# Impacto do Uso da Desigualdade Triangular para Acelerar o Algoritmo k-Means

**Maria do Carmo Nicoletti**

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

{carmo@cc.faccamp.br}

***Abstract.*** *Clustering is one of the many ways to implement Machine Learning (ML), particularly in situations where the available training set has no information about the classes of the data instances. Among the many clustering algorithms, the k-Means stands up due, mainly, to the fact that usually works well and is easy to be implemented. The algorithm, however, is slow when used with a large volume of data. This paper investigates the use of the triangular inequality as a way to turn the algorithm faster.*

***Resumo.*** *Agrupamento é uma das muitas maneiras de implementar Aprendizado de Máquina (AM), particularmente em situações em que o conjunto de instâncias de dados disponibilizado não dispõe da informação sobre a classe das instâncias. Dentre os muitos algoritmos de agrupamento, o k-Means se destaca devido, principalmente, à sua simplicidade, facilidade de ser implementado e dos bons resultados que usualmente obtém. O algoritmo, entretanto, ainda é lento na prática, quando usado em grandes volumes de dados. Esse artigo investiga o uso de desigualdade triangular como uma maneira de tornar o algoritmo mais rápido.*

## 1. Introdução e Contextualização

Via de regra cada um, dentre os inúmeros algoritmos caracterizados como de Aprendizado de Máquina (AM), pertence uma das três subáreas de AM identificadas, respectivamente, como: (1) aprendizado supervisionado, (2) aprendizado não-supervisionado e (3) aprendizado semi-supervisionado. Cada uma dessas subáreas tem características bem definidas e agrega um conjunto de algoritmos apropriados (ver [Mitchell 1997] [Duda *et al*. 2001 [Witten *et al.* 2011] [Han *et al.* 2012] para um panorama geral das muitas diversificações dentro dessas áreas).

De particular interesse neste trabalho de pesquisa é a área de aprendizado não-supervisionado e, dentre as técnicas mais utilizadas em aprendizado não-supervisionado, a conhecida como agrupamento (*clustering*) é uma das mais populares e uma das que têm tido maior sucesso em aplicações do mundo real (ver [Theodoridis & Koutroumbas 1999] e [Duda *et al.*, 2001] para detalhes).

Dentre os muitos algoritmos de agrupamento disponíveis na literatura, aquele conhecido como k-Means [MacQueen 1967] [Lloyd 1982] é o principal objeto de estudo e investigação do projeto de pesquisa sendo conduzido e parcialmente descrito neste artigo, com vistas à investigação de uma técnica particular, para diminuir o tempo computacional de processamento desse algoritmo.

O k-Means, de uma maneira simplificada e objetiva, pode ser descrito como um procedimento que, dado um conjunto com N instâncias de dados (via de regra descritas como vetores de M valores, cada um deles associado a um atributo de um conjunto com M atributos), busca particionar o conjunto dado em k grupos, em que k é um parâmetro geralmente fornecido pelo usuário. Cada instância de dado vai pertencer ao grupo cujo protótipo (geralmente definido como a média das instâncias de dados do grupo) lhe for mais próxima. Esse procedimento induz o particionamento do espaço de dados no que é conhecido como diagramas de Voronoi [Reddy & Jana 2012].

O k-Means tem sido objeto de pesquisa e de tentativas de melhoramento de várias de suas características desde quando foi criado, fato que pode ser evidenciado por meio dos inúmeros trabalhos que investem nesse assunto, particularmente aqueles de descrevem propostas de algoritmos para a fase de inicialização do k-Means, como em [Bradley & Fayyad 1998] [Khan & Ahmad 2004] [Maedeh & Suresh 2013]), na promoção de sua escalabilidade, como em [Farnstron *et al.* 2000], para seu uso em dados com alta dimensionalidade [Sun & Wang 2012] e outros que investem em sua customização para uso em determinados tipos de aplicações, como aqueles descritos em [Montolio *et al.* 1992] e [Nieddu *et al.* 2011].

