT foi utilizado para resolver problemas não-lineares e o CPLEX para resolver problemas lineares inteiros mistos. A partir da implementação e teste dos algoritmos de López e Beasley (2014,2016), bem como da percepção de sua ineficiência em resolver o PCH, formulou-se um novo algoritmo, o qual é apresentado em sequência.

2.1 Algoritmo Proposto

De maneira a apresentar o algoritmo proposto, consideram-se duas formulações: P e P*. P corresponde à formulação original do problema que ser quer resolver, i.e. o PCH. P* é uma formulação alternativa, obtida a partir da modificação de P, pela remoção das restrições de integralidade sobre as variáveis de decisão, bem como a partir da adição da restrição exibida no ANEXO B.

A formulação P* pode ser resolvida rapidamente por meio de um resolvedor não-linear, gerando uma solução ótimo local. A partir desta solução, deve-se obter uma solução viável para o problema original P. A formulação P*, ainda, pode ser perturbada facilmente pela modificação do valor de seu parâmetro δ, o que faz com que o resolvedor gere diferentes soluções para diferentes valores deste parâmetro.

Logo, o algoritmo proposto se baseia em obter uma solução inicial a partir de P*, para então então otimizá-la e torná-la viável para P. O ANEXO C apresenta o pseudocódigo deste novo algoritmo.

3 Resultados e Discussões

Nessa seção analisa-se os resultados obtidos por meio do algoritmo proposto. Primeiramente compara-se os custos da função objetivo dados pelo VNS em (SANTI et al., 2016) aos custos reportados pelo FSS aqui proposto. Para tal , os experimentos computacionais aqui reportados foram realizados a partir das instâncias geradas por meio de um simulação de monte carlo descrita com mais detalhes em (Blanchard, 2012) e representada em ANEXO E. Para estas, a estrutura de grupos e categorias é conhecida (valores das variáveis (z e e)).

Dentre os fatores considerados para as instâncias estão o número total de individuos m, o número total de grupos G, o número total de objetos n, o número total de medianas definidos para os grupos, erros aleatórios introduzidos nas matrizes de distâncias (utilizando-se $N(0, 0.5)$ ou $N(0, 0.1)$) e erros aleatórios introduzidos no número de medianas definido por cada indivíduo (utilizando-se $N(0, 0.5)$ ou $N(0, 0.1)$).

Como critério de parada foi utilizado 10 iterações sem melhorar o custo da solução igualmente realizado por (SANTI et al., 2016) . Para o ANEXO D, a primeira coluna representa as instâncias, a segunda coluna apresenta o limitante inferior (*lower bound*) sendo este o melhor valor obtido em (SANTI et al., 2016) , a terceira coluna representa o menor custo obtido para o VNS, a quarta coluna representa o tempo demandado para se obter tal custo, a

quinta coluna representa o GAP do VNS com base na literatura, a sexta coluna representa o melhor custo obtido pelo FSS, a sétima coluna mostra o tempo demando pelo algoritmo para resolver a instância e a oitava coluna representa o GAP do FSS.

Segundo ANEXO D o custo da solução encontrada pelo FSS é pior em 19 casos, igual em 7 e melhor em 1, 3 soluções ótimas foram obtidas, a diferença de custos entre o VNS e o FSS é em média de 3%, o Tempo de execução do FSS corresponde, em média, a 6% do tempo demandado pelo VNS.

Embora o algoritmo é pior em 19 casos ele obteve um tempo de resolução mais eficiênte que o VNS, as 3 soluções ótimas foram as mesmas encontradas pelo VNS, mas com tempo menor de processamento. Como critério de parada foi utilizado o número de iterações para ambos os casos, com isso o FSS aqui proposto demanda menos tempo para atingir o critério de parada.

4 Conclusão

Utilizando o *Formulation space search* (FSS) para resolver o Problema de *Clustering* Heterogêneo (PCH) foi necessário realizar algumas modificações no algoritmo do (López e Beasley, 2016) , surgindo assim um novo algoritmo. Os resolvedores SNOPT(*Sparse Nonlinear OPTimizer*) e o CPLEX representaram uma boa performace para esse algoritmo que consiste em dividir o problema em não linear e linear. Os resultados do FSS mostraram-se competitivos ao *Variable Neighborhood Search* (VNS) proposto em (SANTI et al., 2016) , visto que foi atingido o mesmo número de soluções ótimas e tempo de execução relativamente menor com custo computacional relativamente baixo, tornando-se assim de viável aplicação.

