

An Introduction to Image Clustering with Affinity Propagation

J. C. Leitão A. de Farias
PPGINF
UFABC
São Paulo, SP – Brazil
julio.leitao@ufabc.edu.br

Abstract— This paper shows how works the affinity propagation method applied to an image dataset and concludes showing the difference between image comparing and face recognition.

Keywords—affinity, propagation, olivettifaces, image.

I. INTRODUÇÃO

Clusterização é um procedimento adotado para criar classes (clusters) para dados utilizando-os de forma a dividi-los em grupos de forma que dados semelhantes fiquem agrupados e dados distintos fiquem separados. Dois dos métodos bastante utilizados são o k-means e sua variação k-medians, cuja diferença é que o valor de atualização dos representantes no algoritmo pode ser baseado na média ou na mediana dos dados, e esse último ajuda a dar robustez aos outliers. Em ambos os métodos é dado como entrada o número de clusters para a divisão do dataset a ser estudado.

Em 2007 foi introduzido um novo algoritmo para clusterização chamado de Affinity Propagation (AP) em [1] que, comparado aos métodos acima, trouxe os benefícios de convergir em menos iterações e não necessitar do número de clusters como entrada do algoritmo.

AP se baseia em uma rede formada por todos os vetores de atributos que estão presentes no dataset. Cada vetor é um nó na rede e todos trocam mensagens entre si para identificar quais são os melhores representantes para cada vetor de atributos. Após a estabilização do algoritmo, um subconjunto dos dados é escolhido como conjunto dos clusters. Cada dado de entrada é categorizado como um dos clusters disponíveis, o seu exemplar, ou seja, seu dado mais representativo dentre os clusters disponíveis.

II. FUNCIONAMENTO DA REDE

A rede funciona com base em dados iniciais sobre a similaridade um a um dos dados e um conjunto de dois tipos de mensagens, as de responsabilidade e de disponibilidade.

A. Similaridade

Conforme definido em [1] a similaridade $s(i, k)$ indica quão bem um dado de índice k está ajustado a outro dado de índice i . Quando o objetivo é minimizar o erro médio quadrático a similaridade $s(i, k)$ recebe o negativo da distância euclidiana, conforme (1):

$$s(i, k) = -|x_i - x_k|^2 \quad (1)$$

Entretanto outros modelos para similaridade podem ser usados.

B. Mensagens de Responsabilidade

São mensagens enviadas de um vetor de nó i para um nó k que indicam a evidência acumulada de quão bem o nó k representa o nó i , e são expressas por (2). Em [1] é atribuída

a denominação de *exemplar* para os nós que se tornam clusters.

$$r(i, k) = s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\} \quad (2)$$

C. Mensagens de Disponibilidade

São mensagens enviadas de nó k para um nó i , conforme (3). Indicam a evidência acumulada de quão apropriado seria o nó i escolher o nó k como exemplar (cluster).

$$a(i, k) = \begin{cases} \min \{0, r(k, k) + \sum_{i' \neq \{i, k\}} \max \{0, r(i', k)\}\}, & \text{se } i \neq k \\ \sum_{i' \neq \{k\}} \max \{0, r(i', k)\}, & \text{se } i = k \end{cases} \quad (3)$$

D. Funcionamento da rede

Para organizar essas mensagens se utilizam matrizes que guardam os valores. As matrizes de disponibilidade, A , e de responsabilidade, R , são iniciadas com zeros, a matriz de similaridade, S , é criada conforme (1). Todas têm dimensões $N \times N$, onde N é a quantidade de nós na rede.