Particularmente, o trabalho de pesquisa em andamento investiga 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 k-Means com mais detalhes e a Seção 3 introduz a motivação para o uso de desigualdade triangular [Elkan 2003], como recurso para acelerar o processamento computacional do k-Means. A Seção 4 finaliza o artigo apresentando os próximos passos pretendidos para a continuação do trabalho e a metodologia cogitada.

## 2. Uma Breve Descrição do Algoritmo k-Means

A descrição e o pseudocódigo do k-Means apresentados nessa seção foram baseados nas referências [Witten *et al.* 2011] e [Han *et al.* 2012]. No que segue o conjunto de N instâncias de dados a serem agrupadas é, de uma maneira geral, referenciado como $CI = \{I_1, I_2, ..., I_N\}$, e cada uma das instâncias $I_i$, $1 \leq i \leq N$, é descrita por valores associados a M atributos $A_j$, $1 \leq j \leq M$. A Figura 1 apresenta um pseudocódigo simplificado do algoritmo k-Means, inspirado naquele encontrado em [Han *et al.* 2012].

Como entrada ao algoritmo k-Means são fornecidos o conjunto de instâncias CI, bem como um valor para o parâmetro k, que indica o número desejado de grupos que o agrupamento, a ser gerado pelo algoritmo, deve ter. Na sua fase de inicialização, que acontece uma única vez logo ao princípio de sua execução, o algoritmo k-Means padrão escolhe randomicamente k instâncias de CI, como os k centroides de grupos. As demais instâncias de CI são então atribuídas ao grupo (inicialmente contendo apenas o centroide), cujo respectivo centroide lhe seja mais próximo, por meio do cálculo da distância (euclidiana, geralmente) de cada instância, a cada um dos k centroides considerados.

Na sequência, a média dos valores de atributos que representam as instâncias que participam de cada grupo (isto é, os respectivos centroides de grupos) é calculada, os centroides são atualizados e todo o processo é repetido, com os novos centroides de grupos. O processo iterativo continua até atingir sua estabilidade que pode ser traduzida como a situação em que as mesmas instâncias de dados são atribuídas aos grupos aos quais já pertencem, em iterações consecutivas.

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

% fase de indução do agrupamento
 (2) repeat
 (3)  (re)atribuir cada instância Ii ∈ CI ao grupo cujo centroide que lhe seja mais próximo;
 (4)  atualizar os centroides de cada grupo, como a média os 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.**

## 3. Da Conveniência do Uso da Desigualdade Triangular

O objetivo do uso da propriedade conhecida como desigualdade triangular, como comenta o autor da proposta em [Elkan 2003], é o de acelerar o k-Means padrão. O uso de tal propriedade vai permitir que muitos dos cálculos de distância, realizados pelo k-Means padrão, possam ser evitados,contribuindo, dessa forma, para acelerar o processo de indução do agrupamento desejado. A proposta 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.

Considerando que o número de instâncias de dados a serem agrupadas é N, que k seja o número de grupos a serem criados e que *e* representa o número de iterações necessárias para o algoritmo convergir, a complexidade em tempo do k-Means padrão é O(Nk*e*). Empiricamente, *e* cresce sublinearmente com k, N e a dimensionalidade M das instâncias de dados.

O número de cálculos de distância realizados pelo k-Means é dado pelo produto Nk*e*. A principal contribuição no uso da propriedade da desigualdade triangular para acelerar o k_Means padrão, como apontado em [Elkan 2003], está na diminuição do número de cálculos de distâncias que, na prática, passa a estar mais perto de N do que de NK*e*. Entretanto, é preciso que o algoritmo acelerado, k-Means_AC, satisfaça três propriedades:

(1) deve ser capaz de começar o processamento a partir de um grupo arbitrário de k centroides (de maneira que todos os métodos de inicialização possam continuar a ser usados);
(2) se um mesmo conjunto inicial de centroides for utilizado, o k-Means_AC deve sempre induzir os mesmos centroides finais, como acontece com o k-Means padrão e
(3) deve ser capaz de usar qualquer métrica de distância (*i.e.*, não deve se restringir à otimização específica, por exemplo, da distância euclidiana). A condição (3), particularmente, é importante uma vez que muitas aplicações

usam uma métrica de distância específica ao domínio de dados da aplicação. Considere a notação e definições que seguem.

