

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Algoritmo ckMeans Aplicado ao Sensoriamento Remoto

Rogério R. de Vargas¹

Cristiano Galafassi²

Laboratório de Sistemas Inteligentes e Modelagem, Unipampa, Itaqui, RS

Natália Carvalho de Amorim³

Engenharia de Agrimensura, Unipampa, Itaqui, RS

Ricardo Freddo⁴

Bacharelado Interdisciplinar em Ciência e Tecnologia, Unipampa, Itaqui, RS

Resumo. O sensoriamento remoto é uma tecnologia para obter imagens e dados da superfície terrestre através da captação e registro da energia refletida/emitida pela superfície sem que haja contato físico entre um sensor e a superfície analisada. Este trabalho propõe o desenvolvimento de um método que adapta e ajusta a entrada e saída de dados chamado de pré e pós-processamento de imagens para aplicar o algoritmo ckMeans no sensoriamento remoto. Para validar a proposta aplicou-se o pré e pós-processamento em uma imagem para executar o algoritmo ckMeans, onde foi possível distinguir as áreas da imagem geradas pelo algoritmo, validando o processo de segmentação de imagens aplicado ao sensoriamento remoto.

Palavras-chave. Agrupamento, ckMeans, Sensoriamento Remoto.

1 Introdução

Segundo [1], o sensoriamento remoto compreende um conjunto de atividades que torna possível a extração de informações de objetos que compõem a superfície terrestre eliminando a necessidade de contato direto com os mesmos. Seu principal objetivo é extrair informações e codificá-las em tabelas, gráficos ou mapas, facilitando assim a leitura destas informações [2]. Os mesmos autores destacam em seu trabalho que a extração das informações de sensoriamento remoto demanda métodos e análises claras e lógicas para que possam ser aplicados a qualquer produto, possibilitando a análise de diferentes dados por diferentes profissionais.

Uma das metodologias utilizadas para análise e interpretação de imagens de satélite é a classificação. O processo de classificação de imagens multiespectrais pode ser definido como o processo de associar cada pixel da imagem a uma quantidade finita de classes, sendo que estas classes são interpretadas como elementos que representam o mundo real [2]. Assim, o

¹rogvar@icloud.com

²cristianogalafassi@gmail.com

³carvalho.natalia03@gmail.com

⁴ricardo.freddo@hotmail.com

produto da classificação é um mapa digital temático, o qual será criteriosamente analisado pelo usuário no processo de extração de informações.

Os algoritmos de classificação mais utilizados pelos profissionais da área são o K-Means e o ISODATA [3], os quais trabalham através da lógica booleana. Além disso, outros autores ainda defendem que algoritmos baseados na lógica fuzzy são ferramentas altamente eficazes, uma vez que são capazes de representar as incertezas que surgem em técnicas de segmentação ou classificação [4].

Desse modo, em [6] aplicou-se o algoritmo ckMeans [5] no processamento de imagens médicas, onde resultados preliminares foram discutidos. Entretanto, estudos referentes ao pré e pós-processamento de imagens não foram detalhados.

Neste contexto, este trabalho aplica o algoritmo fuzzy ckMeans a uma imagem multiespectral com o objetivo de avaliar os mapas temáticos digitais gerados, bem como ressaltar a potencialidade do método no campo do sensoriamento remoto.

Em trabalhos como [7], [8] e [9] os autores trabalham com o algoritmo Fuzzy C-Means aplicado ao sensoriamento remoto. Pelo algoritmo ckMeans ser mais rápido que o algoritmo Fuzzy C-Means decidiu-se aplicar também na área de sensoriamento remoto.

A seção 2 descreve o algoritmo ckMeans e o pré e pós-processamento nas imagens. Na seção 3 é discutido os experimentos com o algoritmo ckMeans aplicado no sensoriamento remoto. Finalmente, a seção 4 mostra as conclusões e trabalhos futuros.

2 Algoritmos

2.1 ckMeans

A ideia basicamente é que o conjunto $X = \{x_1, x_2, \dots, x_n\}$ seja dividido em p clusters, μ_{ij} é o grau de pertinência da amostra x_i ao j -ésimo cluster e o resultado do agrupamento é expresso pelos graus de pertinência na matriz μ .

O algoritmo ckMeans tenta encontrar conjuntos nos dados, minimizando a função objetivo mostrada na Equação (1):

$$J = \sum_{i=1}^n \sum_{j=1}^p \mu_{ij}^m d(x_i; c_j)^2 \quad (1)$$

onde:

- n é o número de dados;
- p é o número de clusters considerados no algoritmo o qual deve ser decidido antes da execução;
- $m > 1$ é o parâmetro da fuzzificação;
- x_i um vetor de dados de treinamento, onde $i = 1, 2, \dots, n$. Cada posição no vetor representa um atributo do dado;

- c_j é o centro de um agrupamento fuzzy ($j = 1, 2, \dots, p$);
- $d(x_i; c_j)$ é a distância.