Mensagens $r(i, k)$ são enviadas conforme (2), a matriz R é atualizada com um coeficiente de amortecimento (damping) λ , segundo (4):

$$R_{t+1} = (1 - \lambda) R_t + \lambda R_{t-1} \quad (4)$$

Analogamente, mensagens $a(i, k)$ são enviadas conforme (3), a matriz A é atualizada com um coeficiente de amortecimento (damping) λ , segundo (5):

$$A_{t+1} = (1 - \lambda) A_t + \lambda A_{t-1} \quad (5)$$

Conforme [1], para cada nó i , o valor de k que maximize $a(i, k) + r(i, k)$ identifica k como sendo o exemplar de i . As iterações ocorrem por um número determinado previamente ou podem ter como condição de parada quando os exemplares não se modificarem após um certo número de iterações.

III. EXPERIMENTO E DISCUSSÃO

Foi analisado o dataset olivettifaces, composto 10 fotos de 40 pessoas, totalizando 400 fotos, cada uma em 56x46 pixels com valores inteiros entre 0 e 255. Cada imagem foi transformada em um vetor de 2576 atributos, conforme Fig.1. Cada vetor funcionou como um nó da rede proposta, como visto na Fig.2.

Após a aplicação do algoritmo, o resultado foi a criação de 66 clusters, de 40 esperados, com acerto de 357 entre pessoas e seus clusters.

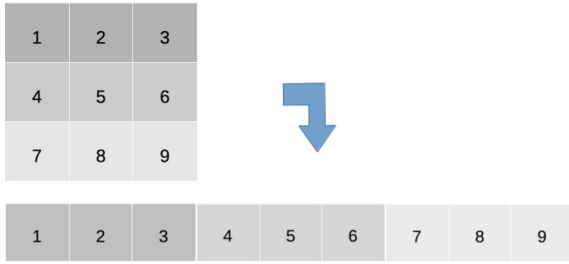


Fig.1. Transformação aplicada às imagens para vetorizá-las.

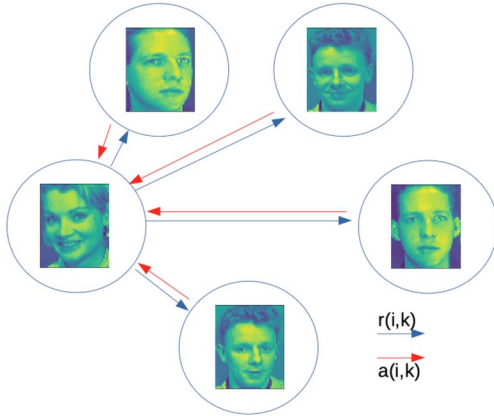


Fig.2. Parte da rede montada com o algoritmo, mensagens são trocadas entre os nós conforme (2) e (3).

É importante notar que os resultados podem apontar imagens parecidas como pertencentes ao mesmo cluster devido ao cálculo ser com base no valor do pixel na posição de uma foto em comparação com um pixel na mesma posição em outra foto.

Conforme Fig.3, é possível notar que a pessoa de cima à esquerda tem sobrancelhas, olhos, boca e cabelo bastante diferentes da pessoa à sua direita, entretanto foram agrupadas no mesmo cluster.

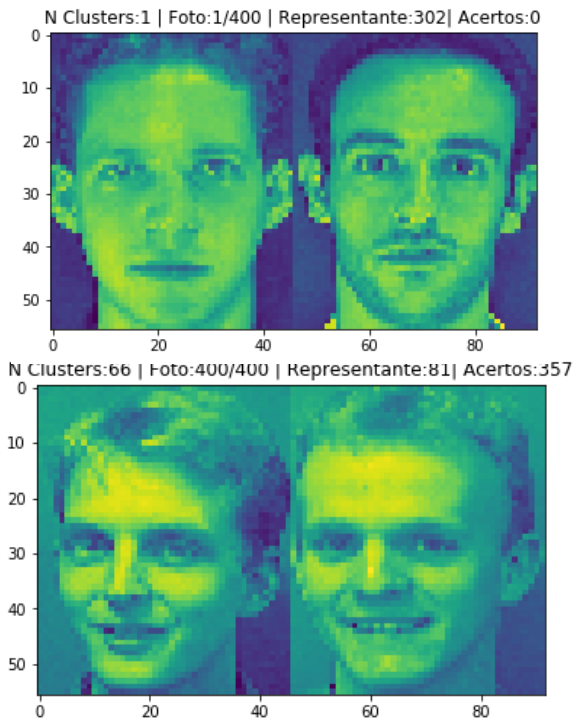


Fig.3. Fotos de pessoas diferentes categorizadas como mesmo cluster devido à análise se dar apenas sobre o valor do pixel na foto.

