Revisão de Estratégias para a Aceleração do Algoritmo *k-Means*

Marcelo Kuchar Matte & Maria do Carmo Nicoletti

Faculdade de Campo Limpo Paulista - FACCAMP
Campo Limpo Paulista - SP, Brasil

{marcelokuchar@gmail.com,carmo@cc.faccamp.br}

Abstract. *The k-Means is a clustering algorithm with a long history of success in a wide range of applications from many different research areas. Such success is due, in particular, to its simplicity, which helps to produce a quick implementation as well as to the good results it provides. Despite success, however, the original k-Means has some shortcomings. One of them concerns the processing time required for the algorithm to end the iterative process it is based upon. This article presents a brief review of some strategies proposed in the literature with the purpose of accelerating the k-Means processing time.*

Resumo. *O k-Means é um algoritmo de agrupamento com uma longa história de sucesso em um vasto número de aplicações, nos mais variados domínios de conhecimento. Tal sucesso se deve, particularmente, à sua simplicidade, à facilidade com que pode ser implementado e aos bons resultados que fornece. Apesar do sucesso, entretanto, o k-Means original apresenta algumas deficiências. Uma delas diz respeito ao tempo de processamento necessário para a finalização do processo iterativo que implementa. Esse artigo apresenta uma breve revisão de algumas estratégias que foram propostas na literatura com o objetivo de acelerar o processamento do algoritmo.*

1. Considerações Iniciais

O algoritmo *k-Means* é reconhecidamente um dos algoritmos de Aprendizado de Máquina (AM) mais populares, dentre os algoritmos caracterizados como algoritmos de agrupamento. Como relatado em [Wikipedia 2019] o termo *k-Means* foi empregado pela primeira vez em 1967, por James MacQueen, em [MacQueen 1967], muito embora o procedimento descrito pelo algoritmo esteja ligado a Hugo Steinhaus desde 1956 [Steinhaus 1957]. O *k-Means* padrão, também referenciado como standard ou como básico, foi proposto em 1957 por Lloyd [Lloyd 1957], como uma técnica para a ondulação *pulse-code* e não foi publicado fora do Latoratório Bell até 1982, quando então foi publicado em [Lloyd 1982]. Em 1965 E. W. Forgy publicou essencialmente o mesmo método em [Forgy 1965], razão pela qual o algoritmo é também referenciado em algumas publicações como algoritmo Lloyd-Forgy. Uma versão mais eficiente foi proposta e publicada em Fortran, por Hartigan & Wong, em [Hartigan & Wong 1979]. Desde a sua proposta o *k-Means*, além de ter sido usado em um grande volume de aplicações nos mais variados domínios de conhecimento, tem também sofrido inúmeras críticas, com relação a algumas de suas características. Particularmente muitas tentativas de melhoramento de sua fase de inicialização foram publicadas, como aquelas descritas em [Bradley & Fayyad 1998] [Maedeh & Suresh 2013] [Kahn & Ahmad 2004], bem como podem ser evidenciados inúmeros trabalhos que propõem estratégias com o objetivo de acelerar o tempo de execução do algoritmo.

Entre alguns aspectos que eventualmente contribuem para o *k-Means* ter um processamento não muito rápido estão (a) um número alto de instâncias a serem agrupadas; (b) o número alto de cálculos de distâncias entre instâncias e centroides que o algoritmo deve realizar e (c) a necessidade do algoritmo requerer muitas iterações para convergir. Como caracterizado em [Hamerly & Drake 2015], os métodos básicos de aceleração de algoritmos de aprendizado de máquina podem ser categorizados como: (1) melhoramentos algorítmicos; (2) paralelização (incluindo *threading*, multiprocesamento e computação distribuída) e (3) aproximação. Particularmente, o trabalho de pesquisa iniciado contempla a categoria (1) e tem por objetivo investigar o impacto do uso da desigualdade triangular com vistas a acelerar o algoritmo *k-Means*, como sugerido em [Elkan 2003]. A Seção 2 apresenta o pseudocódigo do *k-Means* seguida por uma breve descrição do funcionamento do algoritmo. A Seção 3 apresenta a revisão de alguns trabalhos que tem por foco propostas de estratégias para a aceleração do algoritmo *k-Means* e Seção 4 finaliza o artigo informando como o trabalho já realizado terá continuidade.