A entrada do algoritmo são os n dados, o número de clusters p e o valor de m . Os passos são:

1. Inicialize μ com um valor aleatório contínuo entre zero (nenhuma pertinência) e um (pertinência total), onde a soma das pertinências deve ser um;
2. Calcule o centro do cluster j da seguinte maneira: Cria-se uma nova matriz μ , chamada de μC contendo valores 1 ou 0. Cada linha dessa nova matriz tem 1 na posição do maior valor dessa linha na matriz μ e zero nas demais posições da linha. Quando uma coluna da matriz μC , for toda com zeros, é atribuído o valor 1 na posição que corresponde ao maior valor dessa mesma coluna na matriz μ .

O algoritmo ckMeans retorna uma matriz μC conforme é mostrado na Equação (2). Ou seja, μC é a matriz enquanto μC_{ij} é o conteúdo dessa matriz na posição (ij) .

$$\mu C_{ij} = \max \left(\left\lfloor \frac{\mu_{ij}}{\max_{l=1}^p \mu_{il}} \right\rfloor, \left\lfloor \frac{\mu_{ij}}{\max_{l=1}^n \mu_{lj}} \right\rfloor \right) \quad (2)$$

Na função *max* são convertidos os dois argumentos no maior número inteiro que seja menor ou igual ao seu valor real. O primeiro argumento tem que cada dado tenha o valor 1 no cluster ao qual pertence com maior grau de pertinência e grau de pertinência zero nos demais. O segundo argumento tem por objetivo que o maior grau de cada coluna (cluster) seja 1. Para assim garantir que todo cluster tenha pelo menos um elemento. Dessa forma, em raras ocasiões, pode acontecer que uma linha tenha mais de uma valor 1 (o que não ocorre o algoritmo K-Means original), mas como esta matriz é apenas auxiliar, não ocasionará qualquer transtorno.

Os passos do algoritmo para calcular o μC_{ij} é realizado da seguinte forma:

- (a) Leia μ ;
- (b) Em cada linha encontrar o maior valor da matriz μ e atribuir 1 a essa mesma posição em μC e zero nas restantes;
- (c) Armazenar em um vetor a quantidade de 1's que cada coluna de μC possui.

Se uma coluna não tiver 1's marque sumariamente com 1 a posição onde está o maior valor. Após calculada a matriz μC calculam-se os novos centros dos clusters conforme a Equação (3).

$$c_j = \frac{\sum_{i=1}^n x_i \mu C_{ij}}{\sum_{i=1}^n \mu C_{ij}} \quad (3)$$

O c_j é calculado pelo somatório dos dados que pertencem ao cluster (de forma crisp) e dividido pela quantidade de objetos classificados como 1 na matriz μC deste cluster.

3. Calcule um valor inicial para J usando a Equação (1);
4. Calcule a tabela da função de pertinência fuzzy μ conforme mostrado na Equação (4)

$$\mu_{ij} = \frac{\left(\frac{1}{d(x_i; c_j)}\right)^{\frac{2}{m-1}}}{\sum_{k=1}^p \left(\frac{1}{d(x_i; c_k)}\right)^{\frac{2}{m-1}}} \quad (4)$$

5. Retornar a etapa 2 até que uma condição de parada seja alcançada.

Algumas condições de parada possíveis são:

- Um número de iterações pré-fixado for executado;
- O usuário informa um valor de parada $\epsilon > 0$, e se

$$d(J_U; J_A) \leq \epsilon$$

então para, onde J_A é a função objetivo (Equação (1)) calculada na iteração anterior e J_U é a função objetivo da última iteração.

2.2 Pré e Pós-Processamento

Para submeter as imagens ao algoritmo de agrupamento ckMeans foi necessário que estas passassem por um algoritmo de pré-processamento. Tal algoritmo, desenvolvido no *GNU Octave*, consiste em transformar a matriz tridimensional de pixels, a imagem a ser processada, em um vetor coluna S , Figura 1. Uma vez que o algoritmo ckMeans retorna uma matriz na qual cada linha possui o grau de pertinência do pixel aos clusters, faz-se necessário converter a matriz de pixels em um vetor coluna. Sendo assim, deve-se montar a imagem para tornar possível a interpretação dos padrões obtidos pelo algoritmo de agrupamento.