Essas imagens foram categorizadas no mesmo cluster basicamente por proximidade entre os valores dos pixels delas. Apesar de serem pessoas diferentes tratam-se de imagens semelhantes em tonalidade dos pixels de mesma posição. Um deslocamento de uma das pessoas na imagem pode mudar completamente o valor do seu cluster. Atributos que diferem bastante uma pessoa de outra, como sobrancelhas e olhos, acabam ocupando pouco espaço na foto e, conseqüentemente, tendo seu peso reduzido no algoritmo, nesse sentido, a cor do cabelo teria mais importância (maior área na foto) que o formato da boca. O algoritmo detectou bem imagens semelhantes mas não necessariamente detecta pessoas semelhantes.

Existem algoritmos para reconhecimento de pessoas utilizam proporções entre seus atributos faciais e ajudam a minimizar os erros.

Algumas imagens de uma mesma pessoa tiveram variações grandes de posição e isso gerou mais clusters.

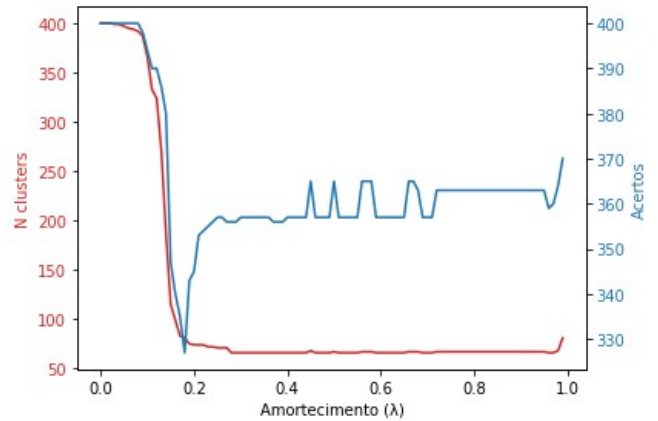


Fig.4. Relação entre o número de clusters criados, número de acertos nas fotos e o coeficiente de amortecimento.

O experimento foi realizado para diversos λ , o resultado é visto na Fig.4. É possível notar que, para esse dataset, o número de clusters criados se estabiliza com $\lambda > 0.2$ e o número de acertos entre os rostos e seus representantes com $\lambda > 0.3$.

O código utilizado para a implementação está disponível em <<https://github.com/julio-cesar-leitao/UFABC-INF317B-ProjetoFinal>>.

IV. CONCLUSÃO

Esse artigo demonstrou o método de AP aplicado a imagens e o funcionamento da rede gerada, idealizada em [1]. O método estudado é capaz de criar os clusters sem recebimento prévio do seu número total. O algoritmo implementado obteve 66 clusters de 40 desejados para 400 imagens. Demonstrou também que há motivação para os erros encontrados na clusterização no caso de uso com o dataset olivettiface e reforçou que imagens de pessoas diferentes podem ser semelhantes.

REFERÊNCIAS

- [1] B.J. Frey and D. Dueck, "Clustering by Passing Messages between Data Points," Science, vol. 315, no. 5814, pp. 972 – 976, Feb. 2007.
- [2] Guan, Renchu & Shi, Xiaohu & Marchese, Maurizio & Yang, Chen & Liang, Yanchun. (2011). Text clustering with seeds affinity propagation. IEEE Trans Knowl Data Eng. IEEE Trans. Knowl. Data Eng., 23, 627-637. 10.1109/TKDE.2010.144.