Seja X um conjunto. Uma métrica em X é uma função (chamada *função distância* ou apenas *distância*) definida como d: $X \times X \rightarrow [0, \infty)$ e para todo x, y e $z \in X$, as seguintes propriedades são satisfeitas:

(1) $d(x,y) \geq 0$;
(2) $d(x, y) = 0 \leftrightarrow x = y$;
(3) $d(x, y) = d(y,x)$;
(4) $d(x,z) \leq d(x,y) + d(y,z)$ (*desigualdade triangular*).

Uma métrica é chamada *ultramétrica* se para todo x, y e $z \in X$ satisfizer uma versão mais restrita da desigualdade triangular expressa por: $d(x,z) \leq \max\{d(x,y), d(y,z)\}$ (*i.e.*, elementos de X nunca podem estar 'entre' outros elementos de X).

Abordando o problema de maneira simplista, considere que x seja uma instância e b e c sejam centroides; é preciso garantir que $d(x,c) \geq d(x,b)$ a fim de evitar ter que, efetivamente, calcular o valor de d(x,c).

## 4. Metodologia de Trabalho e Continuidade da Pesquisa

A continuidade do projeto prevê: (1) estudo e entendimento em detalhes dos resultados teóricos, bem como suas provas, para o uso da desigualdade triangular com o objetivo pretendido *i.e.*, acelerar a execução do processo de indução de um agrupamento pelo k-Means padrão, a partir de um conjunto de instâncias de dados; (2) levantamento e estudo de outras propostas, além daquela descrita em [Elkan 2003], que utilizam a desigualdade triangular com o mesmo propósito; (3) implementação do k_Means padrão e de variantes do k-Means que empregam a desigualdade triangular, como módulos de um sistema computacional para experimentação com algoritmos de AM; (4) definição de conjuntos de instâncias dados que reflitam situações usuais bem como situações limites. Serão identificados conjuntos de instâncias de dados que já tenham sido utilizados em outros trabalhos similares, para evidenciar as contribuições das propostas levantadas, cujos resultados estejam disponíveis em publicações; (5) seleção de índices de validação para viabilizar uma comparação entre os resultados obtidos pelos algoritmos investigados e implementados, em um conjunto de experimentos de agrupamentos, utilizando os conjuntos escolhidos em (4).

### 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.

[Duda *et al.* 2001] Duda, R. O.; Hart, P. F.; Stork, D. G. (2001) Pattern Classification, USA: John Wiley & Sons, Inc.

[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.

[Farnstron *et al.* 2000] Farnstrom, F.; Lewis, J.; Elkan, C. (2000) Scalability for clustering algorithms revisited, ACM SIGKDD Explorations, v. 2, pp. 51–57.

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

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

[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 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.

[Montolio *et al.* 1992] Motolio, P.; Gasull, P.; Monte, A.; Torres, L.; Marques, F. (1992) Analysis and optimization of the k-Means algorithm for remote sensing applications, In: A. Sanfeliu (Ed.), Pattern Recognition and Image Analysis, World Scientific, pp. 155–170.

[Nieddu *et al.* 2011] Nieddu, L.; Manfredi, G.; D'Acunto, S. (2011) A fully automatic K-means-based algorithm for image segmentation, International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2011), pp. 32–37.

[Reddy & Jana 2012] Reddy, D.; Jana, P. K. (2012) Initialization for K-means clustering using Voronoi diagram, Procedia Technology, vol. 4, pp. 395–400.

[Sun & Wang 2012] Sun, W.; Wang, J. (2012) Regularized k-Means clustering of high-dimensional data and its asymptotic consistency, Electronic Journal of Statistics , v. 6, pp. 148–167.

[Theodoridis & Koutroumbas 1999] Theodoridis, S.; Koutroumbas, K. (1999) Pattern Recognition, USA: Academic Press.

[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.