2. O Algoritmo *k-Means*

O pseudocódigo do *k-Means* mostrado na Figura 1 foi composto com base nas descrições do algoritmo encontradas em [Witten *et al.* 2011] e [Han *et al.* 2012]. O algoritmo espera como entrada (a) um conjunto contendo N instâncias de dados $I = \{I_1, I_2, ..., I_N\}$, em que cada instância I_i, $1 \leq i \leq N$, é descrita por valores associados a M atributos A_j, $1 \leq j \leq M$, bem como (b) um valor para o parâmetro k, que representa o número de grupos que o agrupamento a ser induzido deve ter. O algoritmo fornece, como saída, um agrupamento, que no pseudocódigo mostrado na Figura 1 é notado por $AG = \{G_1,G_2,...G_k\}$.

```
procedure k-Means(I,k,AG)
Input: I = {I1, I2, ..., IN}        %conjunto com N instâncias de dados a serem agrupadas
       k                             % número de grupos a serem criados
Output: AG = {G1,G2,...Gk}           %agrupamento formado por k grupos induzidos a partir de I
begin
% Inicialização
%  no passo (1) cada grupo é definido apenas pelo centroide
(1) escolha arbitraria de k instâncias do conjunto I, como centroides dos grupos G1,G2,...Gk

% Indução do agrupamento AG
(2) repeat
   (3) (re)atribuir cada instância Ii ∈ I (i=1, ..., N) ao grupo cujo centroide que lhe
       seja mais próximo;
   (4) atualizar os centroides de cada grupo, como a média dos valores das suas instâncias
(5) until nenhuma alteração aconteça.
end.
return AG = {G1,G2,...Gk}
end_procedure
```

Figura 1. Pseudocódigo em alto nível do *k-Means*.

Na fase de inicialização o *k-Means* padrão escolhe randomicamente k instâncias de I, e elege cada uma delas como centroide (representativo) para cada um dos k grupos. O agrupamento ao final da fase de inicialização é representado por um conjunto com k elementos, em que cada elemento é um conjunto que tem por elemento apenas o centroide. Na fase iterativa do algoritmo, indicada pelo comentário *% Indução do Agrupamento AG*, na Figura 1, cada uma das instâncias restantes de I é então atribuída ao grupo cujo respectivo centroide lhe seja mais próximo, por meio do cálculo da distância de

cada instância, a cada um dos k centroides considerados; via de regra a distância euclidiana é usada.Na sequência, a média dos valores de atributos que representam as instâncias que participam de cada um dos k grupos é calculada e os k centroides são atualizados. Todo o processo é então repetido, com os novos centroides de grupos, até que o processo atinja estabilidade, caracterizada como a situação em que as mesmas instâncias são atribuídas aos grupos aos quais já pertencem, em iterações consecutivas.

3. Estratégias para Aceleração do *k-Means*

Muitos pesquisadores têm investido no desenvolvimento de estratégias com vistas à aceleração do processamento realizado por algoritmos de agrupamento em geral. Como o projeto de pesquisa associado a este artigo tem por foco o algoritmo *k-Means*, o que segue é uma revisão bibliográfica de alguns dos algoritmos encontrados na literatura, propostos com o objetivo exclusivo de acelerar o processamento do *k-Means* e que podem ser caracterizados como melhoramentos algorítmicos. Hamerly em [Hamerly 2010] comenta que quando as instâncias a serem agrupadas têm alta dimensionalidade, esquemas de indexação, como aquele de árvores k-d, não funcionam bem, a ponto do processo do exame de cada instância (*i.e.*, sem o uso de uma estrutura que favoreça a aceleração) ser bem mais rápido do que processos realizados por algoritmos que implementam aceleração para baixa dimensionalidade. Hamerly e Drake em [Hamerly & Drake 2015] comentam que para conjuntos de instâncias descritas com um número pequeno de atributos (*i.e.*, com baixa dimensionalidade horizontal), a indexação das instâncias a serem agrupadas é uma maneira efetiva de acelerar o *k-Means*. Apesar dos trabalhos descritos em [Kanungo *et al.* 2002] e [Pelleg & Moore 1999], terem sido feitos por pessoas distintas em épocas distintas, ambos são similares na maneira como propõem a adaptação de uma árvore k-d padrão para promover uma aceleração do *k-Means*.