O algoritmo de pós processamento recebe esta matriz (denominada matriz C) e verifica a qual grupo o pixel tem maior grau de pertinência. Afim de facilitar o processamento a matriz C é convertida em vetor coluna S conforme mostrado na Figura 1(a). Os passos do algoritmo são descritos abaixo:

1. Avalia o conteúdo da matriz C e cria uma matriz nula M de mesma dimensão da Matriz C ;
2. Procura em cada linha o grupo ao qual o pixel possui maior grau de pertinência;

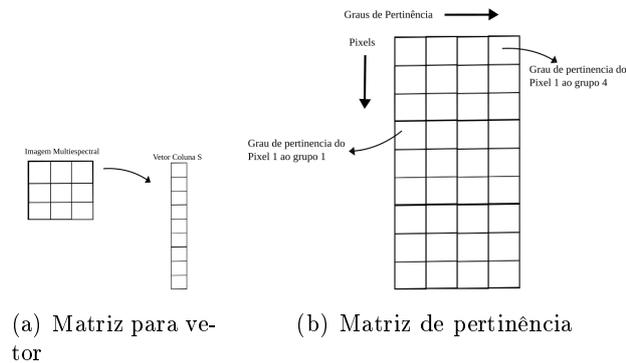


Figura 1: Tarefa realizada pelo algoritmo de pré-processamento.

3. Na matriz M , atribui valor 1 à posição do maior grau de pertinência e zero às demais posições.
 Note que assim é obtida uma matriz binária (com valores 0 e 1), onde cada linha possui apenas um valor 1, ou seja, o pixel está associado à apenas um agrupamento. Este procedimento é descrito a seguir e para melhor entendimento, a Figura 2 esquematiza todo o processo pelo qual a imagem é submetida, desde o pré até o pós-processamento;
4. Cada uma das colunas da Matriz M é transformada em uma nova matriz, ou seja, de M originam-se três novas matrizes contendo valores 0 e 1;
5. As três matrizes são multiplicadas pelo escalar 255, originando assim as três bandas da imagem (*Red*, *Green* e *Blue*);
6. As bandas são novamente unidas e a imagem está completamente montada.

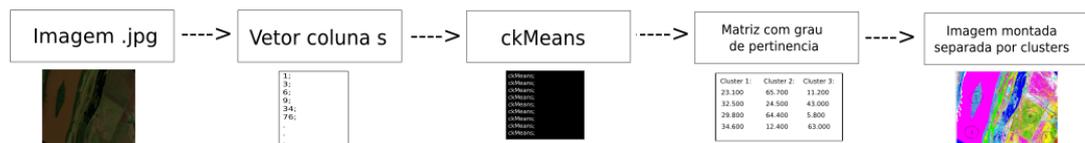


Figura 2: Tarefa realizada pelo algoritmo de pré-processamento.

3 Resultados

A imagem escolhida para validar a proposta foi obtida através do *Google Maps* possuindo a dimensão de 1179X764X3, ou seja, 1179 pixels de largura por 764 pixels de altura e 3 bandas (RGB). Executou-se o algoritmo ckMeans em um computador com sistema operacional GNU Linux, distribuição Ubuntu 14.04, com 4 GB de memória ram, processador Core i5, 2.50 Ghz.

Os parâmetros de entrada do algoritmo ckMeans foram: $\epsilon = 0,01$, $p = 5$ (grupos) e m (*fuzzificação*) variando em 1,25, 1,50 e 2. Devido a inicialização ser aleatória, foram geradas dez imagens para cada parâmetro m e foram selecionadas aquelas que tornam mais fácil a identificação visual dos padrões.

A Figura 3 mostra a imagem original ao lado das imagens obtidas com $m = 1,25$, $m = 1,5$ e $m = 2$.

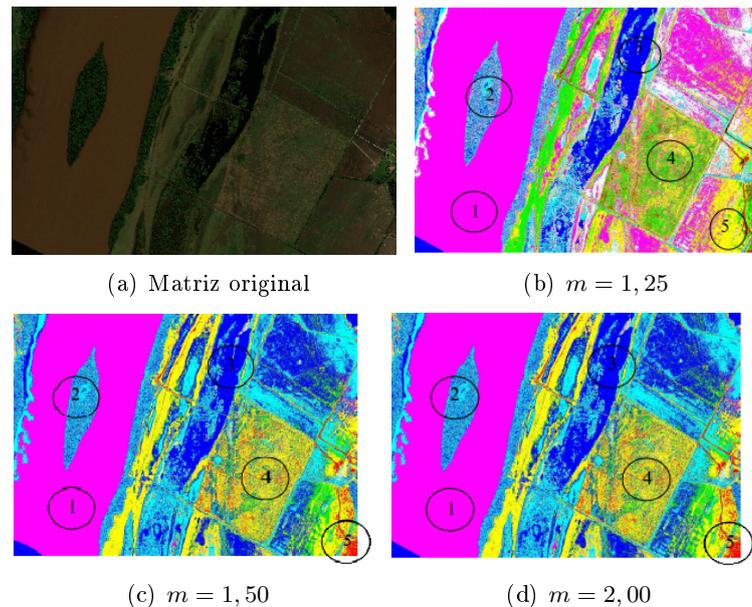


Figura 3: Resultado para $m = 1,25$, $m = 1,5$ e $m = 2$.