Como trabalho futuro pode-se testar este por um período de tempo maior, e indo além, pode-se testar novos resolvedores, para comparar suas aplicações e também poderia acrescentar ou modificar as restrições do problema para maior análise de experimentos.

Referências

[1] Simon J. Blanchard, Daniel Aloise, and Wayne S. Desarbo. The heterogeneous p-median problem for categorization based clustering. 2012.

[2] Santi E., Aloise D., and Blanchard S. J. A model for clustering data from heterogeneous dissimilarities. 2016.

[3] Philip E. GILL, Walter MURRAY, and Michael A. SAUNDERS. User's

guide for snopt version 7: Software for large-scale nonlinear programming. 2008.

[4] C. O. López and J. E. Beasley. A note on solving minlp's using formulation space search. 2013.

[5] C. O. López and J. E. Beasley. A formulation space search heuristic for packing unequal circles in a fixed size circular container. 2016.

[6] Nenad Mladenovic, Frank Plastria, and Dragan Urosevic. Reformulation descent applied to circle packing problems. 2005.

[7] David M. Gay Robert Fourer and Brian W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* AMPL, 2003.

ANEXO A

$$\min \sum_{k=1}^{m} \sum_{g=1}^{G} z^{kg} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^{k} e_{ij}^{g} \tag{1}$$

Sujeito a:

$$\sum_{j=1}^{n} e_{ij}^{g} = 1 \quad \forall g \in \{1, ..., G\}, \quad \forall i \in \{1, ..., n\} \tag{2}$$

$$e_{ij}^{g} \leq e_{jj}^{g} \quad \forall g \in \{1, ..., G\}, \quad \forall i, j \in \{1, ..., n\} \tag{3}$$

$$\sum_{g=1}^{G} z^{kg} = 1 \quad \forall k \in \{1, ..., m\} \tag{4}$$

$$\sum_{k=1}^{m} z^{kg} \geq 1 \quad \forall g \in \{1, ..., G\} \tag{5}$$

$$\sum_{j=1}^{n} e_{jj}^{g} = \left\lfloor \frac{\sum_{k=1}^{m} c^{k} z^{kg}}{\sum_{k=1}^{m} z^{kg}} \right\rfloor \quad \forall g \in \{1, ..., G\} \tag{6}$$

$$e_{ij}^{g} \in \{0, 1\} \quad \forall g \in \{1, ..., G\}, \forall i, j \in \{1, ..., n\} \tag{7}$$

$$z^{kg} \in \{0, 1\} \quad \forall g \in \{1, ..., G\}, \forall k \in \{1, ..., m\} \tag{8}$$

ANEXO B

$$\sum_{g=1}^{G}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}e_{ij}^{g}(1-e_{ij}^{g})+\sum_{k=1}^{m}z^{kg}(1-z^{kg})\right]\leq\delta \tag{9}$$

ANEXO C

$z_{best} \leftarrow \infty \quad \delta \leftarrow 0.5 \quad \beta \leftarrow 0.9$
while *Condição de parada não atingida* **do**

Resolva P^* utilizando algum resolvedor não linear

Arredonde os valores das variáveis e para o inteiro mais próximo

do

Resolva o problema P considerando os valores das variáveis e como fixos, atribuindo o custo da solução a z_1

Resolva o problema P considerando os valores das variáveis z como fixos, atribuindo o custo da solução a z_2

while $z_1 \neq z_2$

Atualize z_{best} caso uma solução viável e de melhor custo tenha sido encontrada