Outro aspecto importante a ser considerado, quando do uso de algoritmos que usam árvores k-d como estrutura para armazenamento das instâncias e de informações complementares, diz respeito aos custos envolvidos na construção e uso de tal estrutura. Para um conjunto com N instâncias de dados, o custo da construção de uma árvore k-d é da ordem de O(Nlog(N)) e aproximadamente duplica o custo de memória necessária. Também, se o conjunto de instâncias sofre mudanças, a atualização da árvore k-d para refletir essas mudanças não é um processo trivial com baixo custo computacional. Como evidenciado nos experimentos descritos em [Pelleg & Moore 2000], o método conhecido como Algoritmo Blacklisting é efetivo para a indução de agrupamentos em conjuntos com um número elevado de instâncias (dimensionalidade vertical alta). Os autores também comentam que o algoritmo se torna lento quando a dimensionalidade horizontal das instâncias a serem agrupadas se torna maior do que 8. Em conjuntos de instâncias com alta dimensionalidade, as instâncias e centroides tendem a ficar longe uns dos outros e a possibilidade de uso de procedimentos de poda acaba sendo reduzida. Em [Moore 2000] o autor descreve uma proposta de estrutura hierárquica, a hierarquia de âncoras, adequada para lidar com instâncias de dados com alta dimensionalidade, que satisfaz a desigualdade triangular. Similarmente ao uso de árvores k-d, a construção e manutenção dessa estrutura hierárquica podem ser complexas e via de regra demandam um investimento computacional alto, tanto em tempo quanto em memória.

Alguns trabalhos, ao invés de lidar diretamente com agrupamentos em conjuntos de instâncias com alta dimensionalidade, lidam com uma projeção desses conjuntos em um espaço com baixa dimensionalidade, e usam algoritmos adequados para agrupamen-

to de instâncias com baixa dimensionalidade. O método para redução de dimensionalidade conhecido como PCA (*Principal Component Analysis*) é abordado em associação ao aprendizado não supervisionado realizado pelo *k-Means*, no trabalho descrito em [Ding & He 2004]. Os resultados obtidos no trabalho indicam que os processos de redução de dimensionalidade não supervisionada e de aprendizado não supervisionado estão fortemente relacionados.

Considerando que o efeito de uma inicialização adequada dos centroides promove a indução mais rápida de um agrupamento, métodos de inicialização de centroides podem, de certa forma, ser abordados como algoritmos de aceleração *do k-Means*. Em uma situação em que o número de grupos do agrupamento a ser induzido é k, na fase de inicialização do *k-Means* original (Figura 1), k instâncias, randomicamente escolhidas, constituem o conjunto inicial dos k centroides. Se, ao invés de uma escolha randômica, um algoritmo for utilizado para uma escolha apropriada dos centroides iniciais, a indução do agrupamento final pode eventualmente ser acelerada. Esse é o caso do algoritmo *Furthest-First* [Hochbaum & Shmoys 1985], como o algoritmo a ser empregado na fase de inicialização do *k-Means*. O *Furthest-First* começa escolhendo randomicamente uma instância como o primeiro centroide e, então, repetidamente seleciona, como o próximo centroide, a instância que está mais longe de qualquer centroide já escolhido. Tal algoritmo, entretanto, apesar de fácil e de ser rapidamente implementado, tem a tendência de escolher *outliers*, como parte do conjunto inicial de centroides, dada a tendência de *outliers* estarem situados próximos às fronteira da massa de dados considerados. Bradley e Fayyad em [Bradley & Fayyad 1998] apresentam um procedimento para calcular um refinamento do processo de inicialização de centroides, que permite que o algoritmo iterativo convirja para um melhor mínimo local. O procedimento em questão pode ser usado agregado a um grande número algoritmos de agrupamento, tanto para dados discretos quanto contínuos. Os resultados obtidos e discutidos no trabalho evidenciam que o conjunto inicial de centroides, quando refinado pelo procedimento proposto e usado com o *k-Means*, provoca um melhoramento nos resultados do *k-Means*. Um estudo empírico sobre a contribuição de estratégias de inicialização do *k-Means* com vistas à diminuição do número de iterações do algoritmo está apresentada em [Oliveira & Nicoletti 2018].