Analisando as imagens, é possível verificar que as imagens geradas com o parâmetro m em 1,25, 1,50 e 2 são similares, sendo que a única diferença encontra-se na cor utilizada em alguns agrupamentos. Através da análise de um especialista, pode-se classificar os grupos da seguinte maneira: 1 (água ou solo encharcado), 2 (mata ciliar), 3 (vegetação densa), 4 (vegetação rasteira) e 5 (solo exposto).

4 Conclusões

A análise de grupos, em muitos casos, não é um processo realizado em apenas uma execução, dado a estocasticidade dos métodos. Em muitas circunstâncias, é necessário uma série de tentativas, repetições e experimentações, uma vez que não há um critério universal e efetivo para guiar a seleção de atributos de algoritmos de agrupamentos. Em geral, critérios de validação provêm impressões sobre a qualidade dos clusters, mas como escolher este mesmo critério é ainda um problema que requer mais esforços [10].

Este trabalho propôs o uso do algoritmo ckMeans no processo de segmentação de imagens com uma aplicação no sensoriamento remoto. Os resultados se mostram satisfatórios por separar as imagens em grupos (distinguindo através de cores), os quais foram analisados

por um especialista da área.

Como trabalhos futuros pretende-se comparar o algoritmo ckMeans com outros algoritmos de agrupamento comparando os seus resultados no sensoriamento remoto e também o processamento separado dos três espectros de cores.

Referências

- [1] E. M. NOVO. *Sensoriamento Remoto. Princípios e Aplicações. 2 ed.* Edgard Blucher, São Paulo, 1993.
- [2] P. R. Menezes, E. E. Sano. Classificação Pixel a Pixel de Imagens. In: *MENEZES, P. R.; DE ALMEIDA T. (org.). Introdução ao processamento de imagens de Sensoriamento Remoto.* Brasília, 2012. p. 191-206.
- [3] A. T. Ohata, J. A. Quintanilha. A. O uso de algoritmos de clustering na mensuração da expansão urbana e detecção de alterações na Região Metropolitana de São Paulo. In: *Simpósio Brasileiro de Sensoriamento Remoto (XVII SBSR)*, Goiânia - Goiás, Brasil, 2005.
- [4] M. G. Noguchi, P. O. Bellon and H. Araki. Um sistema baseado em regras fuzzy para classificar imagem multiespectral do Ikonos. In: *Simpósio de Ciências e Tecnologias da Geoinformação (I SIMGEO)*, Recife - Pernambuco, Brasil, 2004.
- [5] R. R. de Vargas, B. R. C. Bedregal. A comparative study between fuzzy c-means and ckmeans algorithms. In *Proc. Conf. North American Fuzzy Information Processing Society (NAFIPS 2010)*. Toronto, Canada, 2010.
- [6] R. R. de Vargas, B. R. C. Bedregal and G. Dimuro. Using ckMeans algorithm in image segmentation process: Preliminary results on mammography analysis. In *Anais do Congresso Nacional de Matemática Aplicada e Computacional (XXXV CNMAC)*, Natal, Rio Grande do Norte, Brasil, 2014.
- [7] J. Oldeland, W. Dorigo, L. Lieckfeld, A. Lucieer and N. Jürgens. Combining vegetation indices, constrained ordination and fuzzy classification for mapping semi-natural vegetation units from hyperspectral imagery. *Remote Sensing of Environment*. Volume 114, Issue 6, 15 June 2010, Pages 1155-1166.
- [8] F. M. Sousa, S. Nascimento, H. Casimiro and D. Boutov. Identification of upwelling areas on sea surface temperature images using fuzzy clustering. *Remote Sensing of Environment*. Volume 112, Issue 6, 16 June 2008.
- [9] Q. Shen, J. Li, F. Zhang, X. Sun, J. Li, W. Li and B. Zhang. Classification of Several Optically Complex Waters in China Using in Situ Remote Sensing Reflectance. *Remote Sens.* 2015.
- [10] J. Cavalcanti, J. Clusterização Baseada em Algoritmos Fuzzy. *Tese de mestrado, Universidade Federal de Pernambuco* Recife, Brasil. 2006.