Atualize δ como $\delta \leftarrow \beta\delta$

end

Algorithm 1: Algoritmo proposto

ANEXO D

Tabela 1: Resultados

Instância	**Lim. Inf.**	**VNS**			**FSS**		
		Lim. Sup.	Tempo	GAP	Lim. Sup.	Tempo	GAP
1	2707.04	3207.20	1801.30	15.59%	3231.59	92.2874	16.23%
2	2311.75	2388.25	35.97	**3.20%**	2388.25	3.56501	**3.20%**
3	4364.75	4604.21	105.18	**5.20%**	4890.82	7.82572	10.76%
4	1819.56	1869.92	49.49	**2.69%**	1869.92	2.63937	**2.69%**
5	3346.33	3778.12	1801.58	**11.43%**	3933.69	108.526	14.93%
6	1382.45	1525.20	471.19	**9.36%**	1559.84	15.9733	11.37%
7	2410.68	2663.61	46.09	**9.50%**	2665.01	3.35425	9.54%
8	1470.84	1601.38	719.66	**8.15%**	1644.83	23.0191	10.58%
9	6646.22	7384.69	1801.48	**10.00%**	7466.09	102.813	10.98%
10	4773.51	5690.08	1603.06	**16.11%**	5748.96	49.8868	16.97%
11	2516.03	2835.41	304.74	**11.26%**	2835.41	41.4119	**11.26%**
12	3749.99	3749.99	482.42	0.00%	3749.99	20.9112	0.00%
13	3067.9	3562.48	1055.33	**13.88%**	3562.48	43.2626	**13.88%**
14	2905.84	3076.92	1015.23	**5.56%**	3303.5	33.802	12.04%
15	5642.2	5808.60	205.80	**2.86%**	6162.91	16.8136	8.45%
16	9668.71	10120.40	342.92	**4.46%**	10479	13.974	7.73%
17	2669.76	3131.17	552.36	**14.74%**	3232.62	22.5261	17.41%
18	5993.6	6497.31	871.74	**7.75%**	6636.35	50.0204	9.69%
19	1190.01	1206.01	134.36	**1.33%**	1300.67	6.41806	8.51%
20	10935	10935.00	536.99	0.00%	10935	101.799	0.00%
21	3405.42	3593.04	144.62	**5.22%**	3670.46	11.8822	7.22%
22	4231.32	4610.67	149.36	**8.23%**	5171.03	5.02738	18.17%
23	5206.48	5682.85	1236.88	**8.38%**	6259.09	28.4584	16.82%
24	3167.1	3745.28	1801.15	15.44%	3827.35	56.8481	17.25%
25	1142.62	1256.43	319.60	**9.06%**	1431.1	9.47049	20.16%
26	1874.99	1874.99	19.30	0.00%	1874.99	1.84232	0.00%
27	8657.6	8983.98	1804.38	3.63%	8969.4	175.533	**3.48%**

ANEXO E

Tabela 2: Simulação de Monte Carlo

Instância	**Individuos** m	**Grupos** G	**Objetos** n	**Medianas**	**Pertubação** dissimilaridades	**Pertubação** medianas
1	150	10	30	50 percent 3, 50 percent 6	N(0, 0.1)	N(0, 0.5)
2	300	2	18	All 6	N(0, 0.1)	0
3	450	2	18	50 percent 3, 50 percent 6	N(0, 0.05)	0
4	150	2	18	All 3	N(0, 0.05)	N(0, 0.5)
5	450	10	18	All 6	N(0, 0.05)	N(0, 1)
6	150	10	18	50 percent 3, 50 percent 6	N(0, 0.05)	0
7	300	2	18	All 6	0	N(0, 0.5)
8	150	10	18	50 percent 3, 50 percent 6	0	N(0, 1)
9	300	10	30	All 3	N(0, 0.05)	N(0, 0.5)
10	450	6	18	All 3	N(0, 0.1)	N(0, 1)
11	150	6	30	All 6	N(0, 0.1)	0
12	300	10	18	All 3	0	0
13	450	10	18	All 6	N(0, 0.1)	0
14	300	6	18	50 percent 3, 50 percent 6	0	N(0, 1)
15	300	2	30	All 6	N(0, 0.05)	N(0, 1)
16	450	2	30	50 percent 3, 50 percent 6	0	N(0, 1)
17	300	6	18	50 percent 3, 50 percent 6	N(0, 0.1)	N(0, 0.5)
18	300	6	30	50 percent 3, 50 percent 6	N(0, 0.05)	0
19	150	6	18	All 6	0	N(0, 0.5)
20	450	6	30	All 3	0	0
21	150	2	30	All 3	N(0, 0.1)	N(0, 1)
22	450	2	18	50 percent 3, 50 percent 6	N(0, 0.1)	N(0, 0.5)
23	450	6	18	All 3	N(0, 0.05)	N(0, 0.5)
24	300	10	18	All 3	N(0, 0.1)	N(0, 1)
25	150	6	18	All 6	N(0, 0.05)	N(0, 1)
26	150	2	18	All 3	0	0
27	450	10	30	All 6	0	N(0, 0.5)

Fonte: Blanchard[2012]