A referência [Phillips 2002] descreve duas estratégias relativamente simples para acelerar o processamento do algoritmo *k-Means*. O uso de tais estratégias não modifica o resultado obtido pelo algoritmo ou seja, o *k-Means* com ou sem o uso de qualquer das duas estratégias, usando entretanto os mesmos centroides iniciais, sempre induz o mesmo agrupamento. Como comentado em [Elkan 2003], o uso da propriedade conhecida como desigualdade triangular quando da implementação do *k-Means* padrão é o de acelerar o tempo de processamento do algoritmo. O emprego da desigualdade triangular permite que muitos dos cálculos realizados pelo *k-Means* padrão, possam ser evitados, o que acelera o processo de indução executado pelo algoritmo. A proposta nomeada neste texto como *k-Means-Elkan*, contempla o uso da propriedade de duas maneiras distintas, subsidiadas por dois resultados teóricos, bem como de um monitoramento dos limites superiores e inferiores das distâncias entre instâncias e centroides de grupos.

O algoritmo proposto por Hamerly em [Hamerly 2010] é considerado por seu autor não apenas como uma modificação do *k-Means-Elkan* mas, também, como uma simplificação do *k-Means-Elkan*. Assim como o *k-Means-Elkan*, o *k-Means-Hamerly* usa limites para as distâncias, que são eficientemente atualizados, assim como usa a

desigualdade triangular, para evitar cálculos de distâncias entre instâncias e centroides. O *k-Means-Hamerly* emprega dois limites de distância, por instância de dados, para seus dois centroides mais próximos. Um deles é um limite superior na distância da instância ao seu centroide mais próximo e o outro é um limite inferior na distância ao segundo centroide mais próximo. Com base nos resultados obtidos dos experimentos descritos em [Hamerly 2010], o autor comenta que o *k-Means-Hamerly* teve melhor desempenho em conjuntos de dados com dimensionalidade pequena e moderada, enquanto que o *k-Means-Elkan* teve melhor desempenho em dados com alta dimensionalidade. De uma certa maneira esses dois algoritmos podem ser abordados como complementares um do outro, sendo a escolha de um deles sempre dependente da dimensionalidade das instâncias de dados a serem agrupadas. O algoritmo *Yinyang k-Means* [Ding e*t al.* 2015] também usa a desigualdade triangular para acelerar o algoritmo k-Means, de maneira similar àquela utilizada no algoritmo proposto por Elkan. No *Yinyang* k-Means os limites superiores e inferiores são utilizados como filtros para detectar cálculos de distância desnecessários, o que acelera as etapas de atribuição e de atualização dos centroides. De acordo com os autores, o uso desses filtros gera um considerável ganho de performance, superando consistentemente a performance do k-Means, permitindo ser até 3 vezes mais rápido que as principais otimizações conhecidas.

4. Considerações Finais

Este artigo apresenta uma breve revisão bibliográfica de vários algoritmos/estratégias encontrados na literatura técnica, com foco em algoritmos de agrupamento e, particularmente, o algoritmo *k-Means*, que têm como objetivo acelerar o processamento do algoritmo para a indução do agrupamento final. O levantamento feito não foi exaustivo; buscou-se entretanto, identificar as propostas mais relevantes com o intuito de providenciar um contexto e identificar as mais promissoras para uma investigação mais detalhada. O trabalho já realizado, que resultou na revisão apresentada neste artigo, está sendo continuado com foco no uso da desigualdade triangular com vistas à aceleração do processamento do *k-Means*.

Referências

[Bradley & Fayyad 1998] Bradley, P. S.; Fayyad, U. (1998) Refining initial points for k-means clustering, in: Proc. of the 15th International Conference on Machine Learning, pp. 91–99.

[Ding & He 2004] Ding, C.; He, X. (2004) K-means clustering via principal component analysis, In: Proc. of the 21st Int. Conference on Machine Learning, Banff, Canada, pp.255–232.

[Ding *et al.* 2015] Ding, Y.; Zhao, Y.; Shen, X.; Musuvathi, M.; Mytkowicz, T. (2015) Yinyang k-means: A drop-in replacement of the classic k-means with consistent speedup, In: Proc. of the Int. Conference on Machine Learning, pp. 579–587.

[Elkan 2003] Elkan, C. (2003) Using the triangle inequality to accelerate k-Means, In: Proc. of the Twentieth International Conference on Machine Learning (ICML-2003), pp. 147–153.

[Forgy 1965] Forgy, E. W. (1965) Cluster analysis of multivariate data: efficiency versus interpretability of classifications, *Biometrics*, v. 21, no. 3, pp. 768–769.

[Hamerly 2010] Hamerly, G. (2010) Making k-Means even faster, In: Proc. of the SIAM International Conference on Data Mining, pp. 130–140.

[Hamerly & Drake 2015] Hamerly, G.; Drake, J. (2015) Accelerating Lloyd's algorithm for k-means clustering, in: *Partitional Clustering Algorithms*, Springer-Verlag, pp. 41–78. doi:10.1007/978-3-319-09259-1_2.

[Han *et al.* 2012] Han, J.; Kamber, M.; Pei, J. (2012) *Data Mining Concepts and Techniques*, 3rd. Ed., Amsterdam: Morgan Kaufmann Publishers.

[Hartigan & Wong 1979] Hartigan, J. A.; Wong, M. A. (1979) Algorithm AS 136: A k-Means Clustering Algorithm, *J. of the Royal Statistical Society*, Series C, v. 28, no.1, pp. 100–108.

[Hochbaum & Shmoys 1985] Hochbaum, D. S.; Shmoys, D. B. (1985) A best possible heuristics for the k-center problem, *Mathematics of Operations Res.*, v. 10, no. 2, pp. 180–184.

[Kanungo *et al.* 2002] Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silveman, R.; Wu, A. Y. (2002) An efficient k-Means clustering algorithm: analysis and implementation, *IEEE Transactions on Pattern Analysis and Machine Intel.*, v. 24, no. 7, pp. 881–892.

[Khan & Ahmad 2004] Khan, S. S.; Ahmad, A. (2004) Cluster center initialization algorithm for k-Means clustering, *Pattern Recognition Letters*, v. 25, pp. 1293–1302.

[Lloyd 1957] Lloyd, S. P. (1957) Least square quantization in PCM, *Bell Telephone Laboratories Paper*.

[Lloyd 1982] Lloyd, S. P. (1982), Least squares quantization in PCM, *IEEE Transactions on Information Theory*, v. 28, no. 2, pp. 129–137

[Maedeh & Suresh 2013] Maedeh, A.; Suresh, K. (2013) Design of efficient k-means clustering algorithm with improved initial centroids, *MR International Journal of Engineering and Technology*, v. 5, no. 1, pp. 33–37.

[MacQueen 1967] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations, In: Proc. of The 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, vol. 1, pp. 281–297.

[Moore 2000] Moore, A. W. (2000) The anchors hierarchy: using the triangle inequality to survive high dimensional data, In: Proc. of The Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI'00), pp. 397–405.

[Oliveira & Nicoletti 2018] Oliveira, A. F.; Nicoletti, M. C. (2018) Favoring the k-Means algorithm with initialization methods, In: Proc. of The International Conference on Intelligent Systems Design and Applications (ISDA 2018), pp. 21–31.

[Pelleg & Moore 1999] Pelleg, D.; Moore, A. (1999) Accelerating exact k-Means algorithms with geometric reasoning, In: Proc. of The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 277–281.

[Pelleg & Moore 2000] Pelleg, D.; Moore, A. (2000) X-means: extending K-means with efficient estimation of the number of clusters, In: Proceedings of the 17th International Conf. on Machine Learning, pp. 727–734.

[Phillips 2002] Phillips, S. J. (2002) Acceleration of K-Means and related clustering algorithms, In: Mount D. M., Stein C. (eds) Algorithm Engineering and Experiments (ALENEX 2002), *Lecture Notes in Computer Science*, v. 2409, Berlin:Springer-Verlag, pp 166–177.

[Steinhaus 1957] Steinhaus, H. (1957) Sur la division des corps matériels en parties, *Bull. Acad. Polon. Sci.* (em francês), v. 4, no. 12, pp. 801–804.

[Wikipedia 2019] Wikipedia contributors (2019) K-means_clustering, *Wikipedia, The Free Encyclopedia*, 01 Jul. 2019. https://en.wikipedia.org/wiki/K-means_clustering

[Witten *at al.* 2011] Witten, I. H.; Frank E.; Hall, M. A. (2011) *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd. Ed., Amsterdam: Morgan Kaufmann Publishers.